

UNIVERSITY OF THE PHILIPPINES MANILA  
COLLEGE OF ARTS AND SCIENCES  
DEPARTMENT OF PHYSICAL SCIENCES AND MATHEMATICS

COLLABORATIVE TAGGING AND RATING-BASED  
COLLABORATIVE FILTERING IN A CLASSIFIED ADS  
SYSTEM (COLLAB-ADS)

A special problem in partial fulfillment  
of the requirements for the degree of  
**Bachelor of Science in Computer Science**

Submitted by:

Monica K. Castro

April 2014

Permission is given for the following people to have access to this SP:

Available to the general public	Yes
Available only after consultation with author/SP adviser	No
Available only to those bound by confidentiality agreement	No

## ACCEPTANCE SHEET

The Special Problem entitled “Collaborative Tagging and Rating-based Collaborative Filtering in a Classified Ads System (Collab-Ads)” prepared and submitted by Monica K. Castro in partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science has been examined and is recommended for acceptance.

---

**Avegail D. Carpio, M.Sc.**  
Adviser

### EXAMINERS:

	Approved	Disapproved
1. Gregorio B. Baes, Ph.D. ( <i>candidate</i> )	_____	_____
2. Richard Bryann L. Chua, M.Sc.	_____	_____
3. Aldrich Colin K. Co, M.Sc. ( <i>candidate</i> )	_____	_____
4. Ma. Sheila A. Magboo, M.Sc.	_____	_____
5. Vincent Peter C. Magboo, M.D., M.Sc.	_____	_____
6. Geoffrey A. Solano, M.Sc.	_____	_____
7. Bernie B. Terrado, M.Sc. ( <i>candidate</i> )	_____	_____

Accepted and approved as partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science.

<hr/> <b>Ma. Sheila A. Magboo, M.Sc.</b> Unit Head Mathematical and Computing Sciences Unit Department of Physical Sciences and Mathematics	<hr/> <b>Marcelina B. Lirazan, Ph.D.</b> Chair Department of Physical Sciences and Mathematics
---	---

---

**Alex C. Gonzaga, Ph.D., Dr.Eng.**  
Dean  
College of Arts and Sciences

## **Abstract**

As Classified ads systems are growing more in trend than offline classified advertising, more resources are being created and thus creating a need for organization and improved ways for users to discover these resources.

The Collaborative Tagging and Rating-Based Collaborative Filtering in a Classified Ads (Collab-Ads) system addresses these needs by offering a way for users to tag or label resources and allow them to rate these resources to be able to generate recommendations.

*Keywords:* Collaborative Tagging, Collaborative Filtering, Slope One Algorithm

# Contents

Acceptance Sheet	i
Abstract	ii
List of Figures	v
List of Tables	vii
<b>I. Introduction</b>	<b>1</b>
A. Background of the Study . . . . .	1
B. Statement of the Problem . . . . .	2
C. Objectives of the Study . . . . .	3
D. Significance of the Project . . . . .	4
E. Scope and Limitations . . . . .	5
F. Assumptions . . . . .	5
<b>II. Review of Related Literature</b>	<b>7</b>
<b>III. Theoretical Framework</b>	<b>11</b>
A. Classified Ads System . . . . .	11
B. Tagging and Collaborative Tagging Systems . . . . .	12
C. Recommender Systems . . . . .	15
C..1 Collaborative Filtering . . . . .	16
C..2 Weighted Slope One Method . . . . .	17
D. OSClass . . . . .	18
E. Database Management Systems (DBMS) . . . . .	19
<b>IV. Design and Implementation</b>	<b>20</b>
A. Context Diagram . . . . .	20
B. Use Case Diagram . . . . .	21
B..1 Top Level Use Case Diagram . . . . .	21

B..2	Register Account . . . . .	21
B..3	Search Listings . . . . .	22
B..4	Add Listing . . . . .	23
B..5	Report Listing . . . . .	24
B..6	Contact Publisher . . . . .	25
B..7	Commenting on a Listing . . . . .	26
B..8	Manage Owned Listings . . . . .	27
B..9	Rate Listing . . . . .	27
B..10	Manage Tags . . . . .	28
B..11	Manage Listings . . . . .	30
B..12	Manage Users . . . . .	31
B..13	Manage Administrators . . . . .	33
B..14	Manage Categories . . . . .	34
B..15	Manage Comments . . . . .	35
C.	Entity Relationship Diagram . . . . .	37
D.	Data Dictionary . . . . .	37
<b>V.</b>	<b>Technical Architecture</b>	<b>47</b>
<b>VI.</b>	<b>Results</b>	<b>48</b>
<b>VII.</b>	<b>Discussions</b>	<b>58</b>
<b>VIII.</b>	<b>Conclusions</b>	<b>60</b>
<b>IX.</b>	<b>Recommendations</b>	<b>61</b>
<b>X.</b>	<b>Bibliography</b>	<b>62</b>
<b>XI.</b>	<b>Appendix</b>	<b>65</b>
A.	Source Code . . . . .	65
<b>XII.</b>	<b>Acknowledgement</b>	<b>310</b>

## List of Figures

1	Tripartite graph structure of the tagging system . . . . .	13
2	Data model for a simple tagging system [1] . . . . .	14
3	The data model for a collaborative tagging system [1] . . . . .	15
4	Weighted Slope One Algorithm Equation . . . . .	17
5	Context Diagram for the Collab-Ads system . . . . .	20
6	Top Level Use Case Diagram of Collab-Ads . . . . .	21
7	Activity Diagram for Registering an Account in Collab-Ads . . . . .	22
8	Activity Diagram for Using Search in Collab-Ads . . . . .	23
9	Activity Diagram for Adding a Listing in Collab-Ads . . . . .	24
10	Activity Diagram for Reporting and Managing Reported Listings in Collab-Ads . . . . .	25
11	Activity Diagram for Contacting a Publisher in Collab-Ads . . . . .	26
12	Activity Diagram for Commenting on a Listing in Collab-Ads . . . . .	26
13	Activity Diagram for Managing a Users Own Listings in Collab-Ads . . . . .	27
14	Activity Diagram for Rating a Listing in Collab-Ads . . . . .	28
15	Activity Diagram for Adding Tag to a Listing in Collab-Ads . . . . .	29
16	Activity Diagram for Editing Tags in a Listing in Collab-Ads . . . . .	30
17	Administrator Activity Diagram for Adding a Listing . . . . .	30
18	Activity diagram for editing a listing by Administrator or Moderator . . . . .	31
19	Activity diagram for adding a user to the system by the administrator . . . . .	32
20	Activity diagram for editing a user account . . . . .	32
21	Activity diagram for adding an administrator/moderator account . . . . .	33
22	Activity diagram for editing an admin/moderator account . . . . .	34
23	Activity diagram for adding categories . . . . .	34
24	Activity diagram for editing a category . . . . .	35
25	Activity diagram for managing comments . . . . .	36
26	Administrator Login Form . . . . .	48
27	User Management Page . . . . .	48

28	Administrator View: Admin Management Page . . . . .	49
29	Administrator View: Listings Management Page . . . . .	50
30	Administrator View: Reported Listings Page . . . . .	50
31	Administrator View: Plugins Page . . . . .	51
32	Administrator View: Comments Management Page . . . . .	51
33	Registered User Homepage . . . . .	52
34	Listing Publish Form . . . . .	53
35	Registered User: Viewing a Listing . . . . .	54
36	Registered User: Listing Recommendations . . . . .	54
37	Registered User: Listing Comment Section . . . . .	55
38	Registered User: Owner View of a Listing . . . . .	56
39	Registered User: View of Owned Listings . . . . .	56
40	Registered User: Editing Account . . . . .	57

## List of Tables

1	Examples of tag types according to Smith [1] . . . . .	14
2	Example illustrating slope one approach . . . . .	18
3	t_admin - table for administrator and moderator accounts . . . . .	37
4	t_category contains list of all categories where listings are to be filed under . . . . .	38
5	t_category_description holds name and description of each category	38
6	t_category_stats holds basic statistics on listings under each category	38
7	t_city contains list of all cities where listings are located . . . . .	38
8	t_city_area contains list of all areas inside a particular city where listings are located . . . . .	39
9	t_city_stats holds basic statistics on listings in a city . . . . .	39
10	t_country contains list of all countries made available to listings . .	39
11	t_country_stats holds basic statistics on listings for each country . .	39
12	t_dev contains slope one values of rating pairs to be used for com- puting predictions . . . . .	39
13	t_item contains list of all listings posted in the website . . . . .	40
14	t_item_comment contains list of all comments made on a particular listing . . . . .	41
15	t_item_description holds the title and content of a listing . . . . .	41
16	t_item_location holds the location of a listing . . . . .	42
17	t_latest_searches contains latest search entry . . . . .	42
18	t_locale contains list of all available locales in system set by ad- ministrator . . . . .	43
19	t_rated_items contains rating value made by user for a certain item	43
20	t_region contains list of all regions to file listings under . . . . .	44
21	t_region_stats contains basic statistics of listings under a certain region . . . . .	44
22	t_tagger_tags holds information about a tag . . . . .	44



23	t_tagger_tagged_objects	holds information about a tagging . . . . .	44
24	t_user	contains list of all users who registered in the website . . . . .	45
25	t_user_description	holds description of a user in their user account . . . . .	46
26	t_user_email_tmp	holds data for when requesting an email change . . . . .	46

# I. Introduction

## A. Background of the Study

Back in 2003, a social bookmarking website called Del.icio.us was launched. The site allowed users to manage their links by putting a short label, or a tag which allowed them to pull up pages with the same tag. The site was created upon discovering that because of the massive amount of information available on the web, people not only wanted to find good links, but also wanted to find out what others were interested in. [2]

User-generated keywords or tags have appeared to be a lightweight way of augmenting descriptions to online resources, and improving access to these resources through broader indexing. Publicly labelling or categorizing resources in a shared online environment is referred to as social tagging.[3] When many users add tags to resources, we refer to it as collaborative tagging. [4] Because of the easy and intuitive nature of annotating, organizing, and retrieving resources provided by tagging, the popularity of social tagging systems has increased tremendously in recent years.[5] Tagging not only benefits its users but also provides benefits to business. It facilitates collaboration, helps in obtaining descriptive metadata, in identifying patterns, and in augmenting classification efforts, enhances findability, increases participation, and sparks innovation. [1]

Apart from tagging, another innovative tool to help organize or mine content is what we call collaborative filtering. This tool allows users to rate resources which are then used to form recommendations. Through this kind of personalization based on user preferences, other users can find better content, or it can lead them to other resources they might be interested in. [6]

Needing these kinds of benefits is one such technology that has risen in popu-

larity which is the online classified ads. [7] Classified ads sites basically serve as a one-stop-shop for everything. However, something that separates online classified ads sites from auction or shopping websites (e.g. eBay and Amazon.com) is that users don't buy anything directly from these sites. Classified ads sites are used to set up meetings and transactions are conducted in person or by mail. [8]

## **B. Statement of the Problem**

As online classified ads are growing more in trend than offline classified advertising, both for market users as well as for investor markets, the desire to enhance classified ads systems is growing as well. These trends we are seeing in online classified advertising are representative of what may be the start of a significant shift in marketing caused by the power being transferred to the customer by the Internet. These days, buyers are empowered by instant access to information, services, and transactional ability. [7] Something that should be considered is that classified ads sites are dependent on search find obtain (SFO) functionalities since the usage of such sites exhibit a general user-resource retrieval pattern.

As online classified ads can be treated as information or web resource, given the trends, there would also be a growth in number of these resources. As the numbers grow, there will also be a need for these resources to be organized or sorted out. Since a larger amount of resources would inevitably vary in nature, it will become more difficult to sort them out to appropriate categories. Also, users who are trying to look for a resource can have a difficult time finding what they need because of the broad, non-definitive categories. For users who are trying to sell products, it can be difficult to attract possible buyers by simply leaving an ad in a single category. Even if it is under a hierarchy leading to a narrower index, it could still get buried with a lot of other ads that could have a lot of different attributes from it.

Another common drawback in existing classified ads systems is that the systems are supply-oriented and does not allow a user to describe a resource that implies demand. In existing systems, it is not possible to describe both the desired resource and the one up for offer (i.e. exchange, wanted). Also, existing systems lack the feature of making descriptions of search queries available to other users. [9]

These days, user personalization can be one of the keys to many successful web sites. [6] In current classified ads systems, there are a few tools to help organize content and lead users to what they desire compared to the number of resources that is available.

### **C. Objectives of the Study**

To develop the Collaborative Tagging and Rating-based Collaborative Filtering in a Classified Ads System (Collab-Ads) with the following functionalities:

1. Allows the system administrator to:

- (a) Manage users
- (b) Manage moderators
- (c) Manage ads
- (d) Manage categories

2. Allows the moderator to:

- (a) Manage users
- (b) Manage ads

3. Allows a registered user to:

- (a) View existing ad

- (b) Post a new ad
- (c) Leave a comment on an ad
- (d) Send private message to the author of an ad
- (e) Share an ad to a person via integrated email
- (f) Add/delete tags to an ad
- (g) Rate a classified ad
- (h) Search for an ad by keyword or by tags
- (i) Mark an ad with the following markings: spam, misclassified, duplicated, expired, offensive

4. Allows a non-registered user to:

- (a) Register for an account
- (b) View existing ads and contact details of an ad's owner

5. Allows the system to:

- (a) Send a private message to the owner of an ad for any changes done by the system administrator to his ad
- (b) Show users recommended ads

## **D. Significance of the Project**

This project aims to provide a tool for a classified ads system that helps make way for countless potential in information retrieval, such as improving accuracy of searches, discovery of related resources, and organizing and filtering mass information. The project hopes to bring an improved effectiveness in retrieval as in [Lu and Kipp]'s experiment which found that more relevant results are retrieved through the use of collaborative tagging. Additionally, the project also hopes to bring effectiveness in describing resources by using collaborative tagging. [Xu et

al.]

In this project, we make use of collaborative tagging as a way to address the lack of ease in users' experience in classified ads systems. This system would help sellers attract more users to their ads by putting up tags that are informative and relevant to the product and attractive to other users. This system would help prospective sellers to gather information as to which products are in trend by observing the popular tags. This system would help buyers navigate through broad categories, refining their searches to details which they fancy by clicking on tags.

Aside from collaborative tagging, this system will also implement collaborative filtering which allows the users to rate the classified ads then use these ratings to predict other ads they might like. The ads that will be recommended therefore will be formed collectively from other users ratings. This means that users will be able to help each other find good content. Collaborative filtering will help users receive recommendations that are personalized without going through the trouble of asking or looking for it.

## **E. Scope and Limitations**

Listed below are the scope and limitations of the study:

1. The tagging system will not include a tag recommendation feature.
2. The tags will not be sorted into any kind of hierarchy, sorting, or visualization (e.g. clustering, facets).
3. The system will not have any tag spam cleaning scheme.

## **F. Assumptions**

Here are some assumptions taken into consideration when making the project:

1. The users will post ads of items or services that are real.
2. The users will not abuse free tagging to spam with irrelevant, misleading information.

## II. Review of Related Literature

Classified advertising is a form of online advertising which has been given little academic research due to the small availability of data. Craigslist.org, for example, receives around 9 billion pageviews a month (Blodget (2008)), making it a powerhouse in the online classified ads world, but is also notorious for protecting its users privacy, and therefore, does not share its data. [9]

These online classified advertising systems are becoming one of the most popular and rapidly growing classes of Internet technologies. The high traffic to online classified advertising systems proves their relevance and importance. A research by Hitwise in 2009 shows that the amount of traffic to online classified advertising systems continues to grow. It has increased 84 percent in February 2009 compared to 2008. Today, a lot of software tools for development of online classified advertising systems are available (i.e. Esvon Classifieds, e-Classifieds, PHP Classifieds Script). Along with these are standalone systems that were built without using any public software tools where the most popular advertising belong to (i.e. Craigslist, Backpage, Oodle, Geebo, Kijiji, Facebook Marketplace). [10]

Another trend that has grown in popularity on the web is the collaborative form of tagging. Traditionally, a person with authority such as a librarian will perform the categorizing or indexing, or else derived from the material provided by the authors of the documents. [11] Collaborative tagging, in contrast, allows anyone especially its consumers to attach keywords freely to the resources. This is most useful when there is nobody assuming the librarian role or there is simply too much content for assigned authorities to classify; and both of these traits are true of the web in which this kind of system has grown popular. Collaborative tagging is featured in a number of now prominent web sites. Typically, these sites would use this feature to allow users to not only categorize information for themselves, but to also browse information categorized by others. Therefore, public



and private aspects to collaborative tagging systems are present. [4]

BibSonomy [12], an example of a tagging system, allows users to add or edit tags to their own resources which could either be bookmarks or publication references. Sharing of these resources on the other hand is free for all, and adding a resource to your collection from another user grants you the tagging ability for that resource. [13] This particular site is just one of many examples to illustrate the growth of tagging in a variety of media.

A study by Steels [14], discusses collaborative tagging being relevant to distributive cognition. As part of her study, she made examples from the sites Del.icio.us and Flickr. The site Del.icio.us pioneered collaborative tagging sites back in 2004 and was used to tag bookmarked web site links. Soon came after was Flickr whose tags were assigned to pictures. Since then, collaborative tagging exploded, and is rapidly becoming a standard feature of websites. In the study, it was realized that tags are somewhere in between symbols that stand for something in the world, and symbols that act as labels or signposts to aid in navigation.

Golder et al. [4] focused another study on usage patterns in a collaborative tagging system and based it on the site Del.icio.us. Their study discovered regularities that create stable patterns, however frequent or infrequent users tag and however largely varying the sets of their tags are. This allowed minority opinions to coexist with extremely popular ones without causing a disruption to the stable consensus choices of tags made by many users.

According to Halpin et al. [15], there is greater malleability and adaptability in organizing information in tagging systems than in formal classification systems. Mathes [16] states that proponents of tagging systems argue that users only need to agree in a general sense on a tags meaning enough to label a similar material

with terms, and that they don't have to agree on a hierarchy of tags for there to be cooperation and shared value. Additionally, tagging has the ability to retrieve data and share data more efficiently than classifying. It is argued that it is easier to freely type loose associations than to make a decision about the degree of match to a predefined category (especially hierarchical ones). It is compared to 90% of the value of taxonomy, only 10 times simpler. [15] [17]

There have been studies which explored the use of collaborative tagging in different systems. A proposal made by Durieux [18] in 2011 wanted to explore the potential of collaborative tagging in healthcare professional systems. In a project by Back et al. [19], a research on tagging was done on mobile applications and web applications to understand its role on the user experience and also to bring value to companies in developing their future media services. In a study by Bateman et al., collaborative tagging was used for within an e-learning system. This was upon arguing that social collection phenomena and flexible metadata standards are the keys to collecting the kinds of metadata required for adaptable online-learning. While other systems successfully concluded benefits from applying collaborative tagging, this study ended up questioning its benefits to e-learning systems. The project faced issues such as the lack of expert time and willingness to fill in metadata and considered these as significant hurdles in deploying learning objects. [20]

Lastly, recommendation systems have also grown in trend also addressing the issue of selecting from a huge universe of items with widely varying quality. One of its types, collaborative filtering allows users to get recommendations from people with similar taste. Terveen et al. [21] states that there are three pillars for this approach namely, (1) many people must participate, (2) there must be an easy way for people to represent their interests in the system, and (3) algorithms must be able to match people with similar interests. Through time, a collaborative filtering system acquires an increasingly accurate representation of a users prefer-

ences. They also pointed out that the primary strength of collaborative filtering is that recommendations are personalized. Second, users don't need to look for recommenders anywhere else but will be shown recommendations from stating their preferences. Finally, from a computational point of view, the data representation is simple and uniform which makes use of a user-item matrix whose cells represent ratings, amenable to many different computational manipulations.

A few of the earliest collaborative filtering systems that were developed include GroupLens research system and the Bellcore Video Recommender. The GroupLens research system provided a pseudonymous collaborative filtering solution for Usenet news and movies. Video Recommender, meanwhile, is an email and web-based system that generate recommendations on movies. These systems varied in how they weighted the ratings of different users and how they combined the ratings. On the web, collaborative filtering has found many applications as in ecommerce sites such as Amazon.com and CDNow which feature recommendation centers that make use of user ratings and collaborative filtering engines. [20] [22]

### III. Theoretical Framework

#### A. Classified Ads System

MorganStanley defines online classified advertising as an interactive advertisement with search and find functionality, usually brief but linked to relevant information, listed online under headings with others of the same category. Frequently, these advertisements have embedded transaction capabilities that allow a user to facilitate obtaining the product/service he/she is interested in. [7] Classified ads systems are used mainly as a place to find products or services which people can buy, but not necessarily through the site. It is where people can set up meetings or arrange deals towards the item being sold. [8]

Users of classified ads systems would usually want to find a user, a contact and interaction with whom may lead to the matching of their mutual interests. General and vertical search engines are also capable of doing these, but there are some differences that make online classified advertising systems a separate class. A general search engine offers the widest search range, but the lowest depth in search. Vertical search engines, on the other hand, are intended for information retrieval in specific domains. This means they allow the deepest search, but only in a narrow field. Meanwhile, the class of online classified advertising systems is intermediate between the two classes. [10]

According to Gribova et al. [10], in today's existing online classified advertising systems, these three methods are used to match user interests: Firstly, hierarchical classification of the ads wherein each listing in a system belongs to a certain category from a set of hierarchical categories. This way of structuring grants users the ability to look through the subset of all classifieds in a system instead of the full, and possibly massive, list. The other is either structural, or natural language input, or a combination of both. A classified advertising system has a standard

structure of containing a set of two fields, namely the Title and Description. Simple classified ads systems will allow users to fill only these two fields when posting a listing, but in more complicated systems, an administrator can allow users to fill in additional attributes for the listing depending on which category it is under. This causes the description of an ad authors interest more formal and convenient to search. Lastly, there is either full-text search or the parameter search based on the classified ad structure. A standard method for the existing systems is the full-text search which allows users to find listings that contain specific keywords in their descriptions and titles. If the classified ad structure of the system is complex, it can also use additional fields as search parameters. Both search types will allow a user to choose the category in which to find the desired listing. Saving a search query to be used repeatedly as new listings are posted is also enabled in some systems.

## **B. Tagging and Collaborative Tagging Systems**

There are three main entities that compose a tagging system. Each of these entities can be viewed as three separate spaces containing sets of vertices which are linked together by edges (see Figure 1.) The first space consists of the set of all users of the tagging system, where each vertex is a user. The second space is the set of all tags as vertices, where each tag corresponds to a term or neologism in natural language. The third space is the set of all resources where each resource is more commonly denoted by a unique URI. A tagging instance is depicted as two edges linking together a user to a tag to a given resource. [15]

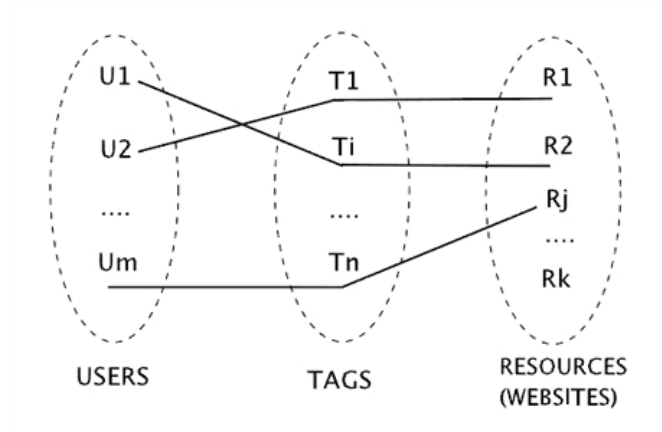


Figure 1: Tripartite graph structure of the tagging system

Users or “taggers” are the people who employ the tagging system. They are the ones who create the tags, and sometimes they also add the resources. Resources are web content which the users tag. Some resources in tagging systems are publications, web pages, videos, photos and can be just about anything. Anything can be tagged as long as there is a way to uniquely identify it. Tags are the keywords added by users. Tags are essentially metadata about the resource. The National Information Standards Organization describes metadata as structured information that describes, explains, locates, or otherwise makes it easier to retrieve, use, or manage an information resource. Tagging is open-ended and this allows tags to be just about any kind of term. They can be descriptions of the resources subject matter, its location, its intended use, a reminder, or something else entirely. It all depends on the users personal tagging patterns.

## Tag Type Examples

Descriptive	css, webdesign, ajax, Minnesota, drama, gardening, zen
Resource	blog, book, video, photo
Ownership/Source	nytimes, genesmith (author), newriders
Opinion	cool, funny,*****, lame, beautiful, crap, defective by design
Self-reference	mystuff, mine, me
Task Organizing	toread, todo, work
Play and Performance	squaredcircle, seenlive, aka vogon poetry

Table 1: Examples of tag types according to Smith [1]

All tagging happens in the context of a system. In a tagging system, it is defined what practice of tagging can take place. For example, adding tags may be limited to the publishers of resources or to everyone. Or, the system might allow users to set tags as public or private. In other words, a tagging system contains the rules about who can tag, what can be tagged, and what kind of tags can be used.

In any case, the architecture of a tagging system requires rules about the user (who they are and how they join the system), the resources (what they are and how they are added to the system), and tags (who can tag which resources). [1]

Figure 2 shows a data model of a simple tagging system. In this system, there is no collaborative tagging. Each user has multiple resources; each resource is unique to each user; and each resource has multiple tags.

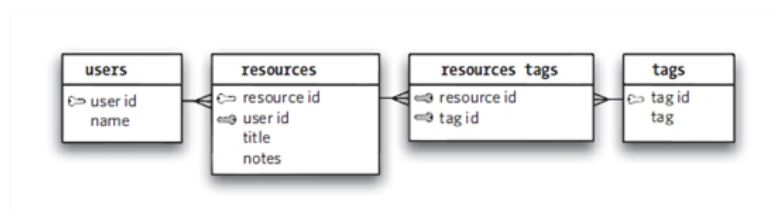


Figure 2: Data model for a simple tagging system [1]

The four main tables for this tagging system are users, resources, tags, and resources\_tags. The resources\_tags table holds the information of which tags are applied to which resources.

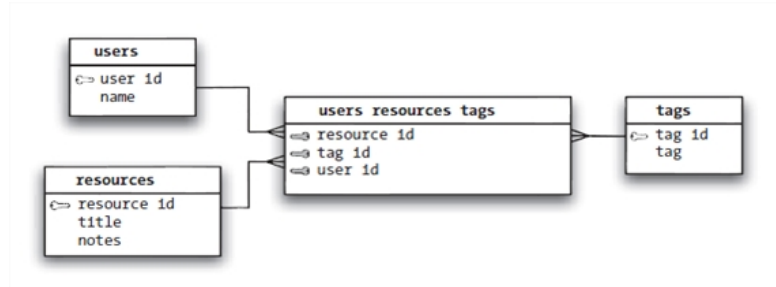


Figure 3: The data model for a collaborative tagging system [1]

There is a slight difference in the tables for the collaborative tagging system. The resources\_tags table is dropped to be replaced by a new intermediary table called the users\_resources\_tags table. This table allows multiple users to tag the same resource. It stores the resource id, tag id, and user id every time a resource is tagged. Thus, a resource with four tags will have four entries in the users\_resources\_tags table. This table also holds the key to a working collaborative tagging system as it makes folksonomies possible, as with the other features usually associated with collaborative tagging (e.g. pivot browsing which is using the user, tag, or resource as pivots to explore a system). [1]

### C. Recommender Systems

Recommender systems are systems whose goal is to generate meaningful recommendations to a collection of users for items or products that might interest them. Some examples of real-world industry-strength recommendation systems are that of the book recommendations in Amazon, or movie suggestions in Netflix. Such recommendation engines were designed according to the domain and the characteristics of data available. For example, Netflix users rate movies by using a scale of 1(disliked) to 5(liked). Such a data source records the quality of interaction between users and products. Recommender systems could differ in the way they



analyze these data sources in order to develop notions of affinity between users and products. These notions of affinity can then be used to identify well-matched pairs (user-product.) [23]

### **C..1 Collaborative Filtering**

Collaborative filtering (CF) is one of the technologies developed for recommender systems and is considered the most popular and efficient. [24] In a CF system, recommendation seekers are asked to express their preferences by rating a dozen or two items. Immediately a CF system merges the roles of the recommendation seeker and preference provider as the seekers will be able to generate recommendations through their own preferences. These systems focus on algorithms for matching people based on their preferences and weighting the interests of people with similar taste to produce a recommendation. [21]

CF approaches can be grouped into two categories: memory-based and model-based. Memory-based CF uses a similarity measure between pairs of users to build a prediction, and typically through weighted average. The similarity measure chosen is used to determine the accuracy of the prediction and numerous alternatives have already been studied. In general, schemes like these that rely on similarities across users cannot be pre-computed for fast online queries. Another issue in the memory-based schemes is that a minimum number of users who entered a minimum number of ratings may be required. [25] Model-based techniques generate recommendations by estimating parameters of statistical models for user ratings. [23] Model-based algorithms empirically may achieve higher prediction accuracy than memory-based methods. Meanwhile, memory-based algorithms provide a concise and intuitive justification for the computed recommendations. [24]

## C..2 Weighted Slope One Method

The weighted slope one algorithm is part of the slope one algorithms under the memory-based item-to-item algorithms for collaborative filtering. It is said to be easy to implement and maintain, updateable on the fly, not demanding from new users, efficient at query time, and reasonably accurate. The said characteristics are said to be desirable in a recommender system. [14][25]

**Notation:** The user is represented by  $u$  or  $v$ , and the item as  $i$  or  $j$ . The rating of a user  $u$  for the item  $i$  is expressed as  $r_{ui}$ , and the corresponding prediction is denoted as  $p_{ui}$ . The  $(u,i)$  pairs for which the ratings are known are stored in the set  $K$ . The set of all items rated by the user  $u$  is  $S(u)$ .  $|S(u)|$  is the number of items in the set. The deviation of item  $i$  with respect to item  $j$  is denoted by  $dev_{ij}$ . Without any special instructions,  $dev_{ji}$  will be the negative of  $dev_{ij}$ . [24]

$$P_{ui} = \frac{\sum_{j \in S(u) - \{i\}} (r_{uj} + dev_{ij}) * c_{ij}}{\sum_{j \in S(u) - \{i\}} c_{ij}}$$

Figure 4: Weighted Slope One Algorithm Equation

The weighted slope one algorithm is based on the slope one algorithm and differs by taking into consideration the number of ratings observed in the items denoted by  $C_{ij}$ . This means, the bigger the  $C_{ij}$  the more reliable the corresponding prediction. [25] In the above approach, we take the average of the predictions calculated from each individual items to generate a final prediction (regarding  $r_{uj} + dev_{ij}$  as a prediction for rating  $r_{ui}$  given  $r_{uj}$ ). The slope one algorithm is based on the intuitive principle of a popular differential (or deviation) between items for users.

For a simple example to show the deviation ( $dev_{ij}$ ), Table 2 shows ratings by

user A and B for items I and J. By the slope one approach, we compute the deviation of item J with respect to item I from user A:  $dev_{ji} = 3 - 2 = 1$ . Then we can infer the rating of user B for item J:  $4 + dev_{ji} = 5$ .

	Item I	Item J
User A	2	3
User B	4	? = 4 + (3 - 2)

Table 2: Example illustrating slope one approach

Formally, the slope one algorithm exploits the predictor of the form  $f(x) = x + b$  to indicate the relation of the item pair, where  $b$  is a constant deviation and  $x$  is a variable representing rating value. So the formal definition of  $dev_{ij}$  can be expressed as: Given two items  $i$  and  $j$  ( $i \neq j$ ), the deviation of item  $i$  respect to item  $j$  is:

$$dev_{ij} = \frac{\sum_{u \in S_{ij}} (r_{ui} - r_{uj})}{|S_{ij}|}$$

In this expression,  $S_{ij}$  denotes the set of users who both rated item  $i$  and item  $j$ , and  $|S_{ij}|$  is the number of users in the set. All deviations construct a deviation matrix which is a skew-symmetric matrix.

With the deviation matrix constructed, it is easy to make predictions by using the slope one algorithm.  $S\{u\}-\{i\}$  represents all items rated by user  $u$  except item  $i$ . [24]

## D. OSClass

OSClass [26] is a PHP script that allows for the creation of a Classified Ads website. Its structure can contain general ads which can be specialized through modifying

modules (e.g. cars, jobs, real estate, dating). It has multi-language support, and has a number of customization options available including currencies, and date-time formats. It is also integrated with TinyMCE for its WYSIWG editor which allows rich text formatting. OSClass is widely extensible through many plugins available, including Google Analytics, Google Maps, and Time Elapsed. In OSClass, defining an unlimited number of categories and sub-categories is possible.

## **E. Database Management Systems (DBMS)**

A DBMS is a system that enables the search and retrieval of information from a database. It controls how the data are stored and organized while addressing problems such as data integrity and security. It is used by an application that sends a request to which it responds by instructing the operating system to transfer the appropriate data.

A Relational Database Management System (RDBMS) is a DBMS that organizes its data into a series of tables which might be related by common fields. Its structure is made up of database tables, fields, and records. RDBMS also allows manipulation of data in the database tables with the use of relational operators. [27]

One of the most popular open source RDBMS products is MySQL. [28]

## IV. Design and Implementation

### A. Context Diagram

The Collaborative Tagging and Rating-based Collaborative Filtering in a Classified Ads System (Collab-Ads) is a system that supports three types of users: the unregistered user, registered user, and the system administrator. Figure 5 shows the context diagram for the system.

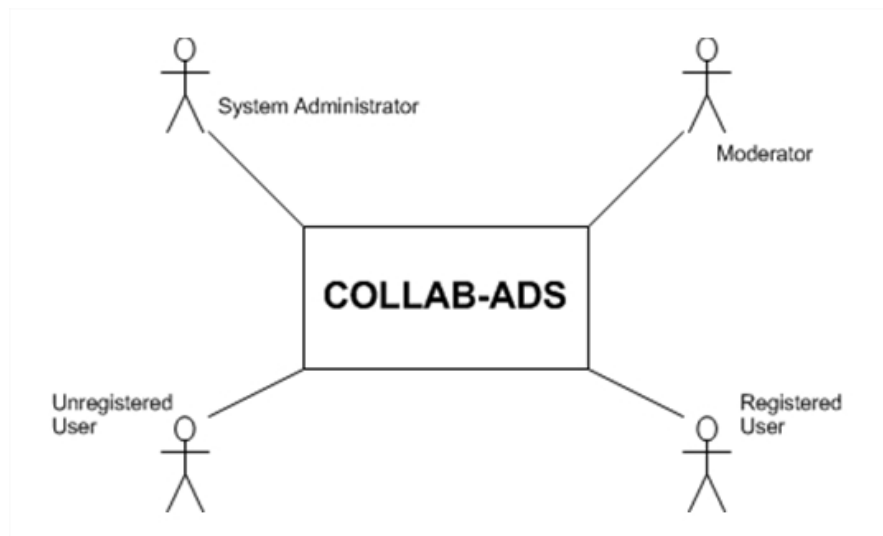


Figure 5: Context Diagram for the Collab-Ads system

## B. Use Case Diagram

### B.1 Top Level Use Case Diagram



Figure 6: Top Level Use Case Diagram of Collab-Ads

### B.2 Register Account

The following figure (Fig. 7) illustrates the registration process of an unregistered user. The unregistered user is asked to fill out a form asking for the users details, and to submit this while ensuring that the details marked required are completely filled out. The system then verifies the information before creating an account for the user in the system.

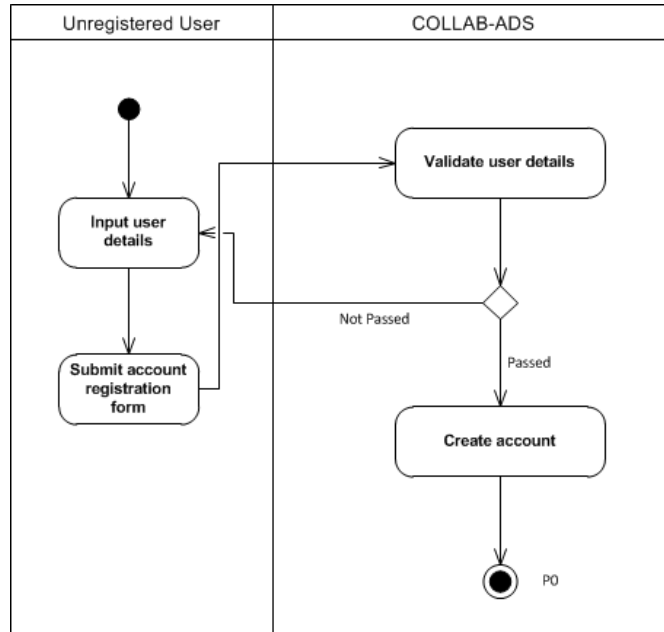


Figure 7: Activity Diagram for Registering an Account in Collab-Ads

### B..3 Search Listings

The system allows viewing and search of listings to registered or unregistered users. Figure 8 illustrates the activity diagram for the search feature. A user can input his/her desired keywords and then specify a category in which to search the listing before the search is performed. When the listings fitting the search criteria are displayed, the user can view each listing or edit the search criteria by specifying more fields. A user can also click the tags available from the displayed listings to browse listings under this tag alongside the category specified beforehand.

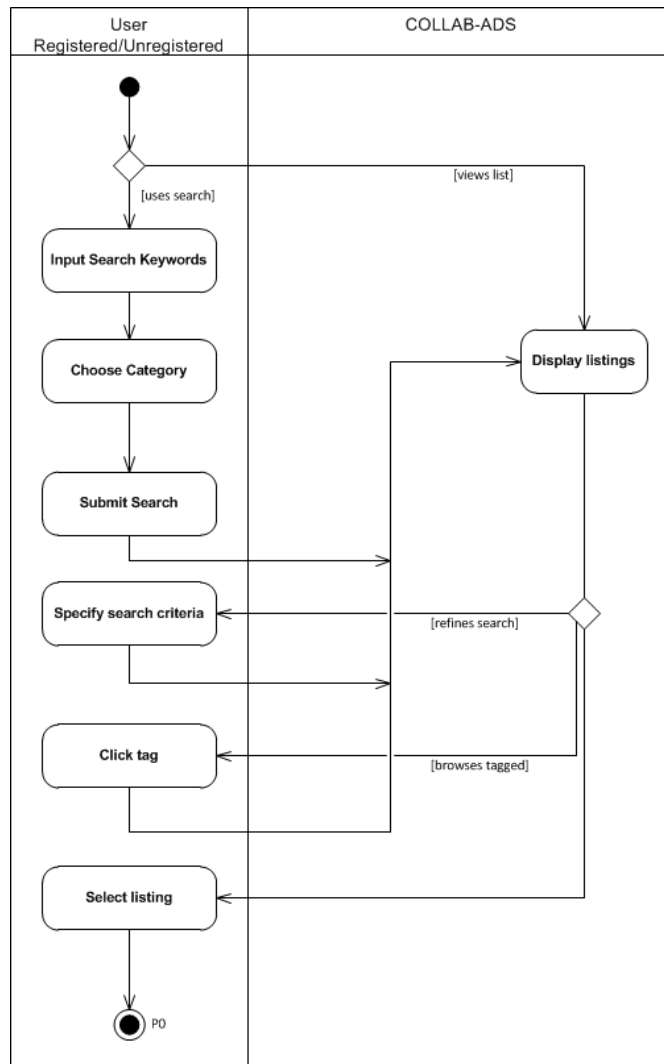


Figure 8: Activity Diagram for Using Search in Collab-Ads

#### B..4 Add Listing

A registered user can add a listing to the database. The figure below (Figure 9) illustrates the activity diagram for this. A listing can contain several details about the item which will be filled out by the author of the listing.



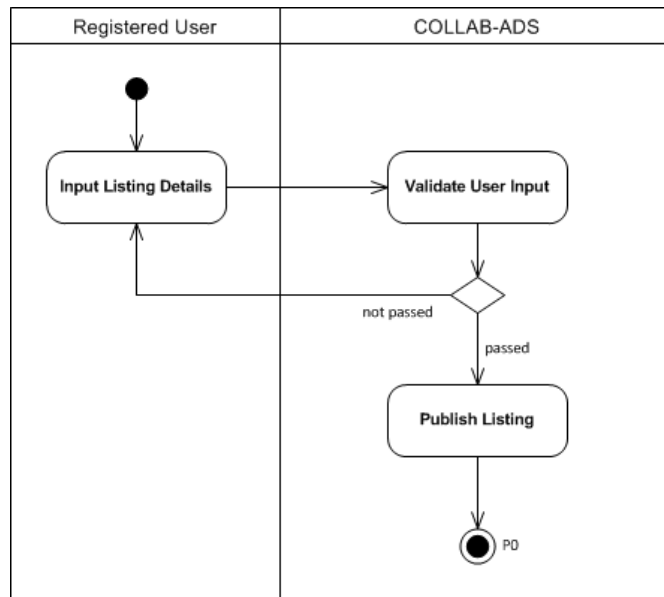


Figure 9: Activity Diagram for Adding a Listing in Collab-Ads

### B..5 Report Listing

A listing can be reported by a registered user and marked as offensive, misclassified, or spam. The reported listing will then be sent to the administrator who will do the review of the marked listing. The following figure (Figure 10) illustrates the activity diagram for this feature.

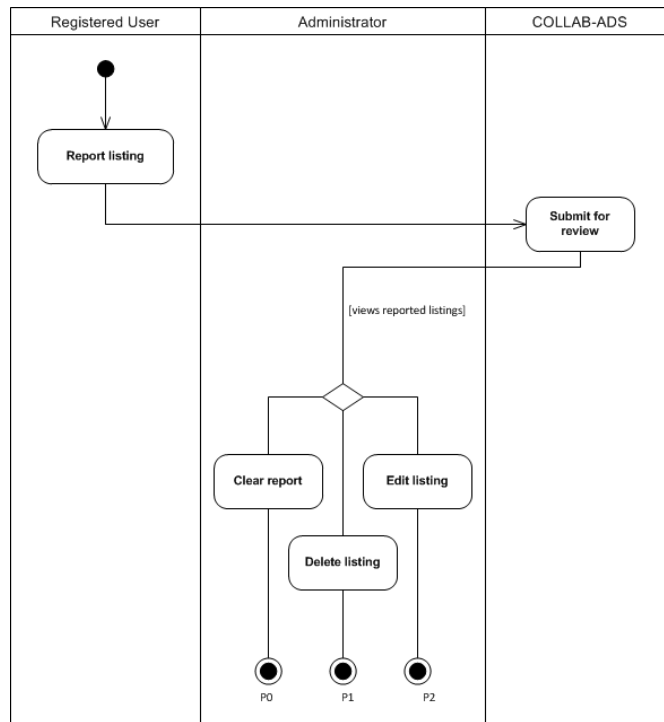


Figure 10: Activity Diagram for Reporting and Managing Reported Listings in Collab-Ads

## B..6 Contact Publisher

A logged-in, registered user can contact the publisher of a posted listing of another user. This user is asked to give an email (required) in which this user can be contacted, and also a contact number (optional) aside from the users name and message. Illustrated below is the activity diagram for contacting the publisher (Fig. 11).

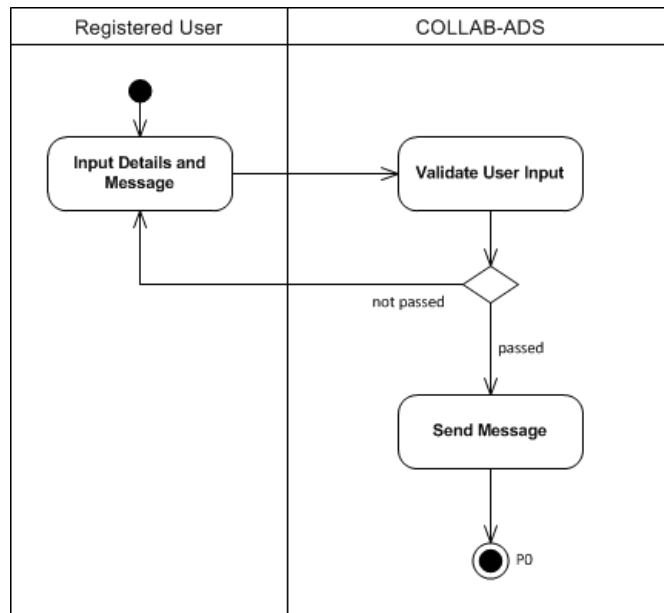


Figure 11: Activity Diagram for Contacting a Publisher in Collab-Ads

### B..7 Commenting on a Listing

A registered user can comment on a listing. Illustrated below (Fig. 12) is the activity diagram for when a user leaves a comment on a listing. After the comment is published, a user can delete his/her own comment(s).

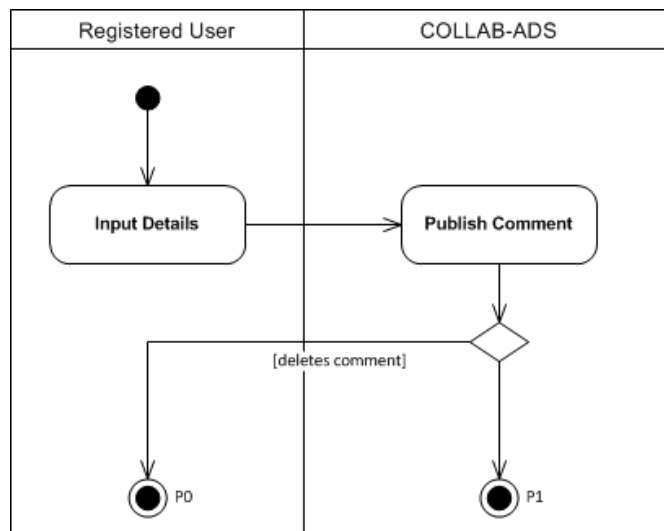


Figure 12: Activity Diagram for Commenting on a Listing in Collab-Ads

## B..8 Manage Owned Listings

A registered user can manage his/her own posted listings. The user has the option to edit and delete any listing this user posted. Upon editing a listing, the user can add or modify details in the listing. Deleting removes the listing permanently from the system. Figure 13 shows the activity diagram for this.

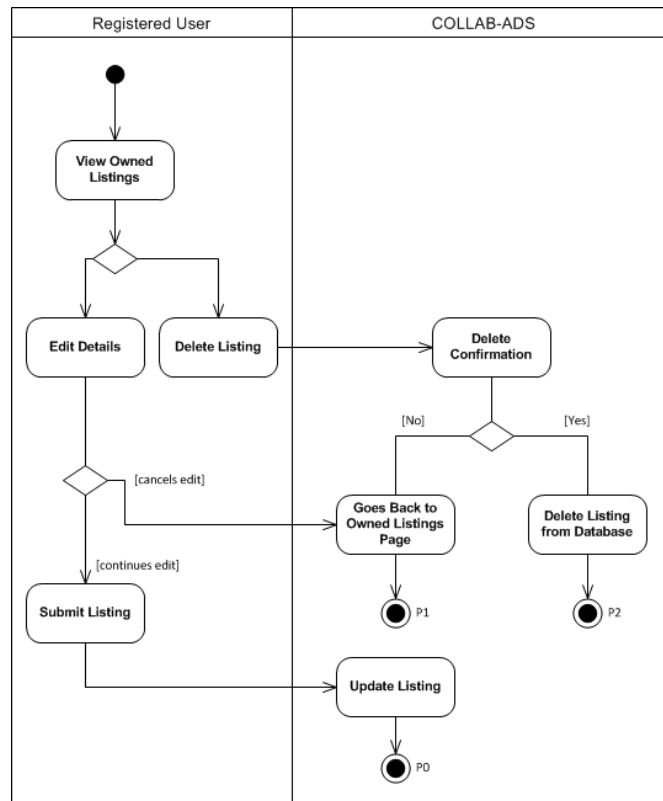


Figure 13: Activity Diagram for Managing a Users Own Listings in Collab-Ads

## B..9 Rate Listing

A registered user can rate an ad. Ratings are represented as stars with a rating value of 1 (lowest) to 5 (highest). Figure 14 below illustrates the activity diagram of this feature.

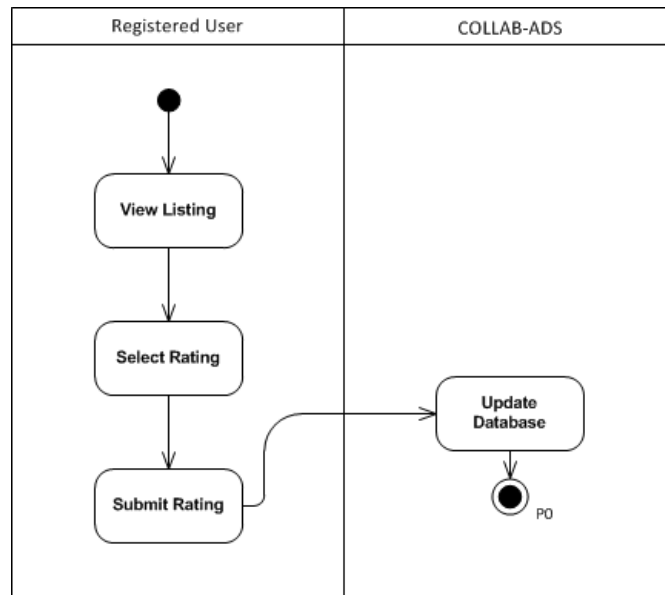


Figure 14: Activity Diagram for Rating a Listing in Collab-Ads

### B..10 Manage Tags

Any registered user can manage tags of any listing. Users have the option to add or remove any tag they may or may not have added in a listing.

#### a. Add New Tags

Adding a new tag involves inputting a tag label by the user. The system will then normalize this tag for display and storage before it adds the tag to the listing. Figure 15 below illustrates the activity diagram for this.

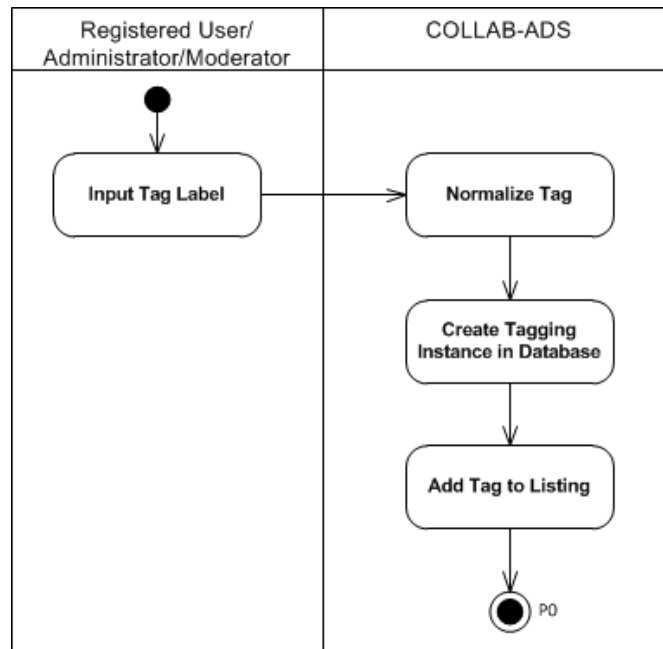


Figure 15: Activity Diagram for Adding Tag to a Listing in Collab-Ads

b. Edit Tags

Editing the tags in a listing involves giving the user the option to add or delete tags. Adding a tag is the same as the process described above. In deleting a tag, the user would have to select which tags he/she wants to delete and clicking a confirmation to submit the changes to the system. The activity diagram for editing tags is shown in Figure 16 below.

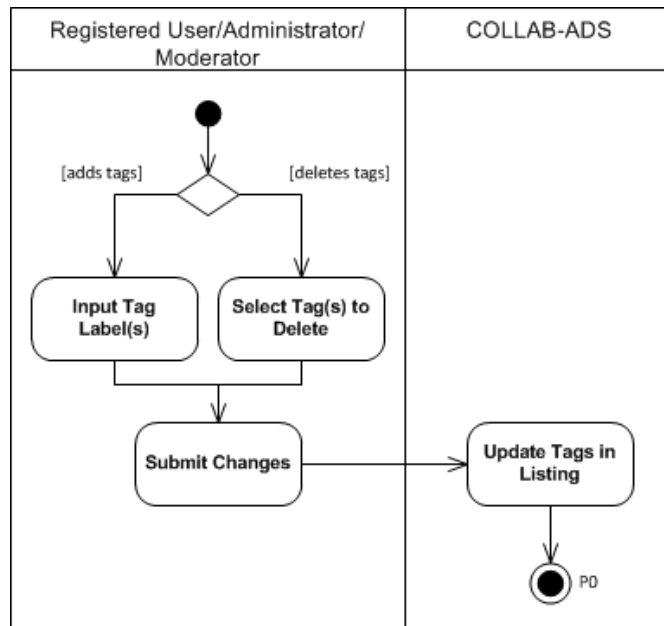


Figure 16: Activity Diagram for Editing Tags in a Listing in Collab-Ads

### B..11 Manage Listings

#### a. Add Listing

The figure below (Fig.17) illustrates the activity diagram for adding a listing through the administrator account. Administrator accounts, or moderator accounts can add listings.

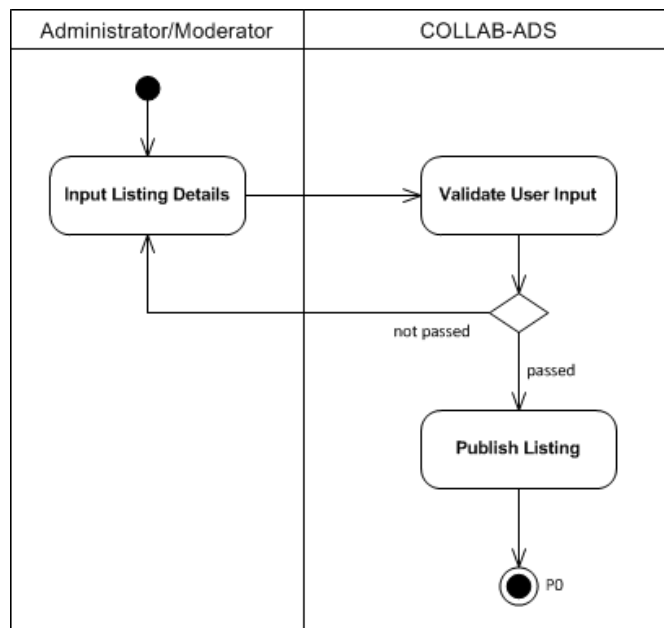


Figure 17: Administrator Activity Diagram for Adding a Listing

### b. Edit Listing

Administrators and moderators can also edit listings. Figure 18 shows the activity diagram for this.

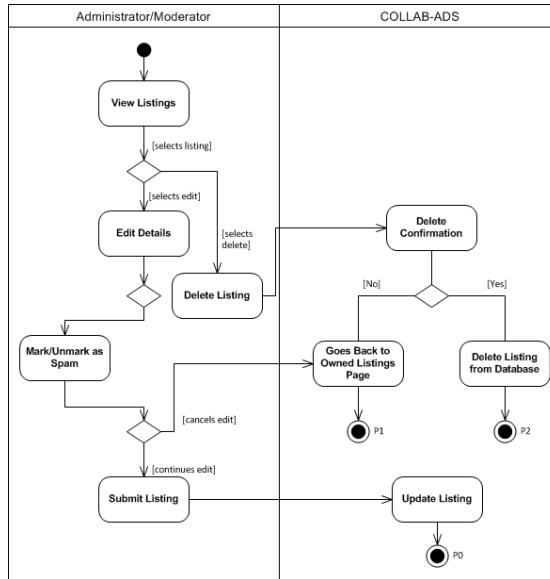


Figure 18: Activity diagram for editing a listing by Administrator or Moderator

## B.12 Manage Users

Users can be managed only by administrator accounts. An administrator can add a user or edit an existing users info, and change their account privileges.

### a. Add User

The activity diagram below (Figure 19) illustrates how an administrator can add a user account in the system.



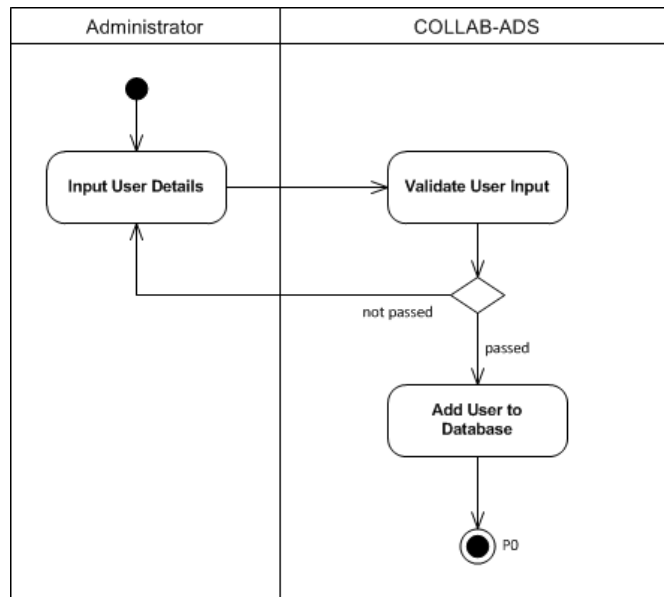


Figure 19: Activity diagram for adding a user to the system by the administrator

b. Edit User

An administrator can block/deactivate or delete a users account. The following diagram illustrates this. (Figure 20)

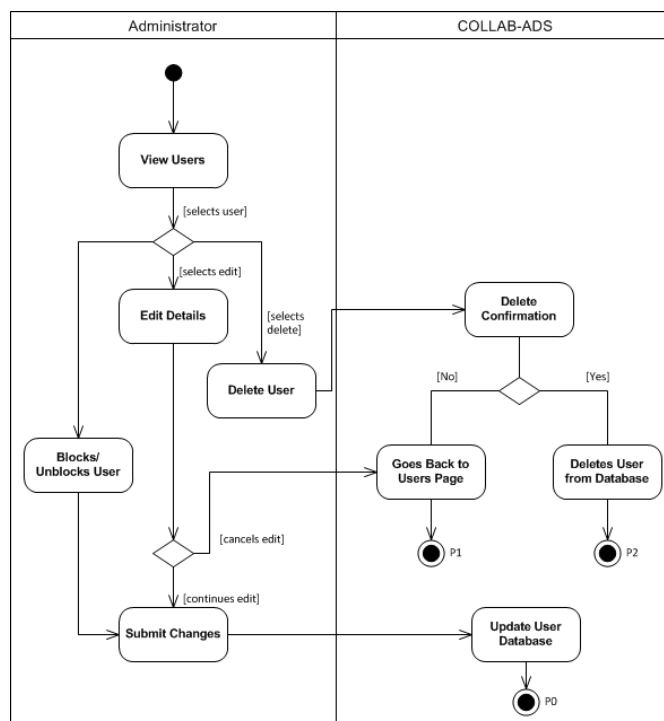


Figure 20: Activity diagram for editing a user account

## B..13 Manage Administrators

### a. Add Admin/Moderator

The following activity diagram (Figure 21) illustrates how an administrator can add other administrator accounts and moderator accounts.

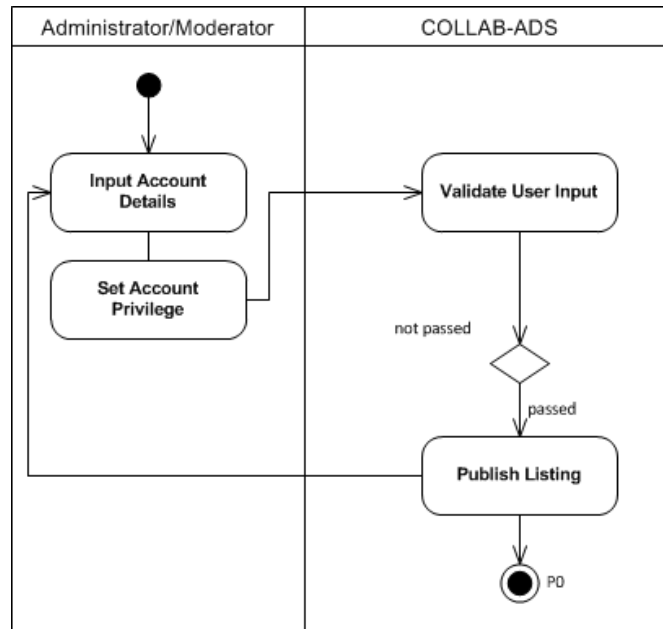


Figure 21: Activity diagram for adding an administrator/moderator account

### b. Edit Admin/Moderator

Editing administrator or moderator accounts can also be done by an administrator. The following figure (figure 22) illustrates how this is done.

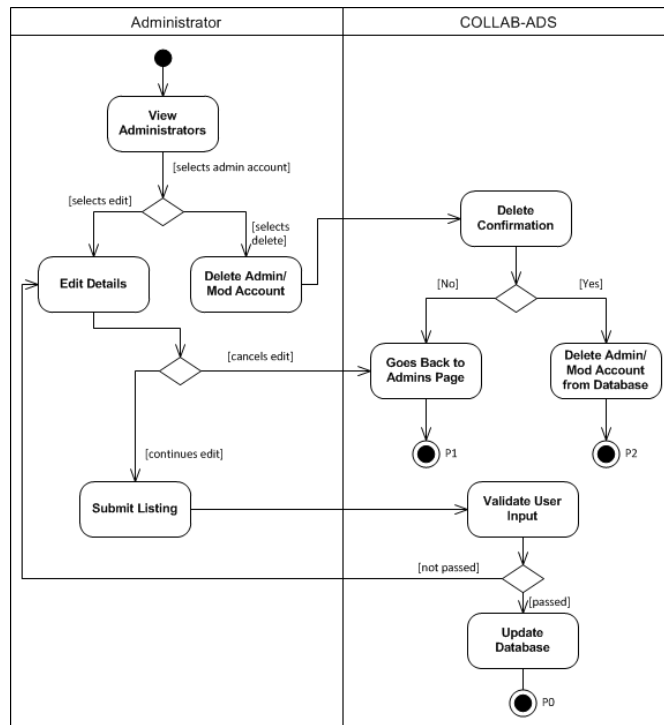


Figure 22: Activity diagram for editing an admin/moderator account

## B..14 Manage Categories

### a. Add Category

An administrator can add categories for listings in the system. This is shown by the following activity diagram. (Figure 23)

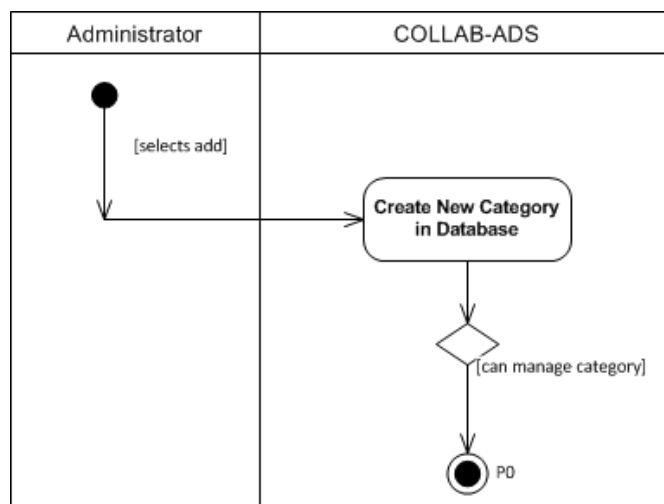


Figure 23: Activity diagram for adding categories

### b. Edit Category

Only administrator account can edit categories of listings in the system. The following activity diagram (figure 24) shows how this is done.

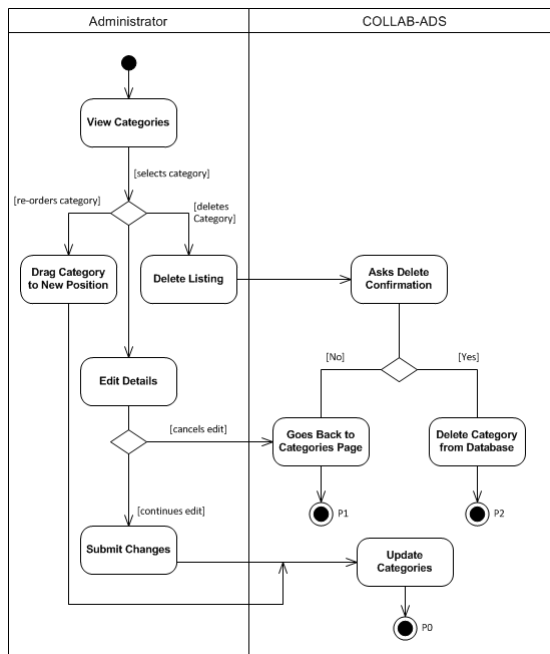


Figure 24: Activity diagram for editing a category

### B..15 Manage Comments

An administrator or moderator account can edit or delete comments posted by users on listings. The activity diagram below (figure 25) illustrates this.

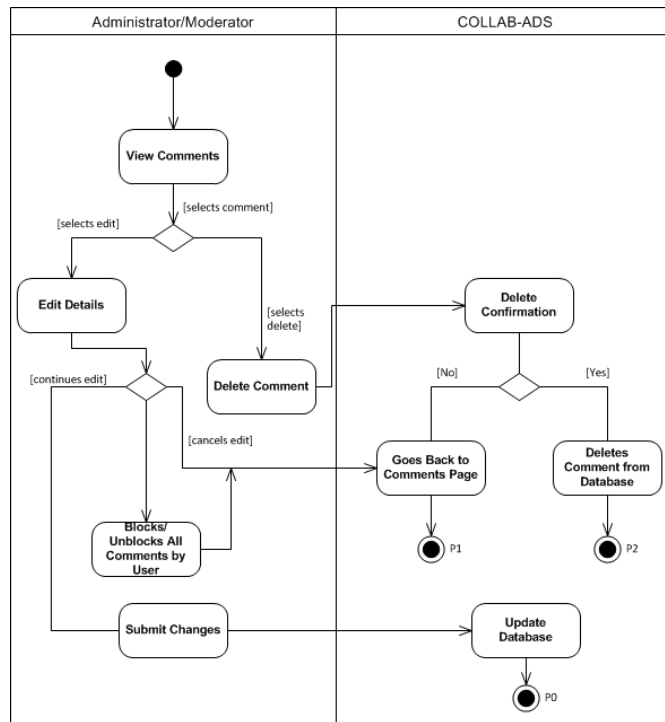


Figure 25: Activity diagram for managing comments



Table 4: t\_category contains list of all categories where listings are to be filed under

Column	Type	Description
<u>pk_i_id</u>	int(10)	Unique ID assigned to a category
fk_i_parent_id	int(10)	Identifies if category is a subcategory
i_expiration_days	int(3)	Days before category expires. Can be set by admin
i_position	int(2)	Order of category in hierarchy

Table 5: t\_category\_description holds name and description of each category

Column	Type	Description
<u>fk_i_category_id</u>	int(10)	ID of category
fk_c_locale_code	char(5)	ID of locale used
s_name	varchar(100)	Name assigned to category
s_description	text	Description assigned to category
s_slug	varchar(100)	Normalized version of name of category

Table 6: t\_category\_stats holds basic statistics on listings under each category

Column	Type	Description
<u>fk_i_category_id</u>	int(10)	ID of category
i_num_items	int(10)	Number of listings under the category

Table 7: t\_city contains list of all cities where listings are located

Column	Type	Description
<u>pk_i_id</u>	int(10)	Unique ID assigned to a city
fk_i_region_id	int(10)	ID of region city is under
s_name	varchar(60)	Name of city
fk_c_country_code	char(2)	ID of country city is under
b_active	tinyint(1)	Flag for enabling/disabling city

Table 8: t\_city\_area contains list of all areas inside a particular city where listings are located

Column	Type	Description
<u>pk_i_id</u>	int(10)	Unique ID assigned to a city area (narrowed down location)
fk_i_city_id	int(10)	ID of city the area is under
s_name	varchar(255)	Name of area

Table 9: t\_city\_stats holds basic statistics on listings in a city

Column	Type	Description
<u>fk_i_city_id</u>	int(10)	ID of city
i_num_items	int(10)	Number of listings in a particular city

Table 10: t\_country contains list of all countries made available to listings

Column	Type	Description
<u>pk_c_code</u>	char(2)	Unique ID assigned to a country
s_name	varchar(80)	Name of country

Table 11: t\_country\_stats holds basic statistics on listings for each country

Column	Type	Description
<u>fk_c_country_code</u>	char(2)	ID of country
i_num_items	int(10)	Number of listings under particular country

Table 12: t\_dev contains slope one values of rating pairs to be used for computing predictions

Column	Type	Description
<u>fk_i_item_id_1</u>	int(10)	ID of 1st item in pair of items rated
<u>fk_i_item_id_2</u>	int(10)	ID of 2nd item in pair of items rated
i_count	int(11)	Stores count when processing differences
i_sum	int(11)	Stores sum of rating differences



Table 13: t\_item contains list of all listings posted in the website

<b>Column</b>	<b>Type</b>	<b>Description</b>
<u>pk_i_id</u>	int(10)	Unique ID assigned to a listing
fk_i_user_id	int(10)	ID of user who created listing
fk_i_category_id	int(10)	ID of category listing is under
dt_pub_date	datetime	Date the listing is published
dt_mod_date	datetime	Date the listing is last modified
f_price	float	Latest price assigned to item in listing
i_price	bigint(20)	Price of item in listing
fk_c_currency_code	char(3)	ID of currency used in listing
s_contact_name	varchar(100)	Name of listing publisher
s_contact_email	varchar(140)	E-mail of listing publisher
b_premium	tinyint(1)	Flag for premium listing
b_enabled	tinyint(1)	Flag for enabled listing
b_active	tinyint(1)	Flag for active listing
b_spam	tinyint(1)	Flag for spam marks
b_show_email	tinyint(1)	Flag for hiding/showing email
dt_expiration	datetime	Date listing expires

Table 14: t\_item\_comment contains list of all comments made on a particular listing

Column	Type	Description
<u>pk_i_id</u>	int(10)	Unique ID assigned to a comment in listing
fk_i_item_id	int(10)	ID of listing
dt_pub_date	datetime	Date comment is published
s_title	varchar(200)	Title of comment
s_author_name	varchar(100)	Name of comment author
s_author_email	varchar(100)	E-mail address of comment author
s_body	text	Holds message or body of the comment
b_enabled	tinyint(1)	Flag for a user accounts ability to comment
b_active	tinyint(1)	Flag for a users activated account
b_spam	tinyint(1)	Flag for a spam-marked comment
fk_i_user_id	int(10)	ID of user who made the comment

Table 15: t\_item\_description holds the title and content of a listing

Column	Type	Description
<u>fk_i_item_id</u>	int(10)	ID of listing
fk_c_locale_code	char(5)	ID of locale used
s_title	varchar(100)	Title of listing
s_description	mediumtext	Description of listing

Table 16: t\_item\_location holds the location of a listing

Column	Type	Description
<u>fk_i_item_id</u>	int(10)	ID of listing
fk_c_country_code	char(2)	ID of country assigned where listing is under
s_country	varchar(40)	Country name listing is under
s_address	varchar(100)	Address of listing
s_zip	varchar(15)	Zip code listing is under
fk_i_region_id	int(10)	ID of region listing is under
s_region	varchar(100)	Region name listing is under
fk_i_city_id	int(10)	ID of city assigned where listing is under
s_city	varchar(100)	Name of city listing is under
fk_i_city_area_id	int(10)	ID of city area assigned where listing is under
s_city_area	varchar(200)	Name of city area

Table 17: t\_latest\_searches contains latest search entry

Column	Type	Description
d_date	datetime	Date of latest search
s_search	varchar(255)	Search entry

Table 18: t\_locale contains list of all available locales in system set by administrator

Column	Type	Description
<u>pk_c_code</u>	char(5)	Unique ID assigned to a locale
s_name	varchar(100)	Name of locale
s_short_name	varchar(40)	Abbreviation of locale
s_description	varchar(100)	Description of locale
s_version	varchar(20)	Version of locale
s_author_name	varchar(100)	Author of locales name
s_author_url	varchar(100)	Author of locales URL
s_currency_format	varchar(50)	Currency symbol
s_dec_point	varchar(2)	Decimal point format
s_thousands_sep	varchar(2)	Thousands format
i_num_dec	tinyint(4)	Number of digits before decimal point
s_date_format	varchar(20)	Date format
b_enabled	tinyint(1)	Flag for enabled/disabled locale

Table 19: t\_rated\_items contains rating value made by user for a certain item

Column	Type	Description
<u>fk_i_user_id</u>	int(10)	ID of user who gave the rating
<u>fk_i_item_id</u>	int(10)	ID of item being rated
i_value	decimal(14,4)	Value a user rated an item with
t_datetimestamp	timestamp	Date when item was rated

Table 20: t\_region contains list of all regions to file listings under

Column	Type	Description
<u>pk_i_id</u>	int(10)	Unique identifier for a region
fk_c_country_code	char(2)	ID of country region is under
s_name	varchar(60)	Name of region
b_active	tinyint(1)	Flag for enabled/disabled region

Table 21: t\_region\_stats contains basic statistics of listings under a certain region

Column	Type	Description
<u>fk_i_region_id</u>	int(10)	ID of region
i_num_items	int(10)	Number of listings under a region

Table 22: t\_tagger\_tags holds information about a tag

Column	Type	Description
<u>pk_tag_id</u>	int(10)	Unique identifier assigned to a tag
tag_label	varchar(50)	Normalized form of raw tag used as label or keyword to tag
raw_tag	varchar(50)	Raw tag form inputted by user

Table 23: t\_tagger\_tagged\_objects holds information about a tagging

Column	Type	Description
<u>tag_id</u>	int(10)	ID of the tag used in a tagging
<u>tagger_id</u>	int(10)	Creator of a tagging
<u>tagged_resource</u>	int(10)	Item tagged in a tagging
tagged_on	datetime	Date a tagging was last modified

Table 24: t\_user contains list of all users who registered in the website

Column	Type	Description
<u>pk_i_id</u>	int(10)	Unique ID assigned to a user
dt_reg_date	datetime	Date the user registered
dt_mod_date	datetime	Date user modified account
s_name	varchar(100)	Name of user
s_password	varchar(40)	Password of user
s_email	varchar(100)	E-mail address used by user to register
s_website	varchar(100)	Website of user
s_phone_land	varchar(45)	Landline number of user
s_phone_mobile	varchar(45)	Mobile phone number of user
b_enabled	tinyint(1)	Flag for blocked/unblocked account
b_active	tinyint(1)	Flag for active account
fk_c_country_code	char(2)	ID of country
s_country	varchar(40)	Country name of user
s_address	varchar(100)	Address of user
s_zip	varchar(15)	Zip code of user
fk_i_region_id	int(10)	ID of region
s_region	varchar(100)	Region name of user
fk_i_city_id	int(10)	ID of city
s_city	varchar(100)	City name of user
fk_i_city_area_id	int(10)	ID of city area
s_city_area	varchar(200)	City area name of user
i_items	int(10)	Number of listings user published
i_comments	int(10)	Number of comments made by user
dt_access_date	datetime	Last login date
s_access_ip	varchar(15)	IP address used in last access

Table 25: t\_user\_description holds description of a user in their user account

Column	Type	Description
<u>fk_i_user_id</u>	int(10)	ID of user account
<u>fk_c_locale_code</u>	char(5)	Locale used by use
s_info	text	Description made by user for account

Table 26: t\_user\_email\_tmp holds data for when requesting an email change

Column	Type	Description
<u>fk_i_user_id</u>	int(10)	ID of user
s_new_email	varchar(100)	E-mail address to change over current one
dt_date	datetime	Date new E-mail was changed

## V. Technical Architecture

### 1. Server Requirements

- (a) Operating System: Windows 7, Linux
- (b) PHP 5.2 or higher
- (c) MySQL 5.0 or higher
- (d) SMTP Server

### 2. Client Requirements

- (a) Browser: Chrome 31.0 or higher, Mozilla Firefox 3.0 or higher, Internet Explorer 8.0 or higher



## VI. Results

Figure 26 shows the login form of administrators and moderators. Logging in will take them to the dashboard where they can manage parts of the system. An administrator is able to see more controls for the system installation including customization of themes, plugins, date/time formatting, currencies, and location settings.

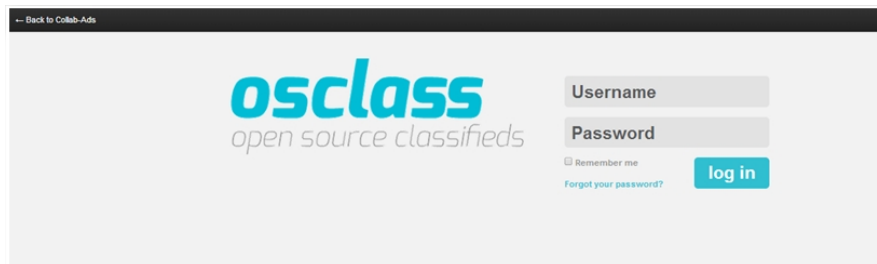


Figure 26: Administrator Login Form

Administrators and moderators of the system can manage users wherein they can permanently delete, block, or deactivate a user. The following figure (Figure 27) shows the page that allows user management.

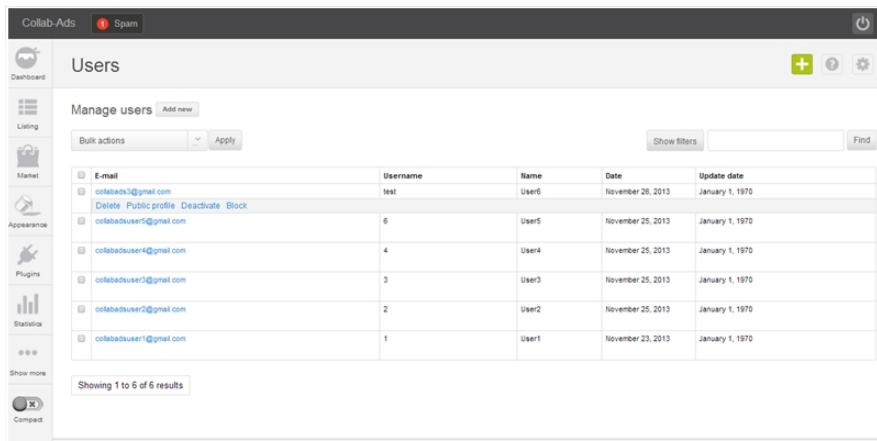


Figure 27: User Management Page

The figure below (Figure 28) shows where administrators can create and manage other administrator accounts. Aside from creating administrators, this page

is also where they can create moderator accounts. Creating an administrator/-moderator account requires an a name, a username, an email, and a password.

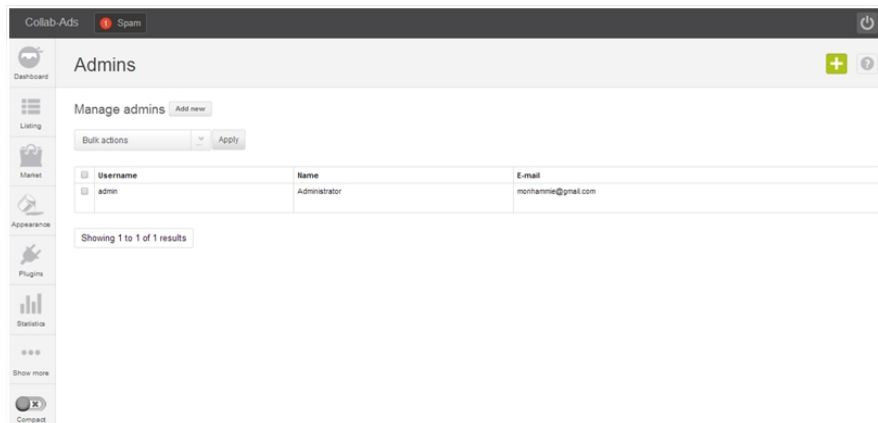


Figure 28: Administrator View: Admin Management Page

The system also allows administrators and moderators to manage the listings. Each listing can be permanently deleted, deactivated, blocked, marked as premium, and marked as spam. Marking a listing as premium promotes a listing. Premium ads appear more often in the site. Spam listings serve as marks for listings to be reviewed for deletion. They could be irrelevant, senseless, or they could contain something that does not adhere to the site rules. Deactivated listings, meanwhile, will not appear in the site. In case a listing is deleted, the system will send the owner of the listing an email notification about the removed listing. The following figure shows the page where listings are managed:

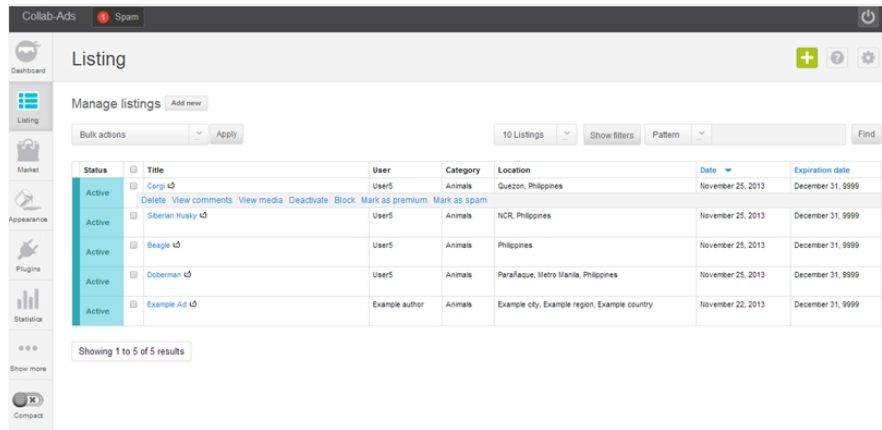


Figure 29: Administrator View: Listings Management Page

Part of keeping listings organized, users, administrators, and moderators can report or mark listings. Figure 30 shows the page where reported listings are listed. These listings will show how many times it has been reported as spam, misclassified, duplicated, expired, or offensive. Any reported listing is subject to review by the administrators or respective moderators for deletion. Also, administrators and moderators can clear reports on a listing in case they do not find any problems with it.

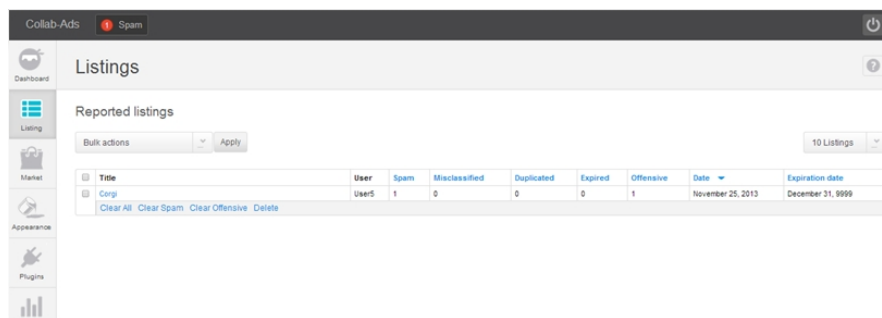


Figure 30: Administrator View: Reported Listings Page

Figure 31 below shows the page wherein plugins are managed. Plugins can be imported from the OSClass market, or the administrator can upload his/her own plugin. A plugin can be configured, disabled, installed or uninstalled. Configuring a plugin usually involves setting up the categories wherein the plugins features will take effect. Disabling a plugin will prevent the plugin from showing on the

site, and uninstalling a plugin removes the components of the plugin from the site and/or database.

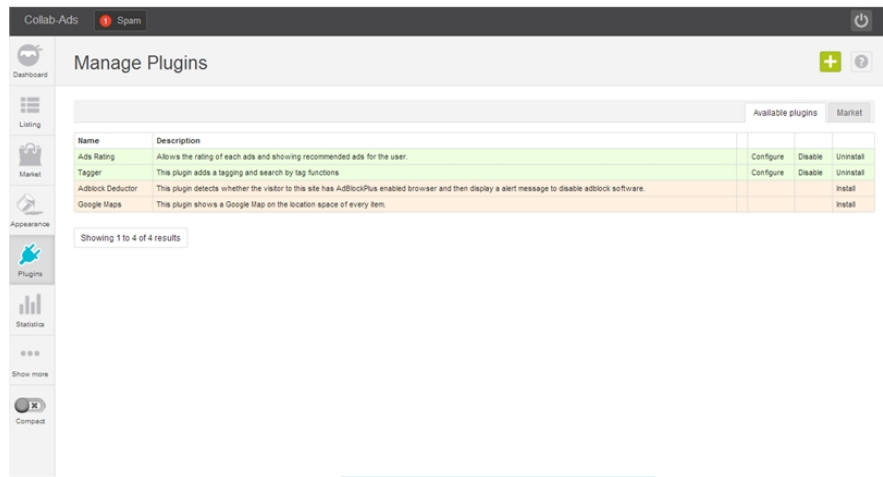


Figure 31: Administrator View: Plugins Page

Administrators can also manage comments in the site. The following figure (Figure 32) shows the page where this is done. In this page, all comments on every listing are shown including the date it was made, and the comments author. A comment can be deleted whenever deemed necessary by the administrator.

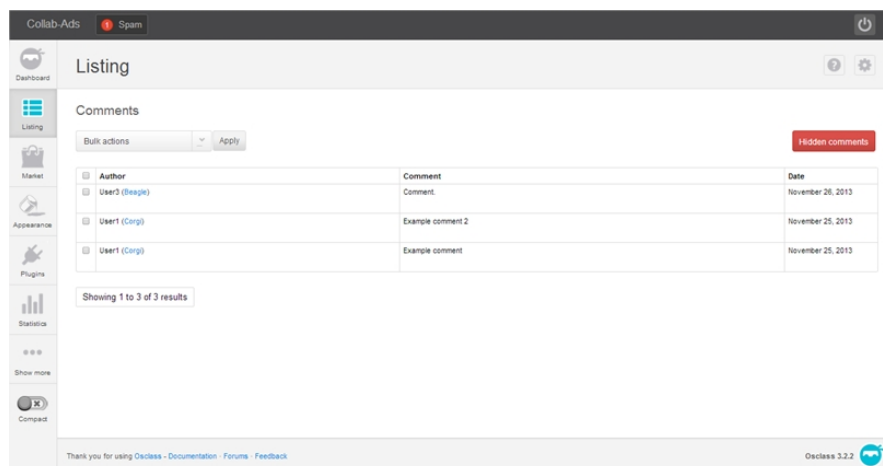


Figure 32: Administrator View: Comments Management Page

The homepage for the system is shown in the following figure (Figure 33). All categories and subcategories are shown including the number of listings in the

subcategories. The numbers of listings sorted by location are shown on a panel on the right while the bottom part of the homepage shows the latest listings created. A registered user can see a link to his/her account management page on the top right corner while an unregistered user can see a link that directs to the registration page instead. The link to posting an ad also appears on the top right, which will take a user to the listing creation form. This option is only available to registered users and unregistered users will see a link that will take them to the registration page.

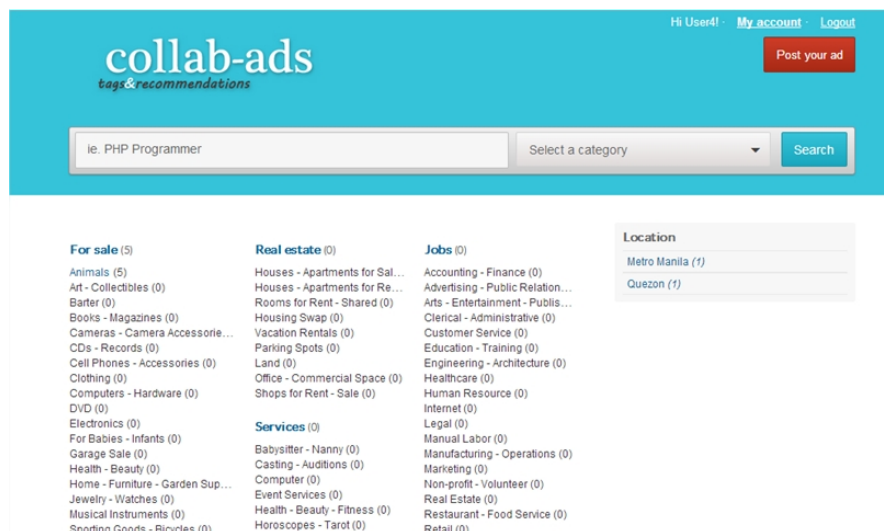


Figure 33: Registered User Homepage

The figure below (Figure 34) shows the entry form when publishing a new listing. A category, title, and description should be specified for the item. The publisher has an option to leave the price unspecified where a note for other users to contact the user about the price will appear on the listing. Other optional inputs for the listing include photos, location, and tags.

Colab-Ads - Publish a listing

## Publish a listing

### General Information

Category:

Title:

Description:

Price:

Php:

### Photos

Photos:  No file chosen

[Remove](#)

[Add new photo](#)

### Listing Location

Country:

Region:

City:

City Area:

Address:

Tags:


Colab-Ads is a website featured with collaborative tagging and collaborative filtering. 2012

Figure 34: Listing Publish Form

The figures below (Figure 35, 36, 37) show an example of a listing. Below the listings description are the tags and the five-star rating options. A registered user who has not yet rated has the option to give a rating once, while a user who has already rated the listing sees the average rating on the item. The right panel of a listing (Figure 35) contains the options for other registered users to mark the listing, and a contact form to the publisher.

**Siberian Husky** 22000.00 Php

Published date: November 25, 2013  
Location: NCR, Philippines



Mark as...  
**Contact publisher**  
 Name: User5  
 Your name:   
 Your e-mail address:   
 Phone number (optional):   
 Message:

description x

Tags :  
[cute](#) [dogs](#) [puppies](#)  
[Modify Tags](#)

Vote ☆☆☆☆☆ | Result ★★★★★ 3 votes

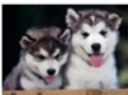
Figure 35: Registered User: Viewing a Listing

Just below the rating options, there is a sharing option available for all users which is used to share the listing via email. The recommended listings also appear next wherein the links and descriptions of the recommended listings are shown to the user.

**Useful information**


- Avoid scams by acting locally or paying with PayPal
- Never pay with Western Union, Moneygram or other anonymous payment services
- Don't buy or sell outside of your country. Don't accept cashier cheques from outside your country
- This site is never involved in any transaction, and does not handle payments, shipping, guarantee transactions, provide escrow services, or offer "buyer protection" or "seller certification"

**Recommended listings**



[Siberian Husky](#)  
Animals - (NCR) - November 25, 2013 **22000.00 Php**

description x

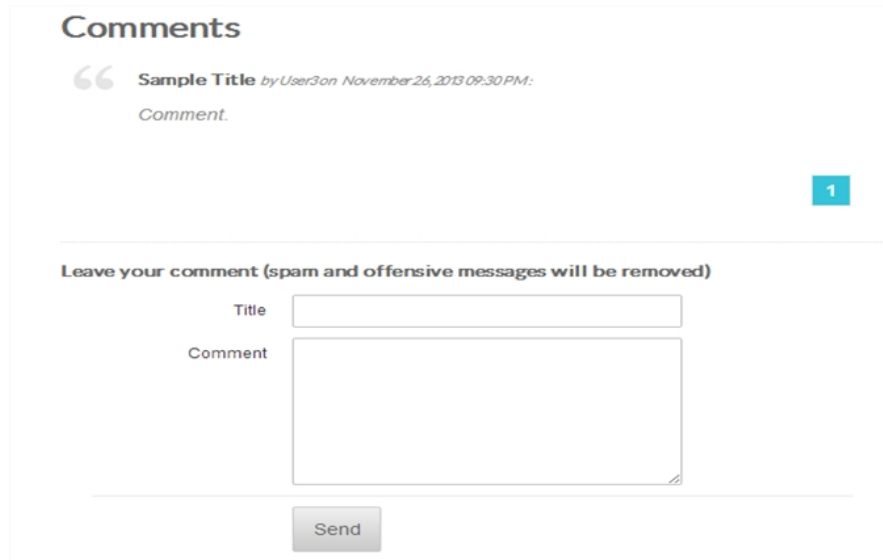


[Doberman](#)  
Animals - Parañaque (Metro Manila) - November 25, 2013 **20000.00 Php**

description

Figure 36: Registered User: Listing Recommendations

Another means to communicate with the publisher is by commenting on the comment form (Figure 37) at the bottom of the page.



The image shows a 'Comments' section. At the top, there is a header 'Comments'. Below it, a sample comment is displayed: 'Sample Title by User3 on November 26, 2013 09:30 PM:'. The comment text is 'Comment.'. To the right of the comment is a small blue square with the number '1'. Below the comment is a horizontal line. Underneath the line, there is a heading 'Leave your comment (spam and offensive messages will be removed)'. This is followed by a form with two input fields: 'Title' and 'Comment'. The 'Comment' field is a larger text area. Below the form is a 'Send' button.

Figure 37: Registered User: Listing Comment Section

To unregistered users, a listing is shown with fewer options. These users cannot rate a listing but can see the average votes it has received. They also are not able to mark a listing, or leave a comment.

The following figure (Figure 38) shows a publishers view of his/her own listing. An option to edit the item is shown, and the contact form is removed.



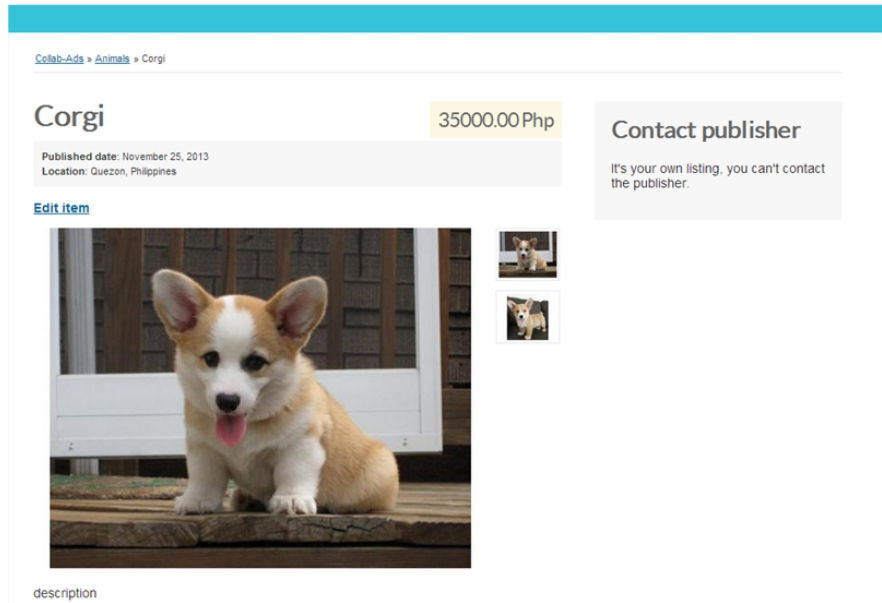


Figure 38: Registered User: Owner View of a Listing

A registered user can view the listings he/she has published shown in Figure 39 below. This can be accessed through the account page and allows the user to easily edit or delete his/her listings.

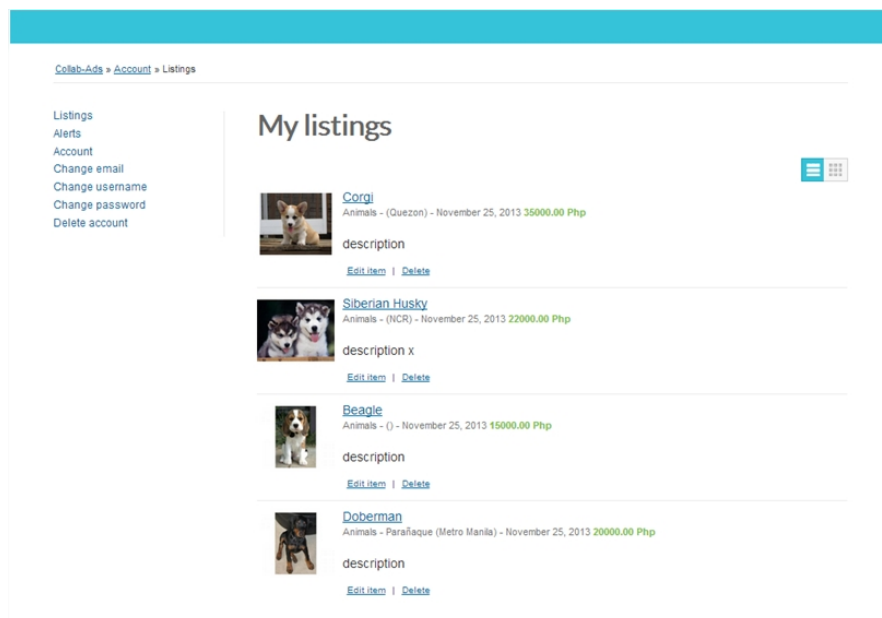
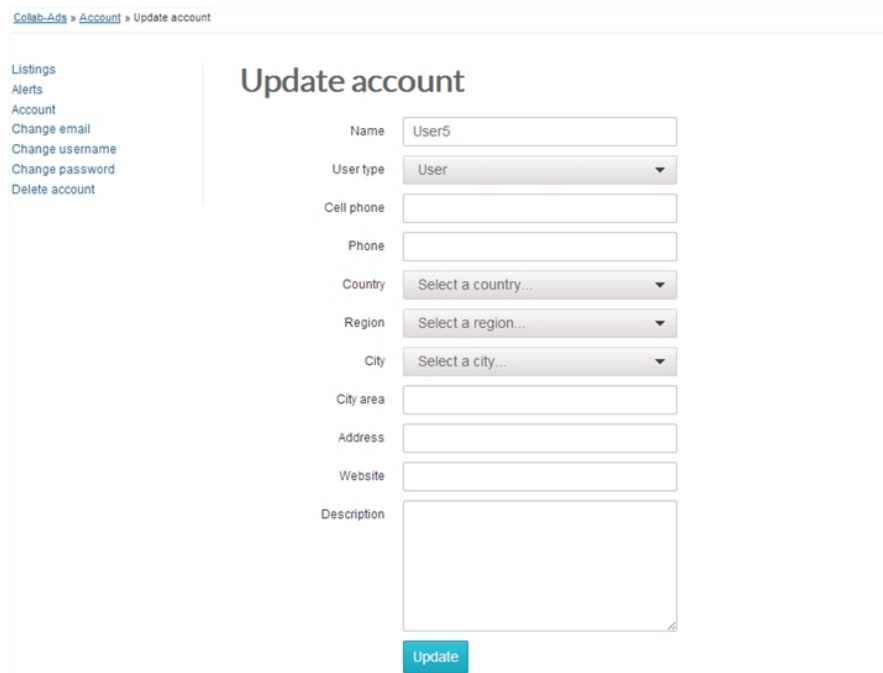


Figure 39: Registered User: View of Owned Listings

A registered user can also update his/her account information in the account

page (Figure 40). The user can specify his/her contact details, location, and website URL. There are also options that allow him/her to change his username, email, and password. Lastly, a user can delete his/her account which removes him/her from the system as well as the listings he/she owns.



The screenshot shows a web application interface for updating a user account. At the top left, there is a breadcrumb trail: [Collab-Ads](#) > [Account](#) > [Update account](#). On the left side, there is a vertical menu with the following items: Listings, Alerts, Account, Change email, Change username, Change password, and Delete account. The main content area is titled "Update account" and contains the following form fields:

- Name:
- User type:  (dropdown menu)
- Cell phone:
- Phone:
- Country:  (dropdown menu)
- Region:  (dropdown menu)
- City:  (dropdown menu)
- City area:
- Address:
- Website:
- Description:

At the bottom of the form, there is a blue "Update" button.

Figure 40: Registered User: Editing Account

## VII. Discussions

The Collab-Ads system is a classified ads system that has been integrated with two main features: collaborative tagging, and collaborative filtering. As a classified ads system, users were allowed to register for an account to be used to publish listings. A registered user can post a listing and specify an items price, location, description, category, and can attach images of the item. Registered users were also allowed to edit the information on their own listings. Aside from this, registered users were allowed to communicate with other publishers by commenting on the listings, and by contacting them privately through the contact publisher form. Also they were able to share a listing by sending its link to the email they provided. Furthermore, registered users were able to mark a listing as misclassified, spam, duplicate, or expired which was then to be reviewed by the administrator. Both registered and unregistered users were also allowed to search for a listing by its title.

The collaborative tagging and filtering features were installed into the system as plugins. Any registered user can collaboratively create, and manage tags on any item regardless of whether they were its publishers. Tags can be attached and modified to a listing in three ways: in the listing creation form, in the listing editing form, and while viewing a listing. To create the tags, the system normalized the tag input to identify if it is a new vocabulary not found in the tags table. If it was a new vocabulary, it was added to the tags table before a tagging table was created. The tagging table contained the information from three separate tables: the user who tagged the item, the item tagged, and the tag used. When a user deleted a tag from the tag form, only the entry from the tagging table was deleted; the tag used still remained kept in the tags table. Searching by tags was also allowed for registered and unregistered users.

As for the collaborative filtering feature, the system allowed registered users to

rate the listings based on how interesting they found an item. From these ratings, the system was able to compute predictions for ads they were yet to rate using the weighted slope one method. Hence, the system had a rating, (the predicted rating), for every unrated listing of a user. A prediction was computed essentially through differences in ratings between pairs of items. This was done to find the most similar items, and then later combined this information with a users ratings on those similar items to be able to get the predictions. The predicted ratings were then ranked from highest to lowest and then suggested to a user. Publishers of a listing were able to rate their own items. Since the recommendations were based on which are similar and how interesting they would be to that user, ratings that appear did not imply the popularity of an item rating his/her own items does not make his/her item popular instead the rating will be taken as a measure of similarity with other items.

Meanwhile, administrators of the system were able to manage accounts; this included being able to create, un/suspend, and delete accounts. After an account is deleted or suspended, an email regarding the accounts removal was also sent by the system to the owner. They are also allowed to manage ads which included being able to delete a listing. This also sends a notification email to the owner of the listing. Additionally, administrators were also able to create moderator accounts that were given the same ability as the administrator to delete users and listings, and moderate comments and media files. The main difference between an administrator and moderator was that administrators had control over the installation of the system.

Finally, unregistered users were able to register an account, and also view and search existing listings, which can be refined by categories.

## VIII. Conclusions

Collab-Ads provides some improvements for classified ads systems. Collaborative tagging allows users to annotate and categorize content. Through this, we have enabled the searchability of the items by being able to make access points from the textual descriptions added by the users. Thus, the system gives users an option to refine their searches through clicking on tags. The textual descriptions can also serve as something that would attract other users to an item. Aside from granting more ways to describe an item, the system also gives automated recommendations from ratings. The ratings give a positive effect on items as it allows users to know which products are interesting based on its average ratings. Meanwhile, the recommendations help users find items which are similar to items they find interesting. Also, without requiring search queries, users get their own personalized content. Furthermore, with the administrators being capable of managing listings, especially ones reported by users, they can maintain the system and keep it clean of irrelevant listings.

## IX. Recommendations

The Collab-Ads system was able to offer two main features collaborative tagging and collaborative filtering on a classified ads system, particularly the one built using the script, OSClass. The script is still far from its mature state and has a lot of room for improvements. Being a basic classified ads system, there are a lot of other features which can be implemented, such as location mapping for items. More work on its current features can be done, for example, on its notification system which could be integrated into the system itself. Also, added features may include: generating popular searches, and spam filtering mechanisms for comments and listings.

While the systems tagging feature offers enhancements on the searchability of its items, and also helps attract other users to a publishers items, this feature can still be improved in many ways. Firstly, searching by multiple tags is a feature that can increase accuracy on searches. Notifications on taggings made can also help with the awareness of listing owners on tags they might not want on their items. Information on who made a tagging can also help users identify potential abusers and spammers. Moreover, CAPTCHA can also be implemented to prevent these events. As for recommendations, it can potentially be generated by basing on similarities on tag information aside from item similarities. Other recommendations that might give additional convenience to users with their searches include information which can be based on views, or comments.

## X. Bibliography

- [1] G. Smith, *Tagging: People-powered Metadata for the Social Web (Voices That Matter)*. New Riders Press, 1 ed., 2008.
- [2] J. Surowiecki, “2006 young innovator: Joshua schachter, 32.” Retrieved: 2012-07-21.
- [3] J. Trant, “Studying social tagging and folksonomy: A review and framework,” *Journal of Digital Information*, vol. 10, no. 1, 2009.
- [4] S. Golder and B. A. Huberman, “The structure of collaborative tagging systems,” *Journal of Information Science*, vol. 32, p. 198208, April 2006. cite arxiv:cs/0508082.
- [5] C. Trattner, C. Körner, and D. Helic, “Enhancing the navigability of social tagging systems with tag taxonomies,” in *Proceedings of the 11th International Conference on Knowledge Management and Knowledge Technologies, i-KNOW '11*, (New York, NY, USA), pp. 18:1–18:8, ACM, 2011.
- [6] D. Lemire and S. McGrath, “Implementing a rating-based item-to-item recommender system in php/sql,” Tech. Rep. D-01, January 2005.
- [7] “Online classified report.” Retrieved: 2012-07-21.
- [8] S. Jones, “Online classifieds.”
- [9] A. Goldfarb and C. E. Tucker, “Online advertising, behavioral targeting, and privacy,” *Commun. ACM*, vol. 54, pp. 25–27, May 2011.
- [10] V. Gribova and P. Kachanov, “An approach to automated user interest matching in online classified advertising systems,” in *ICIC (1)* (D.-S. Huang, K.-H. Jo, H.-H. Lee, H.-J. Kang, and V. Bevilacqua, eds.), vol. 5754 of *Lecture Notes in Computer Science*, pp. 665–673, Springer, 2009.

- [11] J. Rowley and R. Hartley, *Organizing Knowledge: An Introduction to Managing Access to Information*. Ashgate, 2007.
- [12] “<http://www.bibsonomy.org> (bibsonomy).” Accessed: 2012-08-31.
- [13] A. Hotho, R. Jaschke, C. Schmitz, and G. Stumme, “BibSonomy: A social bookmark and publication sharing system,” in *Proceedings of the First Conceptual Structures Tool Interoperability Workshop at the 14th International Conference on Conceptual Structures* (A. de Moor, S. Polovina, and H. Delugach, eds.), (Aalborg), pp. 87–102, Aalborg Universitetsforlag, 2006.
- [14] L. Steels, “Collaborative tagging as distributed cognition,” *Pragmatics & Cognition*, vol. 14, no. 2, pp. 287–292, 2006.
- [15] H. Halpin, V. Robu, and H. Shepherd, “The complex dynamics of collaborative tagging,” in *WWW ’07: Proceedings of the 16th international conference on World Wide Web*, (New York, NY, USA), pp. 211–220, ACM Press, 2007.
- [16] A. Mathes, “Folksonomies-cooperative classification and communication through shared metadata,” *Computer Mediated Communication*, vol. 47, no. 10, pp. 1–13, 2004.
- [17] S. Butterfield, “Folksonomy.”
- [18] V. Durieux, “Collaborative tagging as a user-generated tool for the organization of online healthcare resources,” 2011.
- [19] M. M. N. P. V. S. S. R. S. L. V. S. Back, A., “Tags and tagging: Creating meanings, organizing, and socializing with metadata,” 2008.
- [20] S. Bateman, C. Brooks, G. McCalla, and P. Brusilovsky, “Applying collaborative tagging to e-learning,” in *Proceedings of the Workshop on Tagging and Metadata for Social Information Organization (WWW’07)*, 2007.
- [21] L. Terveen and W. Hill, “Beyond recommender systems: Helping people help each other,” 2001.



- [22] B. M. Sarwar, G. Karypis, J. A. Konstan, and J. Riedl, “Item-based collaborative filtering recommendation algorithms.,” in *WWW*, pp. 285–295, 2001.
- [23] P. Melville and V. Sindhvani, “Recommender Systems,” 2010.
- [24] Y. Wang, L. Yin, B. Cheng, and Y. Yu, “Learning to recommend based on slope one strategy.,” in *APWeb* (Q. Z. Sheng, G. Wang, C. S. Jensen, and G. Xu, eds.), vol. 7235 of *Lecture Notes in Computer Science*, pp. 537–544, Springer, 2012.
- [25] D. Lemire and A. Maclachlan, “Slope one predictors for online rating-based collaborative filtering,” *CoRR*, vol. abs/cs/0702144, 2007.
- [26] “<http://code.google.com/p/osclass> (osclass).” Accessed: 2012-09-30.
- [27] “<http://encyclopedia2.thefreedictionary.com/dbms> (dbms).” Accessed: 2012-09-30.
- [28] “<http://www.databasedir.com/what-is-rdbms> (what is rdbms).”

# XI. Appendix

## A. Source Code

```
<?php

/**
 * The base MySQL settings of Oclass
 */
define('MULTISITE', 0);

/** MySQL database name for Oclass */
define('DB.NAME', 'database.name');

/** MySQL database username */
define('DB.USER', 'username');

/** MySQL database password */
define('DB.PASSWORD', 'password');

/** MySQL hostname */
define('DB.HOST', 'localhost');

/** Database Table prefix */
define('DB.TABLE_PREFIX', 'oc_');

define('REL.WEB.URL', 'rel.here');

define('WEB.PATH', 'http://localhost');

?>

<?php
/*
 *      Oclass      software for creating and publishing online classified
 *                  advertising platforms
 *
 *                  Copyright (C) 2012 OSCLASS
 *
 *      This program is free software: you can redistribute it and/or
 *      modify it under the terms of the GNU Affero General Public License
 *      as published by the Free Software Foundation, either version 3 of
 *      the License, or (at your option) any later version.
 *
 *      This program is distributed in the hope that it will be useful, but
 *      WITHOUT ANY WARRANTY; without even the implied warranty of
 *      MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 *      GNU Affero General Public License for more details.
 *
 *      You should have received a copy of the GNU Affero General Public
 *      License along with this program. If not, see <http://www.gnu.org/licenses/>.
 */

define('ABS_PATH', str_replace('\\', '/', dirname($_SERVER['SCRIPT_FILENAME']) . '/'));
if(PHP_SAPI=='cli') {
    define('CLI', true);
}

require_once ABS_PATH . 'oc-load.php';

if( CLI ) {
    $cli_params = getopt('p:t:');
    Params::setParam('page', $cli_params['p']);
    Params::setParam('cron-type', $cli_params['t']);
    if(Params::getParam('page')=='upgrade') {
        require_once(osc_lib_path() . 'osclass/upgrade-funcs.php');
        exit(1);
    } else if( !in_array(Params::getParam('page'), array('cron')) && !in_array(Params::
        getParam('cron-type'), array('hourly', 'daily', 'weekly')) ) {
        exit(1);
    }
}

if( file_exists(ABS_PATH . 'maintenance') ) {
    if(!osc_is_admin_user_logged_in()) {
        require_once LIB_PATH . 'osclass/helpers/hErrors.php';

        $title = sprintf('--(Maintenance &raquo; %s)', osc_page_title());
        $message = sprintf('--(We are sorry for any inconvenience. %s is undergoing
            maintenance.) . '.', osc_page_title() );

        header('HTTP/1.1 503 Service Temporarily Unavailable');
        header('Status: 503 Service Temporarily Unavailable');
        header('Retry-After: 900');
        osc_die($title, $message);
    } else {
        define('__OSC_MAINTENANCE__', true);
    }
}

if(!osc_users_enabled() && osc_is_web_user_logged_in()) {
    Session::newInstance()->drop('userId');
```

```

        Session::newInstance()->.drop('userName');
        Session::newInstance()->.drop('userEmail');
        Session::newInstance()->.drop('userPhone');

        Cookie::newInstance()->.pop('oc_userId');
        Cookie::newInstance()->.pop('oc_userSecret');
        Cookie::newInstance()->.set();
    }

    if(osc_is_web_user_logged_in()) {
        User::newInstance()->.lastAccess(osc_logged_user_id(), date('Y-m-d H:i:s'), $_SERVER['
        REMOTE_ADDR'], 3600);
    }

    switch( Params::getParam('page') )
    {
        case ('cron'): // cron system
            define('_FROM_CRON_', true);
            require_once(osc_lib_path() . 'osclass/cron.php');
            break;
        case ('user'): // user pages (with security)
            if(Params::getParam('action')==='change_email_confirm' || Params::
            getParam('action')==='activate_alert'
            || (Params::getParam('action')==='unsubscribe' && !
            osc_is_web_user_logged_in())
            || Params::getParam('action')==='contact_post'
            || Params::getParam('action')==='pub_profile') {
                require_once(osc_lib_path() . 'osclass/controller/user-non-
                secure.php');
                $do = new CWebUserNonSecure();
                $do->.doModel();
            } else {
                require_once(osc_lib_path() . 'osclass/controller/user.php');
                $do = new CWebUser();
                $do->.doModel();
            }
            break;
        case ('item'): // item pages
            require_once(osc_lib_path() . 'osclass/controller/item.php');
            $do = new CWebItem();
            $do->.doModel();
            break;
        case ('search'): // search pages
            require_once(osc_lib_path() . 'osclass/controller/search.php');
            $do = new CWebSearch();
            $do->.doModel();
            break;
        case ('page'): // static pages
            require_once(osc_lib_path() . 'osclass/controller/page.php');
            $do = new CWebPage();
            $do->.doModel();
            break;
        case ('register'): // register page
            require_once(osc_lib_path() . 'osclass/controller/register.php');
            $do = new CWebRegister();
            $do->.doModel();
            break;
        case ('ajax'): // ajax
            require_once(osc_lib_path() . 'osclass/controller/ajax.php');
            $do = new CWebAjax();
            $do->.doModel();
            break;
        case ('login'): // login page
            require_once(osc_lib_path() . 'osclass/controller/login.php');
            $do = new CWebLogin();
            $do->.doModel();
            break;
        case ('language'): // set language
            require_once(osc_lib_path() . 'osclass/controller/language.php');
            $do = new CWebLanguage();
            $do->.doModel();
            break;
        case ('contact'): // contact
            require_once(osc_lib_path() . 'osclass/controller/contact.php');
            $do = new CWebContact();
            $do->.doModel();
            break;
        case ('custom'): // contact
            require_once(osc_lib_path() . 'osclass/controller/custom.php');
            $do = new CWebCustom();
            $do->.doModel();
            break;
        default: // home and static pages that are mandatory...
            require_once(osc_lib_path() . 'osclass/controller/main.php');
            $do = new CWebMain();
            $do->.doModel();
            break;
    }

    if(!defined('_FROM_CRON_')) {
        if( osc_auto_cron() ) {
            osc_doRequest(osc_base_url(), array('page' => 'cron'));
        }
    }

    /* file end: ./index.php */
?>
<?php

```

```

/*
 *   Oclass      software for creating and publishing online classified
 *               advertising platforms
 *
 *               Copyright (C) 2012 OSCLASS
 *
 *   This program is free software: you can redistribute it and/or
 *   modify it under the terms of the GNU Affero General Public License
 *   as published by the Free Software Foundation, either version 3 of
 *   the License, or (at your option) any later version.
 *
 *   This program is distributed in the hope that it will be useful, but
 *   WITHOUT ANY WARRANTY; without even the implied warranty of
 *   MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 *   GNU Affero General Public License for more details.
 *
 *   You should have received a copy of the GNU Affero General Public
 *   License along with this program. If not, see <http://www.gnu.org/licenses/>.
 */

define('OSCLASS_VERSION', '3.2.2');

if( !defined('ABS_PATH') ) {
    define( 'ABS_PATH', str_replace('\', '/', dirname(__FILE__) . '/' ));
}

define('LIB_PATH', ABS_PATH . 'oc-includes/');
define('CONTENT_PATH', ABS_PATH . 'oc-content/');
define('THEMES_PATH', CONTENT_PATH . 'themes/');
define('PLUGINS_PATH', CONTENT_PATH . 'plugins/');
define('TRANSLATIONS_PATH', CONTENT_PATH . 'languages/');

if( !file_exists(ABS_PATH . 'config.php') ) {
    require_once LIB_PATH . 'osclass/helpers/hErrors.php';

    $title = 'Oclass &raquo; Error';
    $message = 'There doesn\'t seem to be a <code>config.php</code> file. Oclass isn\'t
        installed. <a href="http://forums.osclass.org/">Need more help?</a></p>';
    $message .= '<p><a class="button" href="' . osc_get_absolute_url() . 'oc-includes/osclass/
        install.php">Install</a></p>';
    osc_die($title, $message);
}

// load database configuration
require_once ABS_PATH . 'config.php';
require_once LIB_PATH . 'osclass/default-constants.php';

// Sets PHP error handling
if( OSC_DEBUG ) {
    ini_set( 'display_errors', 1 );
    error_reporting( E_ALL | E_STRICT );

    if( OSC_DEBUG_LOG ) {
        ini_set( 'display_errors', 0 );
        ini_set( 'log_errors', 1 );
        ini_set( 'error_log', CONTENT_PATH . 'debug.log' );
    }
} else {
    error_reporting( E_CORE_ERROR | E_CORE_WARNING | E_COMPILE_ERROR | E_ERROR | E_WARNING |
        E_PARSE | E_USER_ERROR | E_USER_WARNING );
}

require_once LIB_PATH . 'osclass/db.php';
require_once LIB_PATH . 'osclass/Logger/LogDatabase.php';
require_once LIB_PATH . 'osclass/classes/database/DBConnectionClass.php';
require_once LIB_PATH . 'osclass/classes/database/DBCommandClass.php';
require_once LIB_PATH . 'osclass/classes/database/DBRecordsetClass.php';
require_once LIB_PATH . 'osclass/classes/database/DAO.php';
require_once LIB_PATH . 'osclass/model/SiteInfo.php';
require_once LIB_PATH . 'osclass/helpers/hDatabaseInfo.php';
require_once LIB_PATH . 'osclass/model/Preference.php';
require_once LIB_PATH . 'osclass/helpers/hPreference.php';

// check if Oclass is installed
if( !getBoolPreference('osclass_installed') && MULTISITE ) {
    header('Location: ' . WEB_PATH); die;
} else if( !getBoolPreference('osclass_installed') ) {
    require_once LIB_PATH . 'osclass/helpers/hErrors.php';

    $title = 'Oclass &raquo; Error';
    $message = 'Oclass isn\'t installed. <a href="http://forums.osclass.org/">Need more help
        ?</a></p>';
    $message .= '<p><a class="button" href="' . osc_get_absolute_url() . 'oc-includes/osclass/
        install.php">Install</a></p>';
    osc_die($title, $message);
}

require_once LIB_PATH . 'osclass/helpers/hDefines.php';
require_once LIB_PATH . 'osclass/helpers/hLocale.php';
require_once LIB_PATH . 'osclass/helpers/hMessages.php';
require_once LIB_PATH . 'osclass/helpers/hUsers.php';
require_once LIB_PATH . 'osclass/helpers/hItems.php';
require_once LIB_PATH . 'osclass/helpers/hSearch.php';
require_once LIB_PATH . 'osclass/helpers/hUtils.php';

require_once LIB_PATH . 'osclass/helpers/hCategories.php';
require_once LIB_PATH . 'osclass/helpers/hTranslations.php';
require_once LIB_PATH . 'osclass/helpers/hSecurity.php';

```

```

require_once LIB_PATH . 'osclass/helpers/hSanitize.php';
require_once LIB_PATH . 'osclass/helpers/hValidate.php';
require_once LIB_PATH . 'osclass/helpers/hPage.php';
require_once LIB_PATH . 'osclass/helpers/hPagination.php';
require_once LIB_PATH . 'osclass/helpers/hPremium.php';
require_once LIB_PATH . 'osclass/helpers/hTheme.php';
require_once LIB_PATH . 'osclass/core/Params.php';
require_once LIB_PATH . 'osclass/core/Cookie.php';
require_once LIB_PATH . 'osclass/core/Session.php';
require_once LIB_PATH . 'osclass/core/View.php';
require_once LIB_PATH . 'osclass/core/BaseModel.php';
require_once LIB_PATH . 'osclass/core/AdminBaseModel.php';
require_once LIB_PATH . 'osclass/core/SecBaseModel.php';
require_once LIB_PATH . 'osclass/core/WebSecBaseModel.php';
require_once LIB_PATH . 'osclass/core/AdminSecBaseModel.php';
require_once LIB_PATH . 'osclass/core/Translation.php';

require_once LIB_PATH . 'osclass/Themes.php';
require_once LIB_PATH . 'osclass/AdminThemes.php';
require_once LIB_PATH . 'osclass/WebThemes.php';
require_once LIB_PATH . 'osclass/compatibility.php';
require_once LIB_PATH . 'osclass/utils.php';
require_once LIB_PATH . 'osclass/formatting.php';
require_once LIB_PATH . 'osclass/locales.php';
require_once LIB_PATH . 'osclass/classes/Plugins.php';
require_once LIB_PATH . 'osclass/helpers/hPlugins.php';
require_once LIB_PATH . 'osclass/ItemActions.php';
require_once LIB_PATH . 'osclass/emails.php';
require_once LIB_PATH . 'osclass/model/Admin.php';
require_once LIB_PATH . 'osclass/model/Alerts.php';
require_once LIB_PATH . 'osclass/model/AlertsStats.php';
require_once LIB_PATH . 'osclass/model/Cron.php';
require_once LIB_PATH . 'osclass/model/Category.php';
require_once LIB_PATH . 'osclass/model/CategoryStats.php';
require_once LIB_PATH . 'osclass/model/City.php';
require_once LIB_PATH . 'osclass/model/CityArea.php';
require_once LIB_PATH . 'osclass/model/Country.php';
require_once LIB_PATH . 'osclass/model/Currency.php';
require_once LIB_PATH . 'osclass/model/OSCLocale.php';
require_once LIB_PATH . 'osclass/model/Item.php';
require_once LIB_PATH . 'osclass/model/ItemComment.php';
require_once LIB_PATH . 'osclass/model/ItemResource.php';
require_once LIB_PATH . 'osclass/model/ItemStats.php';
require_once LIB_PATH . 'osclass/model/Page.php';
require_once LIB_PATH . 'osclass/model/PluginCategory.php';
require_once LIB_PATH . 'osclass/model/Region.php';
require_once LIB_PATH . 'osclass/model/User.php';
require_once LIB_PATH . 'osclass/model/UserEmailTmp.php';
require_once LIB_PATH . 'osclass/model/ItemLocation.php';
require_once LIB_PATH . 'osclass/model/Widget.php';
require_once LIB_PATH . 'osclass/model/Search.php';
require_once LIB_PATH . 'osclass/model/LatestSearches.php';
require_once LIB_PATH . 'osclass/model/Field.php';
require_once LIB_PATH . 'osclass/model/Log.php';
require_once LIB_PATH . 'osclass/model/CountryStats.php';
require_once LIB_PATH . 'osclass/model/RegionStats.php';
require_once LIB_PATH . 'osclass/model/CityStats.php';
require_once LIB_PATH . 'osclass/model/BanRule.php';

require_once LIB_PATH . 'osclass/model/LocationsTmp.php';

require_once LIB_PATH . 'osclass/classes/Cache.php';
require_once LIB_PATH . 'osclass/classes/ImageResizer.php';
require_once LIB_PATH . 'osclass/classes/RSSFeed.php';
require_once LIB_PATH . 'osclass/classes/Sitemap.php';
require_once LIB_PATH . 'osclass/classes/Pagination.php';
require_once LIB_PATH . 'osclass/classes/Watermark.php';
require_once LIB_PATH . 'osclass/classes/Rewrite.php';
require_once LIB_PATH . 'osclass/classes/Stats.php';
require_once LIB_PATH . 'osclass/classes/AdminMenu.php';
require_once LIB_PATH . 'osclass/classes/datatables/DataTable.php';
require_once LIB_PATH . 'osclass/classes/AdminToolbar.php';
require_once LIB_PATH . 'osclass/classes/Breadcrumb.php';
require_once LIB_PATH . 'osclass/classes/EmailVariables.php';
require_once LIB_PATH . 'osclass/alerts.php';

require_once LIB_PATH . 'osclass/classes/Dependencies.php';
require_once LIB_PATH . 'osclass/classes/Scripts.php';
require_once LIB_PATH . 'osclass/classes/Styles.php';

require_once LIB_PATH . 'osclass/frm/Form.form.class.php';
require_once LIB_PATH . 'osclass/frm/Page.form.class.php';
require_once LIB_PATH . 'osclass/frm/Category.form.class.php';
require_once LIB_PATH . 'osclass/frm/Item.form.class.php';
require_once LIB_PATH . 'osclass/frm/Contact.form.class.php';
require_once LIB_PATH . 'osclass/frm/Comment.form.class.php';
require_once LIB_PATH . 'osclass/frm/User.form.class.php';
require_once LIB_PATH . 'osclass/frm/Language.form.class.php';
require_once LIB_PATH . 'osclass/frm/SendFriend.form.class.php';
require_once LIB_PATH . 'osclass/frm/Alert.form.class.php';
require_once LIB_PATH . 'osclass/frm/Field.form.class.php';
require_once LIB_PATH . 'osclass/frm/Admin.form.class.php';
require_once LIB_PATH . 'osclass/frm/ManageItems.form.class.php';
require_once LIB_PATH . 'osclass/frm/BanRule.form.class.php';

require_once LIB_PATH . 'osclass/functions.php';
require_once LIB_PATH . 'osclass/helpers/hAdminMenu.php';

define('__OSC_LOADED__', true);

```

```

// Moved from BaseModel, since we need some session magic on index.php;)
Session::newInstance()->session_start();

if( osc_timezone() != '' ) {
    date_default_timezone_set(osc_timezone());
}

function osc_show_maintenance() {
    if(defined('_OSC_MAINTENANCE_')) { ?>
        <div id="maintenance" name="maintenance">
            <?php _e("The website is currently undergoing maintenance"); ?>
        </div>
    <?php }
}

function osc_show_maintenance_css() {
    if(defined('_OSC_MAINTENANCE_')) { ?>
<style>
#maintenance {
    position: static;
    top: 0px;
    right: 0px;
    background-color: #bc0202;
    width: 100%;
    height: 20px;
    text-align: center;
    padding: 5px 0;
    font-size: 14px;
    color: #fefefe;
}
</style>
<?php }
}

function osc_meta_generator() {
    echo '<meta name="generator" content="Osclass ' . OSCCLASS_VERSION . '" />';
}

osc_add_hook('header', 'osc_show_maintenance');
osc_add_hook('header', 'osc_show_maintenance.css');
osc_add_hook('header', 'osc_meta_generator');
osc_add_hook('header', 'osc_load_scripts', 10);
osc_add_hook('header', 'osc_load_styles', 10);

// register scripts
osc_register_script('jquery', osc_assets_url('js/jquery.min.js'));
osc_register_script('jquery-ui', osc_assets_url('js/jquery-ui.min.js'), 'jquery');
osc_register_script('jquery-json', osc_assets_url('js/jquery.json.js'), 'jquery');
osc_register_script('jquery-treeview', osc_assets_url('js/jquery.treeview.js'), 'jquery');
osc_register_script('jquery-nested', osc_assets_url('js/jquery.ui.nestedSortable.js'), 'jquery');
osc_register_script('jquery-validate', osc_assets_url('js/jquery.validate.min.js'), 'jquery');
osc_register_script('tabber', osc_assets_url('js/tabber-minimized.js'), 'jquery');
osc_register_script('tiny-mce', osc_assets_url('js/tiny-mce/tiny-mce.js'));
osc_register_script('colorpicker', osc_assets_url('js/colorpicker/js/colorpicker.js'));
osc_register_script('fancybox', osc_assets_url('js/fancybox/jquery.fancybox.pack.js'), array('jquery'));
osc_register_script('jquery-migrate', osc_assets_url('js/jquery-migrate.min.js'), array('jquery'));
osc_register_script('php-date', osc_assets_url('js/date.js'));

Plugins::init();
osc_csrfguard_start();

if( OC_ADMIN ) {
    // init admin menu
    AdminMenu::newInstance()->init();
    $functions_path = AdminThemes::newInstance()->getCurrentThemePath() . 'functions.php';
    if( file_exists($functions_path) ) {
        require_once $functions_path;
    }
} else {
    Rewrite::newInstance()->init();
}

if( !class_exists('PHPMailer') ) {
    require_once osc_lib_path() . 'phpmailer/class.phpmailer.php';
}
if( !class_exists('SMTP') ) {
    require_once osc_lib_path() . 'phpmailer/class.smtp.php';
}

/* file end: ./oc-load.php */

<?php if ( ! defined('ABS_PATH')) exit('ABS_PATH is not loaded. Direct access is not allowed
. ');

/*
 *      Osclass      software for creating and publishing online classified
 *                   advertising platforms
 *
 *                   Copyright (C) 2012 OSCLASS
 *
 *      This program is free software: you can redistribute it and/or
 *      modify it under the terms of the GNU Affero General Public License
 *      as published by the Free Software Foundation, either version 3 of
 *      the License, or (at your option) any later version.
 *
 *      This program is distributed in the hope that it will be useful, but
 *      WITHOUT ANY WARRANTY; without even the implied warranty of
 *      MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the

```

```

*          GNU Affero General Public License for more details.
*
*          You should have received a copy of the GNU Affero General Public
* License along with this program. If not, see <http://www.gnu.org/licenses/>.
*/

```

```

class CAdminAdmins extends AdminSecBaseModel
{
    //specific for this class
    private $adminManager;

    function __construct()
    {
        parent::__construct();

        if( $this->isModerator() ) {
            if((($this->action!='edit' && $this->action!='edit_post') || Params::getParam('id')!='') && Params::getParam('id') != osc_logged_admin_id()) {
                osc_add_flash_error_message(_m("You don't have enough permissions"), 'admin');
                $this->redirectTo(osc_admin_base_url());
            }
        }

        //specific things for this class
        $this->adminManager = Admin::newInstance();
    }

    //Business Layer...
    function doModel()
    {
        parent::doModel();

        switch($this->action) {
            case('add'): // callin add view
                $this->exportVariableToView( 'admin', null );
                $this->doView( 'admins/frm.php' );
                break;
            case('add_post'): if( defined('DEMO') ) {
                osc_add_flash_warning_message( _m("This action can't be done because it's a demo site"), 'admin');
                $this->redirectTo(osc_admin_base_url(true) . '?page=admins');
            }
            osc_csrf_check();
            // adding a new admin
            $sPassword = Params::getParam('s_password', false, false);
            $sName = Params::getParam('s_name');
            $sEmail = Params::getParam('s_email');
            $sUserName = Params::getParam('s_username');
            $bModerator = Params::getParam('b_moderator')==0?0:1;

            // cleaning parameters
            $sPassword = strip_tags($sPassword);
            $sPassword = trim($sPassword);
            $sName = strip_tags($sName);
            $sName = trim($sName);
            $sEmail = strip_tags($sEmail);
            $sEmail = trim($sEmail);
            $sUserName = strip_tags($sUserName);
            $sUserName = trim($sUserName);

            // Checks for legit data
            if( !osc_validate_email($sEmail, true) ) {
                osc_add_flash_warning_message( _m("Email invalid"), 'admin');
                $this->redirectTo(osc_admin_base_url(true) . '?page=admins&action=add');
            }
            if( !osc_validate_username($sUserName) ) {
                osc_add_flash_warning_message( _m("Username invalid"), 'admin');
                $this->redirectTo(osc_admin_base_url(true) . '?page=admins&action=add');
            }
            if( $sName == '' ) {
                osc_add_flash_warning_message( _m("Name invalid"), 'admin');
                $this->redirectTo(osc_admin_base_url(true) . '?page=admins&action=add');
            }
            if( $sPassword == '' ) {
                osc_add_flash_warning_message( _m("Password invalid"), 'admin');
                $this->redirectTo(osc_admin_base_url(true) . '?page=admins&action=add');
            }
            $admin = $this->adminManager->findByEmail($sEmail);
            if( $admin ) {
                osc_add_flash_warning_message( _m("Email already in use"), 'admin');
                $this->redirectTo(osc_admin_base_url(true) . '?page=admins&action=add');
            }
            $admin = $this->adminManager->findByUsername($sUserName);
            if( $admin ) {
                osc_add_flash_warning_message( _m("Username already in use"), 'admin');
                $this->redirectTo(osc_admin_base_url(true) . '?page=

```

```

        admins&action=add');
    }
    $array = array(
        's_password' => sha1($sPassword),
        's_name' => $sName,
        's_email' => $sEmail,
        's_username' => $sUserName,
        'b_moderator' => $bModerator
    );
    $isInserted = $this->adminManager->insert($array);
    if( $isInserted ) {
        // send email
        osc_run_hook('hook_email_new_admin', array(
            's_name' => $sName,
            's_username' => $sUserName,
            's_password' => $sPassword,
            's_email' => $sEmail
        ));
        osc_add_flash_ok_message( _m('The admin has been added
        '), 'admin');
    } else {
        osc_add_flash_error_message( _m('There has been an
        error adding a new admin'), 'admin');
    }
    $this->redirectTo( osc_admin_base_url(true) . '?page=admins' )
    ;
break;
case('edit') :
    // calling edit admin view
    $adminEdit = null;
    $adminId = Params::getParam('id');
    if( $adminId != '' ) {
        $adminEdit = $this->adminManager->findByPrimaryKey((
        int) $adminId);
    } elseif( Session::newInstance()->get('adminId') != '' ) {
        $adminEdit = $this->adminManager->findByPrimaryKey(
        Session::newInstance()->get('adminId') );
    }
    if( count($adminEdit) == 0 ) {
        osc_add_flash_error_message( _m('There is no admin
        with this id'), 'admin');
        $this->redirectTo( osc_admin_base_url(true) . '?page=
        admins' );
    }
    $this->exportVariableToView("admin", $adminEdit);
    $this->doView('admins/frm.php');
break;
case('edit_post') :
    if( defined('DEMO') ) {
        osc_add_flash_warning_message( _m("This action can't
        be done because it's a demo site"), 'admin');
        $this->redirectTo( osc_admin_base_url(true) . '?page=
        admins' );
    }
    osc_csrf_check();
    // updating a new admin
    $iUpdated = 0;
    $adminId = Params::getParam('id');
    $sPassword = Params::getParam('s_password', false,
    false);
    $sPassword2 = Params::getParam('s_password2', false,
    false);
    $sOldPassword = Params::getParam('old_password', false,
    false);
    $sName = Params::getParam('s_name');
    $sEmail = Params::getParam('s_email');
    $sUserName = Params::getParam('s_username');
    $bModerator = Params::getParam('b_moderator') == 0?0:1;
    // cleaning parameters
    $sPassword = strip_tags($sPassword);
    $sPassword = trim($sPassword);
    $sPassword2 = strip_tags($sPassword2);
    $sPassword2 = trim($sPassword2);
    $sName = strip_tags($sName);
    $sName = trim($sName);
    $sEmail = strip_tags($sEmail);
    $sEmail = trim($sEmail);
    $sUserName = strip_tags($sUserName);
    $sUserName = trim($sUserName);
    // Checks for legit data
    if( !osc_validate_email($sEmail, true) ) {
        osc_add_flash_warning_message( _m("Email invalid"), '
        admin');
        $this->redirectTo( osc_admin_base_url(true) . '?page=
        admins&action=edit&id=' . $adminId );
    }
    if( !osc_validate_username($sUserName) ) {
        osc_add_flash_warning_message( _m("Username invalid"),
        'admin');
        $this->redirectTo( osc_admin_base_url(true) . '?page=
        admins&action=edit&id=' . $adminId );
    }

```



```

}
if( $sName == '' ) {
    osc_add_flash_warning_message( _m("Name invalid"), '
    admin');
    $this->redirectTo(osc_admin_base_url(true) . '?page=
    admins&action=edit&id=' . $adminId);
}
$Admin = $this->adminManager->findByPrimaryKey($adminId);
if( count($Admin) == 0 ) {
    osc_add_flash_error_message( _m("This admin doesn't
    exist"), 'admin');
    $this->redirectTo(osc_admin_base_url(true) . '?page=
    admins');
}
if( $Admin['s_email'] != $sEmail ) {
    if($this->adminManager->findByEmail( $sEmail ) ) {
        osc_add_flash_warning_message( _m('Existing email
        '), 'admin');
        $this->redirectTo(osc_admin_base_url(true) . '?page=
        admins&action=edit&id=' . $adminId);
    }
}
if( $Admin['s_username'] != $sUserName ) {
    if( $this->adminManager->findByUsername( $sUserName )
    ) {
        osc_add_flash_warning_message( _m('Existing
        username'), 'admin');
        $this->redirectTo(osc_admin_base_url(true) . '?
        page=admins&action=edit&id=' . $adminId);
    }
}
$conditions = array('pk_i_id' => $adminId);
$array = array();
if(osc_logged_admin_id()==$adminId) {
    if($sOldPassword != '') {
        if( $sPassword=='' ) {
            osc_add_flash_warning_message( _m("Password
            invalid"), 'admin');
            $this->redirectTo(osc_admin_base_url(true) .
            '?page=admins&action=edit&id=' . $adminId)
            ;
        } else {
            $firstCondition = ( sha1($sOldPassword) ==
            $Admin['s_password'] );
            $secondCondition = ( $sPassword == $sPassword2
            );
            if( $firstCondition && $secondCondition ) {
                $array['s_password'] = sha1($sPassword);
            } else {
                osc_add_flash_warning_message( _m("The
                password couldn't be updated. Passwords
                don't match"), 'admin');
                $this->redirectTo(osc_admin_base_url(true)
                . '?page=admins&action=edit&id=' .
                $adminId);
            }
        }
    }
} else {
    if( $sPassword!='' ) {
        if($sPassword == $sPassword2) {
            $array['s_password'] = sha1($sPassword);
        } else {
            osc_add_flash_warning_message( _m("The
            password couldn't be updated. Passwords
            don't match"), 'admin');
            $this->redirectTo(osc_admin_base_url(true) .
            '?page=admins&action=edit&id=' . $adminId)
            ;
        }
    }
}
if($adminId!=osc_logged_admin_id()) {
    $array['b_moderator'] = $bModerator;
}
$array['s_name'] = Params::getParam('s_name');
$array['s_username'] = $sUserName;
$array['s_email'] = $sEmail;
$iUpdated = $this->adminManager->update($array ,
    $conditions);
if( $iUpdated > 0 ) {
    osc_add_flash_ok_message( _m('The admin has been
    updated'), 'admin');
}
if( $this->isModerator() ) {
    $this->redirectTo(osc_admin_base_url(true));
} else {
    $this->redirectTo(osc_admin_base_url(true) . '?page=

```

```

        admins');
    }
break;
case('delete'):
    if( defined('DEMO') ) {
        osc_add_flash_warning_message( _m("This action can't
        be done because it's a demo site"), 'admin');
        $this->redirectTo(osc_admin_base_url(true) . '?page=
        admins');
    }
    osc_csrf_check();
    // deleting and admin
    $isDeleted = false;
    $adminId = Params::getParam('id');

    if( !is_array($adminId) ) {
        osc_add_flash_error_message( _m("The admin id isn't in
        the correct format"), 'admin');
        $this->redirectTo(osc_admin_base_url(true) . '?page=
        admins');
    }

    // Verification to avoid an administrator trying to remove
    to itself
    if( in_array( Session::newInstance()->get('adminId'),
    $adminId ) ) {
        osc_add_flash_error_message( _m("The operation hasn't
        been completed. You're trying to remove yourself
        !"), 'admin');
        $this->redirectTo(osc_admin_base_url(true) . '?page=
        admins');
    }

    $isDeleted = $this->adminManager->deleteBatch( $adminId );

    if( $isDeleted ) {
        osc_add_flash_ok_message( _m('The admin has been
        deleted correctly'), 'admin');
    } else {
        osc_add_flash_error_message( _m('The admin couldn\'t
        be deleted'), 'admin');
    }
    $this->redirectTo(osc_admin_base_url(true) . '?page=admins
    ');

break;
default:

    if(Params::getParam("action")!=="" ) {
        osc_run_hook("admin_bulk_",Params::getParam("action"),
        Params::getParam('id'));
    }

    if( Params::getParam('iDisplayLength') == '' ) {
        Params::setParam('iDisplayLength', 10 );
    }

    $p_iPage = 1;
    if( is_numeric(Params::getParam('iPage')) && Params::
    getParam('iPage') >= 1 ) {
        $p_iPage = Params::getParam('iPage');
    }
    Params::setParam('iPage', $p_iPage);

    $admins = $this->adminManager->listAll();

    // pagination
    $start = ($p_iPage-1) * Params::getParam('iDisplayLength')
    ;
    $limit = Params::getParam('iDisplayLength');
    $count = count( $admins );

    $displayRecords = $limit;
    if( ($start+$limit) > $count ) {
        $displayRecords = ($start+$limit) - $count;
    }
    // ----
    $aData = array();
    $smax = ($start+$limit);
    if($smax > $count) $smax = $count;
    for($i = $start; $i < $smax; $i++) {

        $admin = $admins[$i];

        $oOptions = array();
        $oOptions[] = '<a href="' . osc_admin_base_url(true) .
        '?page=admins&action=edit&id=' . $admin['
        pk.i.id'] . '">' . _m('Edit') . '</a>';
        $oOptions[] = '<a onclick="return delete_dialog(\'' .
        $admin['pk.i.id'] . '\');" href="' .
        osc_admin_base_url(true) . '?page=admins&action=
        delete&id=' . $admin['pk.i.id'] . '">' . _m(
        'Delete') . '</a>';
        $auxOptions = '<ul>'.PHP_EOL;
        foreach( $oOptions as $aActual ) {
            $auxOptions .= '<li>'. $aActual . '</li>'.PHP_EOL;
        }
        $aActions = '<div class="actions">'. $auxOptions . '</div
        >'.PHP_EOL;

        $row = array();

```

```

        $row[] = '<input type="checkbox" name="id[]" value="'
            . $admin['pk_i_id'] . '" />';
        $row[] = $admin['s_username'] . $actions;
        $row[] = $admin['s_name'];
        $row[] = $admin['s_email'];

        $aData[] = $row;
    }
    $array['iTotalRecords'] = $displayRecords;
    $array['iTotalDisplayRecords'] = count($admins);
    $array['iDisplayLength'] = $limit;
    $array['aData'] = $aData;

    $page = (int)Params::getParam('iPage');
    if(count($array['aData']) == 0 && $page!=1) {
        $total = (int)$array['iTotalDisplayRecords'];
        $maxPage = ceil( $total / (int)$array['iDisplayLength'] );

        $url = osc_admin_base_url(true).'?'.$SERVER['QUERY_STRING'];

        if($maxPage==0) {
            $url = preg_replace('/&iPage=(\d)+/', '&iPage=1', $url);
            $this->redirectTo($url);
        }

        if($page > 1) {
            $url = preg_replace('/&iPage=(\d)+/', '&iPage='.$maxPage, $url);
            $this->redirectTo($url);
        }
    }

    $bulk_options = array(
        array('value' => '', 'data-dialog-content' => '', 'label' => __('Bulk actions')),
        array('value' => 'delete', 'data-dialog-content' => sprintf(__('Are you sure you want to %s the selected admins?'), strtolower(__('Delete'))), 'label' => __('Delete'))
    );
    $bulk_options = osc_apply_filter("admin_bulk_filter", $bulk_options);
    $this->exportVariableToView('bulk_options', $bulk_options);

    $this->exportVariableToView('aAdmins', $array);
    // calling manage admins view
    $this->doView('admins/index.php');

    }
    }
}

//hopefully generic...
function doView($file)
{
    osc_run_hook("before_admin_html");
    osc_current_admin_theme_path($file);
    Session::newInstance()->clearVariables();
    osc_run_hook("after_admin_html");
}

}

/* file end: ./oc-admin/admins.php */
?>

<?php if ( ! defined('ABS_PATH')) exit('ABS_PATH is not loaded. Direct access is not allowed
');

/**
 * Oclass software for creating and publishing online classified advertising
 * platforms
 *
 * Copyright (C) 2012 OSCLASS
 *
 * This program is free software: you can redistribute it and/or modify it under the terms
 * of the GNU Affero General Public License as published by the Free Software Foundation,
 * either version 3 of the License, or (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful, but WITHOUT ANY
 * WARRANTY;
 * without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
 * PURPOSE.
 * See the GNU Affero General Public License for more details.
 *
 * You should have received a copy of the GNU Affero General Public
 * License along with this program. If not, see <http://www.gnu.org/licenses/>.
 */

class CAdminAppearance extends AdminSecBaseModel
{
    function __construct()
    {
        parent::__construct();
    }

    //Business Layer...

```

```

function doModel()
{
    parent::doModel();
    //specific things for this class
    switch ($this->action) {
        case('add'):
            $this->doView("appearance/add.php");
            break;
        case('add_post'):
            if( defined('DEMO') ) {
                osc_add_flash_warning_message( _m("This action can't be done because
                it's a demo site"), 'admin');
                $this->redirectTo( osc_admin_base_url(true) . '?page=appearance' );
            }
            osc_csrf_check();
            $filePackage = Params::getFiles('package');
            if( isset($filePackage['size']) && $filePackage['size']!=0 ) {
                $path = osc_themes_path();
                (int) $status = osc_unzip_file($filePackage['tmp_name'], $path);
            } else {
                $status = 3;
            }

            switch ($status) {
                case(0): $msg = _m('The theme folder is not writable');
                        osc_add_flash_error_message($msg, 'admin');
                        break;
                case(1): $msg = _m('The theme has been installed correctly');
                        osc_add_flash_ok_message($msg, 'admin');
                        break;
                case(2): $msg = _m('The zip file is not valid');
                        osc_add_flash_error_message($msg, 'admin');
                        break;
                case(3): $msg = _m('No file was uploaded');
                        osc_add_flash_error_message($msg, 'admin');
                        $this->redirectTo( osc_admin_base_url(true) . "?page=
                        appearance&action=add");
                        break;
                case(-1):
                default: $msg = _m('There was a problem adding the theme');
                        osc_add_flash_error_message($msg, 'admin');
                        break;
            }

            $this->redirectTo( osc_admin_base_url(true) . "?page=appearance" );
            break;
        case('delete'):
            if( defined('DEMO') ) {
                osc_add_flash_warning_message( _m("This action can't be done because
                it's a demo site"), 'admin');
                $this->redirectTo( osc_admin_base_url(true) . '?page=appearance' );
            }
            osc_csrf_check();
            $theme = Params::getParam('webtheme');
            if($theme!='') {
                if($theme!= osc_current_web_theme()) {
                    if( osc_deleteDir( osc_content_path(). "themes/" . $theme . "/" ) ) {
                        osc_add_flash_ok_message( _m("Theme removed successfully"), "
                        admin");
                    } else {
                        osc_add_flash_error_message( _m("There was a problem removing
                        the theme"), "admin");
                    }
                } else {
                    osc_add_flash_error_message( _m("Current theme can not be deleted")
                    , "admin");
                }
            } else {
                osc_add_flash_error_message( _m("No theme selected"), "admin");
            }

            $this->redirectTo( osc_admin_base_url(true) . "?page=appearance" );
            break;
        /* widgets */
        case('widgets'):
            $info = WebThemes::newInstance()->loadThemeInfo( osc_theme() );

            $this->exportVariableToView("info", $info);

            $this->doView('appearance/widgets.php');
            break;
        case('add_widget'):
            $this->doView('appearance/add_widget.php');
            break;
        case('edit_widget'):
            $id = Params::getParam('id');

            $widget = Widget::newInstance()->findByPrimaryKey($id);
            $this->exportVariableToView("widget", $widget);

            $this->doView('appearance/add_widget.php');
            break;
        case('delete_widget'):
            osc_csrf_check();
            Widget::newInstance()->delete(
                array('pk_i_id' => Params::getParam('id') )
            );
            osc_add_flash_ok_message( _m('Widget removed correctly'), 'admin');
            $this->redirectTo( osc_admin_base_url(true) . "?page=appearance&action=

```

```

        widgets" );
break;
case('edit_widget_post'):
    osc_csrf_check();
    if(!osc_validate_text(Params::getParam("description"))) {
        osc_add_flash_error_message( _m('Description field is required'), '
            admin' );
        $this->redirectTo( osc_admin_base_url(true) . "?page=appearance&action
            =widgets" );
    }

    $res = Widget::newInstance()->update(
        array(
            's_description' => Params::getParam('description'),
            's_content' => Params::getParam('content', false, false)
        ),
        array('pk.i.id' => Params::getParam('id') )
    );

    if( $res ) {
        osc_add_flash_ok_message( _m('Widget updated correctly'), 'admin');
    } else {
        osc_add_flash_ok_message( _m('Widget cannot be updated correctly'), '
            admin' );
    }
    $this->redirectTo( osc_admin_base_url(true) . "?page=appearance&action=
        widgets" );
break;
case('add_widget_post'):
    osc_csrf_check();
    if(!osc_validate_text(Params::getParam("description"))) {
        osc_add_flash_error_message( _m('Description field is required'), '
            admin' );
        $this->redirectTo( osc_admin_base_url(true) . "?page=appearance&action
            =widgets" );
    }

    Widget::newInstance()->insert(
        array(
            's_location' => Params::getParam('location'),
            'e_kind' => 'html',
            's_description' => Params::getParam('description'),
            's_content' => Params::getParam('content', false, false)
        )
    );
    osc_add_flash_ok_message( _m('Widget added correctly'), 'admin');
    $this->redirectTo( osc_admin_base_url(true) . "?page=appearance&action=
        widgets" );
break;
/* /widget */
case('activate'):
    osc_csrf_check();
    osc_set_preference('theme', Params::getParam('theme'));
    osc_add_flash_ok_message( _m('Theme activated correctly'), 'admin');
    osc_run_hook("theme_activate", Params::getParam('theme'));
    $this->redirectTo( osc_admin_base_url(true) . "?page=appearance" );
break;
case('render'):
    $this->exportVariableToView('file', osc_base_path() . Params::getParam("
        file"));
    $this->doView('appearance/view.php');
break;
default:
    // $marketError = Params::getParam('marketError');
    // $slug = Params::getParam('slug');
    // if($marketError!='') {
    //     if($marketError == '0') { // no error installed ok
    //         $help = '<br/><br/><b>' . --('You only need to activate or
    // preview the theme').</b>';
    //         osc_add_flash_ok_message( --('Everything was OK!') . ' ( ' .
    // $slug . ' ) ' . $help, 'admin');
    //     } else {
    //         osc_add_flash_error_message( --('Error occurred') . ' ( ' .
    // $slug . ' ) ', 'admin');
    //     }
    // }

    // force the recount of themes that need to be updated
    if(Params::getParam('checkUpdated') != '') {
        osc_admin_toolbar_update_themes(true);
    }

    $themes = WebThemes::newInstance()->getListThemes();

    //preparing variables for the view
    $this->exportVariableToView("themes", $themes);

    $this->doView('appearance/index.php');
break;
}
}

//hopefully generic...
function doView($file)
{
    osc_run_hook("before_admin.html");
    osc_current_admin_theme_path($file);
    Session::newInstance()->clearVariables();
    osc_run_hook("after_admin.html");
}

```

```

    }
}
/* file end: ./oc-admin/appearance.php */
?>
<?php if ( ! defined('ABS_PATH')) exit('ABS_PATH is not loaded. Direct access is not allowed
');

/*
 *   Oclass      software for creating and publishing online classified
 *               advertising platforms
 *
 *               Copyright (C) 2012 OSCLASS
 *
 *   This program is free software: you can redistribute it and/or
 *   modify it under the terms of the GNU Affero General Public License
 *   as published by the Free Software Foundation, either version 3 of
 *   the License, or (at your option) any later version.
 *
 *   This program is distributed in the hope that it will be useful, but
 *   WITHOUT ANY WARRANTY; without even the implied warranty of
 *   MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 *   GNU Affero General Public License for more details.
 *
 *   You should have received a copy of the GNU Affero General Public
 *   License along with this program. If not, see <http://www.gnu.org/licenses/>.
 */

class CAdminCategories extends AdminSecBaseModel
{
    //specific for this class
    private $categoryManager;

    function __construct()
    {
        parent::__construct();

        //specific things for this class
        $this->categoryManager = Category::newInstance();
    }

    //Business Layer...
    function doModel()
    {
        parent::doModel();

        //specific things for this class
        switch ($this->action)
        {
            case('add_post_default'): // add default category and reorder parent
                categories
                    osc_csrf_check();
                    $fields['fk_i_parent_id'] = NULL;
                    $fields['i_expiration_days'] = 0;
                    $fields['i_position'] = 0;
                    $fields['b_enabled'] = 1;
                    $fields['b_price_enabled'] = 1;

                    $default_locale = osc_language();
                    $aFieldsDescription[$default_locale]['s_name'] = "NEW
                    CATEGORY, EDIT ME!";

                    $categoryId = $this->categoryManager->insert($fields ,
                    $aFieldsDescription);

                    // reorder parent categories. NEW category first
                    $rootCategories = $this->categoryManager->
                    findRootCategories();
                    foreach($rootCategories as $cat){
                        $order = $cat['i_position'];
                        $order++;
                        $this->categoryManager->updateOrder($cat['pk_i_id']
                        , $order);
                    }
                    $this->categoryManager->updateOrder($categoryId , '0');

                    $this->redirectTo(osc_admin_base_url(true) . '?page=
                    categories ');

                break;
                default: //
                    $this->exportVariableToView("categories" , $this->
                    categoryManager->toTreeAll() );
                    $this->doView("categories/index.php");
                }
        }

    //hopefully generic...
    function doView($file)
    {
        osc_run_hook("before_admin_html");
        osc_current_admin_theme_path($file);
        Session::newInstance()->clearVariables();
        osc_run_hook("after_admin_html");
    }
}

/* file end: ./oc-admin/categories.php */

```

?>

```
<?php if ( ! defined('ABS_PATH')) exit('ABS_PATH is not loaded. Direct access is not allowed
. ');

/*
 *   Oclass      software for creating and publishing online classified
 *               advertising platforms
 *
 *               Copyright (C) 2012 OSCLASS
 *
 *   This program is free software: you can redistribute it and/or
 *   modify it under the terms of the GNU Affero General Public License
 *   as published by the Free Software Foundation, either version 3 of
 *   the License, or (at your option) any later version.
 *
 *   This program is distributed in the hope that it will be useful, but
 *   WITHOUT ANY WARRANTY; without even the implied warranty of
 *   MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 *   GNU Affero General Public License for more details.
 *
 *   You should have received a copy of the GNU Affero General Public
 *   License along with this program. If not, see <http://www.gnu.org/licenses/>.
 */

class CAdminItemComments extends AdminSecBaseModel
{
    private $itemCommentManager;

    function __construct ()
    {
        parent::__construct ();

        //specific things for this class
        $this->itemCommentManager = ItemComment::newInstance ();
    }

    //Business Layer...
    function doModel ()
    {
        parent::doModel ();

        //specific things for this class
        switch ($this->action) {
            case ('bulk_actions') :

                osc_csrf_check ();
                $id = Params::getParam ('id ');
                if ( $id ) {
                    switch ( Params::getParam ('bulk_actions') ) {
                        case ('delete_all') :
                            $this->
                                itemCommentManager->delete (array (
                                    DB_CUSTOMCOND
                                    => '
                                        pk_i_id IN
                                        (
                                        .
                                        implode (',
                                        ', $id) .
                                        ',
                                    )'
                                ));
                            foreach ($id as
                                $id) {
                                $iUpdated =
                                    $this->
                                        itemCommentManager
                                        ->delete (
                                            array (
                                                'pk_i_id'
                                                =>
                                                    $id
                                            )
                                        );
                                osc_add_hook ("
                                    delete_comment
                                    ", $id);
                            }
                            osc_add_flash_ok_message
                                ( _m ('The
                                    comments have
                                    been deleted')
                                , 'admin');
                        break;
                        case ('activate_all') :
                            foreach ($id as
                                $id) {
                                $iUpdated =
                                    $this->
                                        itemCommentManager
                                        ->update (
                                            array ('
                                                b_active
                                                ' =>
                                                    1)
                                            , array ('
                                                pk_i_id
                                                ' =>
                                                    $id)
                                            );
                                if ($iUpdated)
                                    {

```

```

        $this->
            sendCommentActivated
            ($_id)
            ;
    }
    osc_add_hook("
        activate_comment
        ", $_id);
}
osc_add_flash_ok_message
( _m('The
comments have
been approved
'), 'admin' );
break;
case('deactivate_all'):
    foreach ($_id as
        $_id) {
        $this->
            itemCommentManager
            ->update(
                array('
                    b_active
                    ', =>
                    0),
                array('
                    pk_i_id
                    ', =>
                    $_id)
                );
        osc_add_hook("
            deactivate_comment
            ", $_id);
    }
    osc_add_flash_ok_message
    ( _m('The
    comments have
    been
    disapproved'),
    'admin' );
break;
case('enable_all'):
    foreach ($_id as
        $_id) {
        $iUpdated =
            $this->
            itemCommentManager
            ->update(
                array('
                    b_enabled
                    ', =>
                    1),
                array('
                    pk_i_id
                    ', =>
                    $_id)
                );
        if($iUpdated)
        {
            $this->
                sendCommentActivated
                ($_id)
                ;
        }
        osc_add_hook("
            enable_comment
            ", $_id);
    }
    osc_add_flash_ok_message
    ( _m('The
    comments have
    been unblocked
    '), 'admin' );
break;
case('disable_all'):
    foreach ($_id as
        $_id) {
        $this->
            itemCommentManager
            ->update(
                array('
                    b_enabled
                    ', =>
                    0),
                array('
                    pk_i_id
                    ', =>
                    $_id)
                );
        osc_add_hook("
            disable_comment
            ", $_id);
    }
    osc_add_flash_ok_message
    ( _m('The
    comments have
    been blocked')
    , 'admin' );
break;

```



```

        default:
            if (Params::getParam("bulk_actions")
                != "") {
                osc_run_hook("item_bulk_", Params::
                    getParam("bulk_actions"),
                    Params::getParam('id'));
            }
            break;
    }
}
$this->redirectTo( osc_admin_base_url(true) . "?"
    page=comments" );

break;
case('status'):

    osc_csrf_check();
    $id = Params::getParam('id');
    $value = Params::getParam('value');

    if (!$id) return false;
    $id = (int) $id;
    if (!is_numeric($id)) return false;
    if (!in_array($value, array('ACTIVE', 'INACTIVE',
        'ENABLE', 'DISABLE'))) return false;

    if( $value == 'ACTIVE' ) {
        $iUpdated = $this->itemCommentManager->update(
            array('b_active' => 1)
            ,array('pk_i_id' => $id)
        );
        if($iUpdated) {
            $this->sendCommentActivated($id);
        }
        osc_add_hook("activate_comment", $id);
        osc_add_flash_ok_message( _m('The comment has
            been approved'), 'admin');
    } else if ($value=='INACTIVE') {
        $iUpdated = $this->itemCommentManager->update(
            array('b_active' => 0)
            ,array('pk_i_id' => $id)
        );
        osc_add_hook("deactivate_comment", $id);
        osc_add_flash_ok_message( _m('The comment has
            been disapproved'), 'admin');
    } else if ($value=='ENABLE') {
        $iUpdated = $this->itemCommentManager->update(
            array('b_enabled' => 1)
            ,array('pk_i_id' => $id)
        );
        osc_add_hook("enable_comment", $id);
        osc_add_flash_ok_message( _m('The comment has
            been enabled'), 'admin');
    } else if ($value=='DISABLE') {
        $iUpdated = $this->itemCommentManager->update(
            array('b_enabled' => 0)
            ,array('pk_i_id' => $id)
        );
        osc_add_hook("disable_comment", $id);
        osc_add_flash_ok_message( _m('The comment has
            been disabled'), 'admin');
    }

    $this->redirectTo( osc_admin_base_url(true) . "?"
        page=comments" );

break;
case('comment_edit'):
    $comment = ItemComment::newInstance()->
        findByPrimaryKey( Params::getParam('id') );

    $this->_exportVariableToView('comment', $comment);
    $this->doView('comments/frm.php');

break;
case('comment_edit_post'):

    osc_csrf_check();

    $msg = '';
    if(!osc_validate_email(Params::getParam('
        authorEmail'),true)) {
        $msg .= _m('Email is not correct')."<br/>";
    }
    if(!osc_validate_text(Params::getParam('body'),1
        ,true)) {
        $msg .= _m('Comment is required')."<br/>";
    }

    if($msg!='') {
        osc_add_flash_error_message( $msg, 'admin' );
        $this->redirectTo( osc_admin_base_url(true) .
            "?page=comments&action=comment_edit&id=" .
            Params::getParam('id') );
    }

    $this->itemCommentManager->update(
        array(
            's_title' => Params::getParam('
                title'),
            's_body' => Params::getParam('body'),
            's_author_name' => Params::getParam('
                authorName'),
            's_author_email' => Params::getParam('

```

```

        authorEmail')
    ),
    array(
        'pk.i.id' => Params::getParam('id')
    )
);
osc_run_hook( 'edit_comment', Params::getParam('id') );

osc_add_flash_ok_message( _m('Great! We just updated your comment'), 'admin' );
$this->redirectTo( osc_admin_base_url(true) . "?page=comments" );

break;
case('delete'):

    osc_csrf_check();
    $this->itemCommentManager->deleteByPrimaryKey( Params::getParam('id') );
    osc_add_flash_ok_message( _m('The comment has been deleted'), 'admin' );
    osc_run_hook( 'delete_comment', Params::getParam('id') );
    $this->redirectTo( osc_admin_base_url(true) . "?page=comments" );

break;
default:

require_once osc_lib_path()."osclass/classes/datatables/CommentsDataTable.php";

// set default iDisplayLength
if( Params::getParam('iDisplayLength') != '' ) {
    Cookie::newInstance()->push( 'listing.iDisplayLength', Params::getParam('iDisplayLength') );
    Cookie::newInstance()->set();
} else {
    // set a default value if it's set in the cookie
    if( Cookie::newInstance()->get_value( 'listing.iDisplayLength' ) != '' ) {
        Params::setParam( 'iDisplayLength', Cookie::newInstance()->get_value( 'listing.iDisplayLength' ) );
    } else {
        Params::setParam( 'iDisplayLength', 10 );
    }
}
$this->_exportVariableToView( 'iDisplayLength', Params::getParam('iDisplayLength') );

// Table header order by related
if( Params::getParam('sort') == '' ) {
    Params::setParam( 'sort', 'date' );
}
if( Params::getParam('direction') == '' ) {
    Params::setParam( 'direction', 'desc' );
}

$page = (int)Params::getParam('iPage');
if($page==0) { $page = 1; };
Params::setParam('iPage', $page);

$params = Params::getParamsAsArray("get");

$commentsDataTable = new CommentsDataTable();
$commentsDataTable->table($params);
$aData = $commentsDataTable->getData();

if(count($aData['aRows']) == 0 && $page!=1) {
    $total = (int)$aData['iTotalDisplayRecords'];
    $maxPage = ceil( $total / (int)$aData['iDisplayLength'] );

    $url = osc_admin_base_url(true) . '?' . $_SERVER['QUERY_STRING'];

    if($maxPage==0) {
        $url = preg_replace('/&iPage=(\d)+/', '&iPage=1', $url);
        $this->redirectTo($url);
    }

    if($page > 1) {
        $url = preg_replace('/&iPage=(\d)+/', '&iPage='.$maxPage, $url);
        $this->redirectTo($url);
    }
}

$this->_exportVariableToView( 'aData', $aData );
$this->_exportVariableToView( 'aRawRows', $commentsDataTable->rawRows() );

$bulk_options = array(
    array('value' => '', 'data-dialog-content' => '', 'label' => _--('Bulk actions')),
    array('value' => 'delete_all', 'data-dialog-

```



```

switch( $this->action ) {
    default:
        $categories = Category::newInstance()->toTreeAll();
        $selected = array();
        foreach($categories as $c) {
            $selected[] = $c['pk_i_id'];
            foreach($c['categories'] as $cc) {
                $selected[] = $cc['pk_i_id'];
            }
        }
        $this->_exportVariableToView('categories', $categories);
        $this->_exportVariableToView('default_selected', $selected);
        $this->_exportVariableToView('fields', $this->fieldManager->listAll());
        $this->doView("fields/index.php");
        break;
    }
}

//hopefully generic...
function doView($file)
{
    osc_run_hook("before_admin_html");
    osc_current_admin_theme_path($file);
    Session::newInstance()->_clearVariables();
    osc_run_hook("after_admin_html");
}

}

/* file end: ./oc-admin/custom_fields.php */
?>

<?php if ( ! defined('ABS_PATH')) exit('ABS_PATH is not loaded. Direct access is not allowed
');

/*
 *      Oclass      software for creating and publishing online classified
 *                  advertising platforms
 *
 *                  Copyright (C) 2012 OSCLASS
 *
 *      This program is free software: you can redistribute it and/or
 *      modify it under the terms of the GNU Affero General Public License
 *      as published by the Free Software Foundation, either version 3 of
 *      the License, or (at your option) any later version.
 *
 *      This program is distributed in the hope that it will be useful, but
 *      WITHOUT ANY WARRANTY; without even the implied warranty of
 *      MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 *      GNU Affero General Public License for more details.
 *
 *      You should have received a copy of the GNU Affero General Public
 *      License along with this program. If not, see <http://www.gnu.org/licenses/>.
 */

class CAdminEmails extends AdminSecBaseModel
{
    //specific for this class
    private $emailManager;

    function __construct()
    {
        parent::__construct();

        //specific things for this class
        $this->emailManager = Page::newInstance();
    }

    //Business Layer...
    function doModel()
    {
        parent::doModel();

        //specific things for this class
        switch($this->action) {

            case 'edit':
                if(Params::getParam("id")=='') {
                    $this->redirectTo(osc_admin_base_url(true)."?page=emails");
                }

                $form = count(Session::newInstance()->getForm());
                $keepForm = count(Session::newInstance()->getKeepForm());
                if($form == 0 || $form == $keepForm) {
                    Session::newInstance()->_dropKeepForm();
                }

                $this->_exportVariableToView("email", $this->emailManager->
                    findByPrimaryKey(Params::getParam("id")));
                $this->doView("emails/frm.php");
                break;

            case 'edit_post':
                osc_csrf_check();
                $id = Params::getParam("id");
                $s_internal_name = Params::getParam("s_internal_name");

                $aFieldsDescription = array();
                $postParams = Params::getParamsAsArray('', false);
                $not_empty = false;
                foreach ($postParams as $k => $v) {

```

```

        if(preg_match('|(.+?)#(.+)|', $k, $m)) {
            if($m[2]== 's_title' && $v!='') { $not_empty = true; };
            $aFieldsDescription[$m[1]][$m[2]] = $v;
        }
    }

    Session::newInstance()->_setForm('s_internal_name', $s_internal_name);
    Session::newInstance()->_setForm('aFieldsDescription', $aFieldsDescription);
    ;

    if($not_empty) {
        foreach($aFieldsDescription as $k => $_data) {
            $this->emailManager->updateDescription($id, $k, $_data['s_title'],
                $_data['s_text']);
        }

        if(!$this->emailManager->internalNameExists($id, $s_internal_name)) {
            if(!$this->emailManager->isIndelible($id)) {
                $this->emailManager->updateInternalName($id, $s_internal_name);
            }
        }
        Session::newInstance()->_clearVariables();
        osc_add_flash_ok_message(_m('The email/alert has been updated'),
            'admin');
        $this->redirectTo(osc_admin_base_url(true)."?page=emails");
    }
    osc_add_flash_error_message(_m('You can\'t repeat internal name'),
        'admin');
} else {
    osc_add_flash_error_message(_m('The email couldn\'t be updated, at
        least one title should not be empty'), 'admin');
}
$this->redirectTo(osc_admin_base_url(true)."?page=emails&action=edit&id="
    . $id);
break;
default:
    //-
    if( Params::getParam('iDisplayLength') == '' ) {
        Params::setParam('iDisplayLength', 10 );
    }

    $p_iPage = 1;
    if( is_numeric(Params::getParam('iPage')) && Params::getParam('iPage') >=
        1 ) {
        $p_iPage = Params::getParam('iPage');
    }
    Params::setParam('iPage', $p_iPage);

    $prefLocale = osc_current_admin_locale();
    $emails = $this->emailManager->listAll(1);

    // pagination
    $start = ($p_iPage-1) * Params::getParam('iDisplayLength');
    $limit = Params::getParam('iDisplayLength');
    $count = count( $emails );

    $displayRecords = $limit;
    if( ($start+$limit) > $count ) {
        $displayRecords = ($start+$limit) - $count;
    }
    // ----
    $aData = array();
    $smax = ($start+$limit);
    if($smax > $count) $smax = $count;
    for($i = $start; $i < $smax; $i++) {
        $email = $emails[$i];

        if(isset($email['locale'][$prefLocale]) && !empty($email['locale'][$prefLocale][ 's_title' ])) {
            $title = $email['locale'][$prefLocale];
        } else {
            $title = current($email['locale']);
        }
        $options = array();
        $options[] = '<a href="' . osc_admin_base_url(true) . "?page=emails&
            amp;action=edit&id=" . $email["pk_i_id"] . "'>' . --('Edit') .
            '</a>';

        $auxOptions = '<ul>'.PHP_EOL;
        foreach( $options as $actual ) {
            $auxOptions .= '<li>'. $actual . '</li>'.PHP_EOL;
        }
        $actions = '<div class="actions">'. $auxOptions . '</div>'.PHP_EOL;

        $row = array();
        $row[] = $email['s_internal_name'] . $actions;
        $row[] = $title['s_title'];
        $aData[] = $row;
    }
    // ----
    $array['iTotalRecords'] = $displayRecords;
    $array['iTotalDisplayRecords'] = count($emails);
    $array['iDisplayLength'] = $limit;
    $array['aData'] = $aData;

    $page = (int)Params::getParam('iPage');
    if(count($array['aData']) == 0 && $page!=1) {
        $total = (int)$array['iTotalDisplayRecords'];
        $smaxPage = ceil( $total / (int)$array['iDisplayLength'] );
    }

```

```

        $url = osc_admin_base_url(true).'?'.$_SERVER['QUERY_STRING'];

        if($maxPage==0) {
            $url = preg_replace('/&iPage=(\d)+/', '&iPage=1', $url);
            $this->redirectTo($url);
        }

        if($page > 1) {
            $url = preg_replace('/&iPage=(\d)+/', '&iPage='.$maxPage, $url);
            $this->redirectTo($url);
        }
    }

    $this->exportVariableToView('aEmails', $array);

    $this->doView("emails/index.php");
}
}

//hopefully generic...
function doView($file)
{
    osc_run_hook("before_admin_html");
    osc_current_admin_theme_path($file);
    Session::newInstance()->clearVariables();
    osc_run_hook("after_admin_html");
}
}

/* file end: ./oc-admin/emails.php */
?>

<?php
/*
 *      Oclass      software for creating and publishing online classified
 *                  advertising platforms
 *
 *                  Copyright (C) 2012 OSCLASS
 *
 *      This program is free software: you can redistribute it and/or
 *      modify it under the terms of the GNU Affero General Public License
 *      as published by the Free Software Foundation, either version 3 of
 *      the License, or (at your option) any later version.
 *
 *      This program is distributed in the hope that it will be useful, but
 *      WITHOUT ANY WARRANTY; without even the implied warranty of
 *      MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 *      GNU Affero General Public License for more details.
 *
 *      You should have received a copy of the GNU Affero General Public
 *      License along with this program. If not, see <http://www.gnu.org/licenses/>.
 */

define('ABS_PATH', str_replace('\', '/', dirname(dirname($_SERVER['SCRIPT_FILENAME']))) .
'/');
define('OC_ADMIN', true);

require_once ABS_PATH . 'oc-load.php';

if( file_exists(ABS_PATH . '.maintenance') ) {
    define('_OSC_MAINTENANCE_', true);
}

// register admin scripts
osc_register_script('admin-osc', osc_current_admin_theme_js_url('osc.js'), 'jquery');
osc_register_script('admin-ui-osc', osc_current_admin_theme_js_url('ui-osc.js'), 'jquery');
;
osc_register_script('admin-location', osc_current_admin_theme_js_url('location.js'), '
jquery');

// enqueue scripts
osc_enqueue_script('jquery');
osc_enqueue_script('jquery-ui');
osc_enqueue_script('admin-osc');
osc_enqueue_script('admin-ui-osc');

osc_add_hook('admin_footer', array('FieldForm', 'i18n_datePicker'));

// enqueue css styles
osc_enqueue_style('jquery-ui', osc_assets_url('css/jquery-ui/jquery-ui.css'));
osc_enqueue_style('admin-css', osc_current_admin_theme_styles_url('main.css'));

switch( Params::getParam('page') )
{
    case('items'):
        require_once(osc_admin_base_path() . 'items.php');
        $do = new CAdminItems();
        $do->doModel();

        break;
    case('comments'):
        require_once(osc_admin_base_path() . 'comments.php');
        $do = new CAdminItemComments();
        $do->doModel();

        break;
    case('media'):
        require_once(osc_admin_base_path() . 'media.php');
        $do = new CAdminMedia();
        $do->doModel();

        break;
    case('login'):
        require_once(osc_admin_base_path() . 'login.php');
        $do = new CAdminLogin();

```

```

        $do->doModel();
break;
case('categories'): require_once(osc_admin_base_path() . 'categories.php');
                    $do = new CAdminCategories();
                    $do->doModel();

break;
case('emails'):     require_once(osc_admin_base_path() . 'emails.php');
                    $do = new CAdminEmails();
                    $do->doModel();

break;
case('pages'):      require_once(osc_admin_base_path() . 'pages.php');
                    $do = new CAdminPages();
                    $do->doModel();

break;
case('settings'):   require_once(osc_admin_base_path() . 'settings.php');
                    $do = new CAdminSettings();
                    $do->doModel();

break;
case('plugins'):    require_once(osc_admin_base_path() . 'plugins.php');
                    $do = new CAdminPlugins();
                    $do->doModel();

break;
case('languages'):  require_once(osc_admin_base_path() . 'languages.php');
                    $do = new CAdminLanguages();
                    $do->doModel();

break;
case('admins'):     require_once(osc_admin_base_path() . 'admins.php');
                    $do = new CAdminAdmins();
                    $do->doModel();

break;
case('users'):      require_once(osc_admin_base_path() . 'users.php');
                    $do = new CAdminUsers();
                    $do->doModel();

break;
case('ajax'):       require_once(osc_admin_base_path() . 'ajax/ajax.php');
                    $do = new CAdminAjax();
                    $do->doModel();

break;
case('appearance'): require_once(osc_admin_base_path() . 'appearance.php');
                    $do = new CAdminAppearance();
                    $do->doModel();

break;
case('tools'):      require_once(osc_admin_base_path() . 'tools.php');
                    $do = new CAdminTools();
                    $do->doModel();

break;
case('stats'):      require_once(osc_admin_base_path() . 'stats.php');
                    $do = new CAdminStats();
                    $do->doModel();

break;
case('cfields'):    require_once(osc_admin_base_path() . 'custom_fields.php');
                    $do = new CAdminCFields();
                    $do->doModel();

break;
case('upgrade'):    require_once(osc_admin_base_path() . 'upgrade.php');
                    $do = new CAdminUpgrade();
                    $do->doModel();

break;
case('market'):     require_once(osc_admin_base_path() . 'market.php');
                    $do = new CAdminMarket();
                    $do->doModel();

break;
default:           //login of oc-admin
                    require_once(osc_admin_base_path() . 'main.php');
                    $do = new CAdminMain();
                    $do->doModel();
}

/* file end: ./oc-admin/index.php */

<?php if ( ! defined('ABS_PATH')) exit('ABS_PATH is not loaded. Direct access is not allowed
. ');

/*
 *      Oclass      software for creating and publishing online classified
 *                  advertising platforms
 *
 *                  Copyright (C) 2012 OSCLASS
 *
 *      This program is free software: you can redistribute it and/or
 *      modify it under the terms of the GNU Affero General Public License
 *      as published by the Free Software Foundation, either version 3 of
 *      the License, or (at your option) any later version.
 *
 *      This program is distributed in the hope that it will be useful, but
 *      WITHOUT ANY WARRANTY; without even the implied warranty of
 *      MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 *      GNU Affero General Public License for more details.
 *
 *      You should have received a copy of the GNU Affero General Public
 *      License along with this program. If not, see <http://www.gnu.org/licenses/>.
 */

class CAdminItems extends AdminSecBaseModel
{
    //specific for this class
    private $itemManager;

    function __construct()

```

```

{
    parent::__construct();

    //specific things for this class
    $this->itemManager = Item::newInstance();
}

//Business Layer...
function doModel()
{
    parent::doModel();

    if(osc_is_moderator() && ($this->action=='settings' || $this->action=='
        settings_post')) {
        osc_add_flash_error_message(_m("You don't have enough permissions"), "admin");
        $this->redirectTo(osc_admin_base_url());
    }

    //specific things for this class
    switch ($this->action)
    {
        case 'bulk_actions':
            osc_csrf_check();
            $mItems = new ItemActions( true );
            switch ( Params::getParam('bulk_actions') )
            {
                case 'enable_all':
                    $id = Params::getParam('id');
                    if ($id) {
                        $numSuccess = 0;
                        foreach ($id as $_id) {
                            if( $mItems->enable($_id) ) {
                                $numSuccess++;
                            }
                        }
                        osc_add_flash_ok_message( sprintf(_mn('%d
                            listing has been enabled', '%d
                            listings have been enabled',
                            $numSuccess), $numSuccess), 'admin');
                    }
                    break;
                case 'disable_all':
                    $id = Params::getParam('id');
                    if ($id) {
                        $numSuccess = 0;
                        foreach ($id as $_id) {
                            if( $mItems->disable((int)$_id) ) {
                                $numSuccess++;
                            }
                        }
                        osc_add_flash_ok_message( sprintf(_mn('%d
                            listing has been disabled', '%d
                            listings have been disabled',
                            $numSuccess), $numSuccess), 'admin');
                    }
                    break;
                case 'activate_all':
                    $id = Params::getParam('id');
                    if ($id) {
                        $numSuccess = 0;
                        foreach ($id as $_id) {
                            if( $mItems->activate($_id) ) {
                                $numSuccess++;
                            }
                        }
                        osc_add_flash_ok_message( sprintf(_mn('%d
                            listing has been activated', '%d
                            listings have been activated',
                            $numSuccess), $numSuccess), 'admin');
                    }
                    break;
                case 'deactivate_all':
                    $id = Params::getParam('id');
                    if ($id) {
                        $numSuccess = 0;
                        foreach ($id as $_id) {
                            if( $mItems->deactivate($_id) ) {
                                $numSuccess++;
                            }
                        }
                        osc_add_flash_ok_message( sprintf(_m('%d
                            listing has been deactivated', '%d
                            listings have been deactivated',
                            $numSuccess), $numSuccess), 'admin');
                    }
                    break;
                case 'premium_all':
                    $id = Params::getParam('id');
                    if ($id) {
                        $numSuccess = 0;
                        foreach ($id as $_id) {
                            if( $mItems->premium($_id) ) {
                                $numSuccess++;
                            }
                        }
                        osc_add_flash_ok_message( sprintf(_mn('%d
                            listing has been marked as premium', '%d
                            listings have been marked as premium',
                            $numSuccess), $numSuccess), 'admin

```



```

        ');
    }
    break;
    case 'depremium_all ':
        $id = Params::getParam('id');
        if ($id) {
            $numSuccess = 0;
            foreach ($id as $_id) {
                if( $mItems->premium($_id, false) ) {
                    $numSuccess++;
                }
            }
            osc_add_flash_ok_message( sprintf(_mn('%d
            change has been made', '%d changes
            have been made', $numSuccess) ,
            $numSuccess), 'admin');
        }
    }
    break;
    case 'spam_all ':
        $id = Params::getParam('id');
        if($id) {
            $numSuccess = 0;
            foreach ($id as $_id) {
                if( $mItems->spam($_id) ) {
                    $numSuccess++;
                }
            }
            osc_add_flash_ok_message( sprintf(_mn('%d
            listing has been marked as spam', '%d
            listings have been marked as spam',
            $numSuccess), $numSuccess), 'admin');
        }
    }
    break;
    case 'despam_all ':
        $id = Params::getParam('id');
        if ($id) {
            $numSuccess = 0;
            foreach ($id as $_id) {
                if( $mItems->spam($_id, false) ) {
                    $numSuccess++;
                }
            }
            osc_add_flash_ok_message( sprintf(_mn('%d
            change has been made', '%d changes
            have been made', $numSuccess),
            $numSuccess), 'admin');
        }
    }
    break;
    case 'delete_all ':
        $id = Params::getParam('id');
        $success = false;

        if($id) {
            $numSuccess = 0;
            foreach($id as $i) {
                if ($i) {
                    $item = $this->itemManager->
                    findByPrimaryKey($i);
                    $success = $mItems->delete($item['
                    s_secret'], $item['pk-i-id']);
                    if($success) {
                        $numSuccess++;
                    }
                }
            }
            osc_add_flash_ok_message( sprintf(_mn('%d
            listing has been deleted', '%d
            listings have been deleted',
            $numSuccess), $numSuccess), 'admin');
        }
    }
    break;
    case 'clear_spam_all ':
        $id = Params::getParam('id');
        $success = false;

        if($id) {
            $numSuccess = 0;
            foreach($id as $i) {
                if ($i) {
                    $success = $this->itemManager->
                    clearStat($i, 'spam');
                    if($success) {
                        $numSuccess++;
                    }
                }
            }
            osc_add_flash_ok_message( sprintf(_mn('%d
            listing has been unmarked as spam', '%d
            listings have been unmarked as spam',
            $numSuccess), $numSuccess), 'admin');
        }
    }
    break;
    case 'clear_bad_all ':
        $id = Params::getParam('id');
        $success = false;

        if($id) {

```

```

$numSuccess = 0;
foreach($id as $i) {
    if ($i) {
        $success = $this->itemManager->
            clearStat($i , 'bad' );
        if($success) {
            $numSuccess++;
        }
    }
}
osc_add_flash_ok_message( sprintf(_mn('%d
listing has been unmarked as
missclassified ', '%d listings have
been unmarked as missclassified ',
$numSuccess), $numSuccess), 'admin');
}
break;
case 'clear_dupl_all';
$id = Params::getParam('id');
$success = false;

if($id) {
    $numSuccess = 0;
    foreach($id as $i) {
        if ($i) {
            $success = $this->itemManager->
                clearStat($i , 'duplicated' );
            if($success) {
                $numSuccess++;
            }
        }
    }
}
osc_add_flash_ok_message( sprintf(_mn('%d
listing has been unmarked as
duplicated ', '%d listings have been
unmarked as duplicated ', $numSuccess),
$numSuccess), 'admin');
}
break;
case 'clear_expi_all';
$id = Params::getParam('id');
$success = false;

if($id) {
    $numSuccess = 0;
    foreach($id as $i) {
        if ($i) {
            $success = $this->itemManager->
                clearStat($i , 'expired' );
            if($success) {
                $numSuccess++;
            }
        }
    }
}
osc_add_flash_ok_message( sprintf(_mn('%d
listing has been unmarked as expired ',
'%d listings have been unmarked as
expired ', $numSuccess), $numSuccess),
'admin');
}
break;
case 'clear_offe_all';
$id = Params::getParam('id');
$success = false;

if($id) {
    $numSuccess = 0;
    foreach($id as $i) {
        if ($i) {
            $success = $this->itemManager->
                clearStat($i , 'offensive' );
            if($success) {
                $numSuccess++;
            }
        }
    }
}
osc_add_flash_ok_message( sprintf(_mn('%d
listing has been unmarked as offensive
', '%d listings have been unmarked as
offensive ', $numSuccess), $numSuccess),
'admin');
}
break;
case 'clear_all';
$id = Params::getParam('id');
$success = false;

if($id) {
    $numSuccess = 0;
    foreach($id as $i) {
        if ($i) {
            $success = $this->itemManager->
                clearStat($i , 'all' );
            if($success) {
                $numSuccess++;
            }
        }
    }
}
osc_add_flash_ok_message( sprintf(_mn('%d

```

```

        listing has been unmarked', '%d
        listings have been unmarked',
        $numSuccess), $numSuccess), 'admin');
    }
    break;
default:
    if(Params::getParam("bulk_actions")!="") {
        osc_run_hook("item_bulk".Params::getParam(
            ("bulk_actions"), Params::getParam('id
            '));
    }
    break;
}
$this->redirectTo( $_SERVER['HTTP_REFERER'] );

break;
case 'delete':
//delete
osc_csrf_check();
$id = Params::getParam('id');
$success = false;

foreach( $id as $i ) {
    if ( $i ) {
        $item = $this->itemManager->
            findByPrimaryKey( $i );
        $mItems = new ItemActions( true );
        $success = $mItems->delete( $item['s_secret
            '], $item['pk_iid'] );
    }
}

if( $success ) {
    osc_add_flash_ok_message( _m('The listing has been
        deleted'), 'admin');
} else {
    osc_add_flash_error_message( _m("The listing
        couldn't be deleted"), 'admin');
}

$this->redirectTo( $_SERVER['HTTP_REFERER'] );

break;
case 'status':
//status
osc_csrf_check();
$id = Params::getParam('id');
$value = Params::getParam('value');

if (!$id)
    return false;

$id = (int) $id;

if (!is_numeric($id))
    return false;

if (!in_array($value, array('ACTIVE', 'INACTIVE', '
    ENABLE', 'DISABLE')))
    return false;

$item = $this->itemManager->findByPrimaryKey($id);
$mItems = new ItemActions( true );

switch ( $value ) {
    case 'ACTIVE':

        $success = $mItems->activate( $id );
        if( $success && $success > 0 ) {
            osc_add_flash_ok_message( _m('The listing
                has been activated'), 'admin');
        } else if ( !$success ){
            osc_add_flash_error_message( _m('An error
                has occurred'), 'admin');
        } else {
            osc_add_flash_error_message( _m("The
                listing can't be activated because it'
                s blocked"), 'admin');
        }

        break;
    case 'INACTIVE':

        $success = $mItems->deactivate( $id );
        if( $success && $success > 0 ) {
            osc_add_flash_ok_message( _m('The listing
                has been deactivated'), 'admin');
        } else {
            osc_add_flash_error_message( _m('An error
                has occurred'), 'admin');
        }

        break;
    case 'ENABLE':

        $success = $mItems->enable( $id );
        if( $success && $success > 0 ) {
            osc_add_flash_ok_message( _m('The listing
                has been enabled'), 'admin');
        } else {
            osc_add_flash_error_message( _m('An error
                has occurred'), 'admin');
        }

}

```

```

        break;
        case 'DISABLE':
            $success = $mItems->disable( $id );
            if( $success && $success > 0 ) {
                osc_add_flash_ok_message( _m('The listing
                    has been disabled'), 'admin');
            } else {
                osc_add_flash_error_message( _m('An error
                    has occurred'), 'admin');
            }
            break;
    }
    $this->redirectTo( $_SERVER['HTTP_REFERER'] );
break;
case 'status-premium': //status premium
    osc_csrf_check();
    $id = Params::getParam('id');
    $value = Params::getParam('value');

    if (!$id)
        return false;

    $id = (int) $id;

    if (!is_numeric($id))
        return false;

    if (!in_array($value, array(0, 1)))
        return false;

    $mItems = new ItemActions(true);

    if ($mItems->premium($id, $value==1?true:false) ) {
        osc_add_flash_ok_message( _m('Changes have been
            applied'), 'admin');
    } else {
        osc_add_flash_error_message( _m('An error has
            occurred'), 'admin');
    }

    $this->redirectTo( $_SERVER['HTTP_REFERER'] );
break;
case 'status-spam': //status spam
    osc_csrf_check();
    $id = Params::getParam('id');
    $value = Params::getParam('value');

    if (!$id)
        return false;

    $id = (int) $id;

    if (!is_numeric($id))
        return false;

    if (!in_array($value, array(0, 1)))
        return false;

    $mItems = new ItemActions(true);

    if( $mItems->spam($id, $value==1?true:false) ){
        osc_add_flash_ok_message( _m('Changes have been
            applied'), 'admin');
    } else {
        osc_add_flash_error_message( _m('An error has
            occurred'), 'admin');
    }

    $this->redirectTo( $_SERVER['HTTP_REFERER'] );
break;
case 'clear-stat':

    osc_csrf_check();
    $id = Params::getParam('id');
    $stat = Params::getParam('stat');

    if (!$id)
        return false;

    if (!$stat)
        return false;

    $id = (int) $id;

    if (!is_numeric($id))
        return false;

    $success = $this->itemManager->clearStat($id, $stat)
        ;

    if($success) {
        osc_add_flash_ok_message( _m('The listing has been
            unmarked as')." $stat", 'admin');
    } else {
        osc_add_flash_error_message( _m("The listing hasn'
            t been unmarked as")." $stat", 'admin');
    }

```

```

}
$this->redirectTo( $_SERVER['HTTP_REFERER'] );
break;
case 'item_edit ':
    // edit item
    $id = Params::getParam('id');

    $item = Item::newInstance()->findByPrimaryKey($id);
    if (count($item) <= 0) {
        $this->redirectTo( osc_admin_base_url(true) . "?
            page=items" );
    }

    $csrf_token = osc_csrf_token_url();
    if( $item['b_active'] ) {
        $actions[] = '<a class="btn float-left" href="'.
            osc_admin_base_url(true).'? page=items&
            action=status&id='.$item['pk_i_id'].'&
            '.$csrf_token.'&value=INACTIVE">'.--('
            Deactivate') .'</a>';
    } else {
        $actions[] = '<a class="btn btn-red float-left"
            href="'.osc_admin_base_url(true).'? page=items&
            amp;action=status&id='.$item['pk_i_id'].'&
            amp;'.$csrf_token.'&value=ACTIVE">'.--('
            Activate') .'</a>';
    }
    if( $item['b_enabled'] ) {
        $actions[] = '<a class="btn float-left" href="'.
            osc_admin_base_url(true).'? page=items&
            action=status&id='.$item['pk_i_id'].'&
            '.$csrf_token.'&value=DISABLE">'.--('
            Block') .'</a>';
    } else {
        $actions[] = '<a class="btn btn-red float-left"
            href="'.osc_admin_base_url(true).'? page=items&
            amp;action=status&id='.$item['pk_i_id'].'&
            amp;'.$csrf_token.'&value=ENABLE">'.--('
            Unblock') .'</a>';
    }
    if( $item['b_premium'] ) {
        $actions[] = '<a class="btn float-left" href="'.
            osc_admin_base_url(true).'? page=items&
            action=status_premium&id='.$item['pk_i_id']
            '.'&'.$csrf_token.'&value=0">'.--('
            Unmark as premium') .'</a>';
    } else {
        $actions[] = '<a class="btn float-left" href="'.
            osc_admin_base_url(true).'? page=items&
            action=status_premium&id='.$item['pk_i_id']
            '.'&'.$csrf_token.'&value=1">'.--('
            Mark as premium') .'</a>';
    }
    if( $item['b_spam'] ) {
        $actions[] = '<a class="btn btn-red float-left"
            href="'.osc_admin_base_url(true).'? page=items&
            amp;action=status_spam&id='.$item['pk_i_id']
            '.'&'.$csrf_token.'&value=0">'.--('
            Unmark as spam') .'</a>';
    } else {
        $actions[] = '<a class="btn float-left" href="'.
            osc_admin_base_url(true).'? page=items&
            action=status_spam&id='.$item['pk_i_id']
            '.'&'.$csrf_token.'&value=1">'.--('
            Mark as spam') .'</a>';
    }
}

$this->exportVariableToView(" actions", $actions);

$form = count(Session::newInstance()->getForm());
$keepForm = count(Session::newInstance()->getKeepForm
());

if($form==0 || $form==$keepForm) {
    Session::newInstance()->dropKeepForm();
}

// save referer if belongs to manage items
// redirect only if ManageItems or ReportedListngs
if( isset($_SERVER['HTTP_REFERER']) ) {
    $referer = $_SERVER['HTTP_REFERER'];
    if(preg_match('/page=items/', $referer) ) {
        if(preg_match("/action=(\\p{L}|-|+)+/u",
            $referer, $matches) ) {
            if( $matches[1] == 'items-reported' ) {
                Session::newInstance()->_set( '
                    osc_admin_referer', $referer );
            }
        } else {
            // no actions - Manage Listings
            Session::newInstance()->_set( '
                osc_admin_referer', $referer );
        }
    }
}

$this->exportVariableToView(" item", $item);
$this->exportVariableToView(" new_item", FALSE);

```

```

        osc_run_hook("before_item_edit", $item);
        $this->doView('items/frm.php');

break;
case 'item_edit_post ':
    osc_csrf_check();
    $mItems = new ItemActions(true);

    $mItems->prepareData(false);
    // set all parameters into session
    foreach( $mItems->data as $key => $value ) {
        Session::newInstance()->_setForm($key, $value);
    }

    $meta = Params::getParam('meta');
    if(is_array($meta)) {
        foreach( $meta as $key => $value ) {
            Session::newInstance()->_setForm('meta_'.$key,
                $value);
            Session::newInstance()->_keepForm('meta_'.$key);
        }
    }

    $success = $mItems->edit();

    if($success==1){
        osc_add_flash_ok_message( _m('Changes saved
            correctly'), 'admin');
        $url = osc_admin_base_url(true) . "?page=items";
        // if Referer is saved that means referer is
        ManageListings or ReportListings
        if(Session::newInstance()->_get('osc_admin_referer')
            != '') {
            $url = Session::newInstance()->_get('
                osc_admin_referer');
        }
        Session::newInstance()->_clearVariables();
        if(is_array($meta)) {
            foreach( $meta as $key => $value ) {
                Session::newInstance()->_dropKeepForm('
                    meta_'.$key);
            }
        }

        $this->redirectTo( $url );
    } else {
        osc_add_flash_error_message( $success, 'admin');
        $this->redirectTo( osc_admin_base_url(true) . "?
            page=items&action=item_edit&id=" . Params::
            getParam('id') );
    }

break;
case 'deleteResource ': //delete resource
    osc_csrf_check();
    $id = Params::getParam('id');
    $name = Params::getParam('name');
    $fkid = Params::getParam('fkid');

    // delete files
    osc_deleteResource($id, true);
    Log::newInstance()->insertLog('items', 'deleteResource',
        $id, $id, 'admin', osc_logged_admin_id());

    $result = ItemResource::newInstance()->delete(array('
        pk.i.id' => $id, 'fk.i.item.id' => $fkid, 's.name'
        => $name));
    if($result == false) {
        osc_add_flash_error_message( _m('An error has
            occurred'), 'admin');
    } else {
        osc_add_flash_ok_message( _m('Resource deleted'),
            'admin');
    }
    $this->redirectTo( osc_admin_base_url(true) . "?page=
        items" );

break;
case 'post ':
    // add item
    $form = count(Session::newInstance()->_getForm());
    $keepForm = count(Session::newInstance()->_getKeepForm(
        ));
    if($form == 0 || $form == $keepForm) {
        Session::newInstance()->_dropKeepForm();
    }

    $this->_exportVariableToView("new_item", TRUE);
    osc_run_hook('post_item');
    $this->doView('items/frm.php');

break;
case 'post_item ':
    //post item
    osc_csrf_check();
    $mItem = new ItemActions(true);

    $mItem->prepareData(true);
    // set all parameters into session
    foreach( $mItem->data as $key => $value ) {
        Session::newInstance()->_setForm($key, $value);
    }

    $meta = Params::getParam('meta');

```

```

        if(is_array($meta)) {
            foreach( $meta as $key => $value ) {
                Session::newInstance()->_setForm('meta_'. $key ,
                    $value);
                Session::newInstance()->_keepForm('meta_'. $key
                    );
            }
        }
        $success = $mItem->add();

        if( $success==1 || $success==2 ) {
            $url = osc_admin_base_url(true) . "?page=items";
            // if Referer is saved that means referer is
            // ManageListings or ReportListings
            if(Session::newInstance()->_get('osc_admin_referer')
                != '') {
                $url = Session::newInstance()->_get('
                    osc_admin_referer');
                Session::newInstance()->_drop('
                    osc_admin_referer');
            }
            Session::newInstance()->_clearVariables();
            if(is_array($meta)) {
                foreach( $meta as $key => $value ) {
                    Session::newInstance()->_dropKeepForm('
                        meta_'. $key);
                }
            }
            osc_add_flash_ok_message( _m('A new listing has
                been added'), 'admin');

            $this->redirectTo( $url );
        } else {
            osc_add_flash_error_message( $success, 'admin');
            $this->redirectTo( osc_admin_base_url(true) . "?
                page=items&action=post" );
        }

        break;
        case('settings') :
            // calling the items settings view
            $this->doView('items/settings.php');

        break;
        case('settings_post') :
            // update item settings
            osc_csrf_check();
            $iUpdated = 0;
            $enabledRecaptchaItems = Params::getParam('
                enabled_recaptcha_items');
            $enabledRecaptchaItems = (($enabledRecaptchaItems
                == '1') ? true : false);
            $moderateItems = Params::getParam('
                moderate_items');
            $moderateItems = (($moderateItems != '')
                ? true : false);
            $numModerateItems = Params::getParam('
                num_moderate_items');
            $itemsWaitTime = Params::getParam('
                items_wait_time');
            $loggedUserItemValidation = Params::getParam('
                logged_user_item_validation');
            $loggedUserItemValidation = ((
                $loggedUserItemValidation != '') ? true : false);
            $regUserPost = Params::getParam('
                reg_user_post');
            $regUserPost = (($regUserPost != '') ?
                true : false);
            $notifyNewItem = Params::getParam('
                notify_new_item');
            $notifyNewItem = (($notifyNewItem != '')
                ? true : false);
            $notifyContactItem = Params::getParam('
                notify_contact_item');
            $notifyContactItem = (($notifyContactItem !=
                '') ? true : false);
            $notifyContactFriends = Params::getParam('
                notify_contact_friends');
            $notifyContactFriends = (($notifyContactFriends
                != '') ? true : false);
            $enabledFieldPriceItems = Params::getParam('
                enable_field_price@items');
            $enabledFieldPriceItems = ((
                $enabledFieldPriceItems != '') ? true : false);
            $enabledFieldImagesItems = Params::getParam('
                enable_field_images@items');
            $enabledFieldImagesItems = ((
                $enabledFieldImagesItems != '') ? true : false);
            $numImagesItems = Params::getParam('
                num_images@items');
            if($numImagesItems=='') { $numImagesItems = 0; }
            $regUserCanContact = Params::getParam('
                reg_user_can_contact');
            $regUserCanContact = (($regUserCanContact !=
                '') ? true : false);
            $contactItemAttachment = Params::getParam('
                item_attachment');
            $contactItemAttachment = (($contactItemAttachment
                != '') ? true : false);
            $warnExpiration = Params::getParam('
                warn_expiration');

```

```

SwarnExpiration          = (int) $warnExpiration;

$msg = '';
if (!osc_validate_int (Params::getParam(" items_wait_time
"))) {
    $msg .= _m("Wait time must only contain numeric
characters")."<br/>";
}
if (Params::getParam(" num_moderate_items")!='' && !
osc_validate_int (Params::getParam("
num_moderate_items"))) {
    $msg .= _m("Number of moderated listings must only
contain numeric characters")."<br/>";
}
if (!osc_validate_int ($numImagesItems)) {
    $msg .= _m("Images per listing must only contain
numeric characters")."<br/>";
}
if (!osc_validate_int ($warnExpiration)) {
    $msg .= _m("Number of expiration days has to be a
numeric value")."<br/>";
}
if ($msg!='') {
    osc_add_flash_error_message( $msg, 'admin');
    $this->redirectTo(osc_admin_base_url(true) . '?
page=items&action=settings ');
}

$iUpdated += osc_set_preference('
enabled_recaptcha_items', $enabledRecaptchaItems);
if ($moderateItems) {
    $iUpdated += osc_set_preference('moderate_items',
$numModerateItems);
} else {
    $iUpdated += osc_set_preference('moderate_items',
'-1');
}
$iUpdated += osc_set_preference('
logged_user_item_validation',
$loggedUserItemValidation);
$iUpdated += osc_set_preference('reg-user-post',
$regUserPost);
$iUpdated += osc_set_preference('notify_new_item',
$notifyNewItem);
$iUpdated += osc_set_preference('notify_contact_item',
$notifyContactItem);
$iUpdated += osc_set_preference('
notify_contact_friends', $notifyContactFriends);
$iUpdated += osc_set_preference('enableField#
f_price@items', $enabledFieldPriceItems);
$iUpdated += osc_set_preference('enableField#
images@items', $enabledFieldImagesItems);
$iUpdated += osc_set_preference('items_wait_time',
$itemsWaitTime);
$iUpdated += osc_set_preference('numImages@items',
$numImagesItems);
$iUpdated += osc_set_preference('reg-user-can-contact',
$regUserCanContact);
$iUpdated += osc_set_preference('item_attachment',
$contactItemAttachment);
$iUpdated += osc_set_preference('warn_expiration',
$warnExpiration);

if ($iUpdated > 0) {
    osc_add_flash_ok_message( _m("Listings' settings
have been updated"), 'admin');
}
$this->redirectTo(osc_admin_base_url(true) . '?page=
items&action=settings ');

break;
case('items_reported'):

require_once osc_lib_path()."osclass/classes/
datatables/ItemsDataTable.php";

// set default iDisplayLength
if( Params::getParam('iDisplayLength') != '' ) {
    Cookie::newInstance()->push('
listing-iDisplayLength', Params::getParam('
iDisplayLength'));
    Cookie::newInstance()->set();
} else {
    // set a default value if it's set in the cookie
    if( Cookie::newInstance()->get_value('
listing-iDisplayLength') != '' ) {
        Params::setParam('iDisplayLength', Cookie::
newInstance()->get_value('
listing-iDisplayLength'));
    } else {
        Params::setParam('iDisplayLength', 10 );
    }
}
$this->exportVariableToView('iDisplayLength', Params
::getParam('iDisplayLength'));

```



```

// Table header order by related
if( Params::getParam('sort') == '' ) {
    Params::setParam('sort', 'date');
}
if( Params::getParam('direction') == '' ) {
    Params::setParam('direction', 'desc');
}

$page = (int)Params::getParam('iPage');
if($page==0) { $page = 1; };
Params::setParam('iPage', $page);

$params = Params::getParamsAsArray("get");

$itemsDataTable = new ItemsDataTable();
$itemsDataTable->tableReported($params);
$aData = $itemsDataTable->getData();

if(count($aData['aRows']) == 0 && $page!=1) {
    $total = (int)$aData['iTotalDisplayRecords'];
    $maxPage = ceil( $total / (int)$aData['iDisplayLength'] );

    $url = osc_admin_base_url(true).'?'.$_SERVER['QUERY_STRING'];

    if($maxPage==0) {
        $url = preg_replace('/&iPage=(\d)+/', '&iPage=1', $url);
        $this->redirectTo($url);
    }

    if($page > 1) {
        $url = preg_replace('/&iPage=(\d)+/', '&iPage='.$maxPage, $url);
        $this->redirectTo($url);
    }
}

$this->exportVariableToView('aData', $aData);
$this->exportVariableToView('aRawRows', $itemsDataTable->rawRows());

//calling the view...
$this->doView('items/reported.php');

break;
default:

// default
require_once osc_lib_path()."osclass/classes/datatables/ItemsDataTable.php";

// set default iDisplayLength
if( Params::getParam('iDisplayLength') != '' ) {
    Cookie::newInstance()->push('listing_iDisplayLength', Params::getParam('iDisplayLength'));
    Cookie::newInstance()->set();
} else {
    // set a default value if it's set in the cookie
    if( Cookie::newInstance()->get_value('listing_iDisplayLength') != '' ) {
        Params::setParam('iDisplayLength', Cookie::newInstance()->get_value('listing_iDisplayLength'));
    } else {
        Params::setParam('iDisplayLength', 10 );
    }
}
$this->exportVariableToView('iDisplayLength', Params::getParam('iDisplayLength'));

// Table header order by related
if( Params::getParam('sort') == '' ) {
    Params::setParam('sort', 'date');
}
if( Params::getParam('direction') == '' ) {
    Params::setParam('direction', 'desc');
}

$page = (int)Params::getParam('iPage');
if($page==0) { $page = 1; };
Params::setParam('iPage', $page);

$params = Params::getParamsAsArray("get");

$itemsDataTable = new ItemsDataTable();
$aData = $itemsDataTable->table($params);

if(count($aData['aRows']) == 0 && $page!=1) {
    $total = (int)$aData['iTotalDisplayRecords'];
    $maxPage = ceil( $total / (int)$aData['iDisplayLength'] );

    $url = osc_admin_base_url(true).'?'.$_SERVER['QUERY_STRING'];

    if($maxPage==0) {
        $url = preg_replace('/&iPage=(\d)+/', '&iPage=1', '&iPage=1', $url);
    }
}

```

```

        =1', $url);
        $this->redirectTo($url);
    }
    if($page > 1) {
        $url = preg_replace('/&iPage=(\d)+/', '&iPage
            ='.$maxPage, $url);
        $this->redirectTo($url);
    }
}

$this->_exportVariableToView('aData', $aData);
$this->_exportVariableToView('withFilters',
    $itemsDataTable->withFilters());
$this->_exportVariableToView('aRawRows',
    $itemsDataTable->rawRows());

$bulk_options = array(
    array('value' => '', 'data-dialog-content' => '',
        'label' => __('Bulk actions')),
    array('value' => 'delete_all', 'data-dialog-
        content' => sprintf(__('Are you sure you want
            to %s the selected listings?'), strtolower(
                __('Delete'))), 'label' => __('Delete')),
    array('value' => 'activate_all', 'data-dialog-
        content' => sprintf(__('Are you sure you want
            to %s the selected listings?'), strtolower(
                __('Activate'))), 'label' => __('Activate')),
    array('value' => 'deactivate_all', 'data-dialog-
        content' => sprintf(__('Are you sure you want
            to %s the selected listings?'), strtolower(
                __('Deactivate'))), 'label' => __('Deactivate')),
    array('value' => 'disable_all', 'data-dialog-
        content' => sprintf(__('Are you sure you want
            to %s the selected listings?'), strtolower(
                __('Block'))), 'label' => __('Block')),
    array('value' => 'enable_all', 'data-dialog-
        content' => sprintf(__('Are you sure you want
            to %s the selected listings?'), strtolower(
                __('Unblock'))), 'label' => __('Unblock')),
    array('value' => 'premium_all', 'data-dialog-
        content' => sprintf(__('Are you sure you want
            to %s the selected listings?'), strtolower(
                __('Mark as premium'))), 'label' => __('Mark as
                premium')),
    array('value' => 'depremium_all', 'data-dialog-
        content' => sprintf(__('Are you sure you want
            to %s the selected listings?'), strtolower(
                __('Unmark as premium'))), 'label' => __('Unmark
                as premium')),
    array('value' => 'spam_all', 'data-dialog-content'
        => sprintf(__('Are you sure you want to %s
            the selected listings?'), strtolower(
                __('Mark as spam'))), 'label' => __('Mark as spam')),
    array('value' => 'despam_all', 'data-dialog-
        content' => sprintf(__('Are you sure you want
            to %s the selected listings?'), strtolower(
                __('Unmark as spam'))), 'label' => __('Unmark as
                spam'))
);
$bulk_options = osc_apply_filter("item_bulk_filter",
    $bulk_options);
$this->_exportVariableToView('bulk_options',
    $bulk_options);

//calling the view...
$this->doView('items/index.php');
}
}

//hopefully generic...
function doView($file)
{
    osc_run_hook("before_admin_html");
    osc_current_admin_theme_path($file);
    Session::newInstance()->_clearVariables();
    osc_run_hook("after_admin_html");
}
}

/* file end: ./oc-admin/items.php */
?>

<?php if ( ! defined('ABS_PATH')) exit('ABS_PATH is not loaded. Direct access is not allowed
. ');

/*
 *   Oclass      software for creating and publishing online classified
 *               advertising platforms
 *
 *   Copyright (C) 2012 OSCLASS
 *
 *   This program is free software: you can redistribute it and/or
 *   modify it under the terms of the GNU Affero General Public License
 *   as published by the Free Software Foundation, either version 3 of
 *   the License, or (at your option) any later version.
 *
 */

```

```

*      This program is distributed in the hope that it will be useful, but
*      WITHOUT ANY WARRANTY; without even the implied warranty of
*      MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
*      GNU Affero General Public License for more details.
*
*      You should have received a copy of the GNU Affero General Public
*      License along with this program. If not, see <http://www.gnu.org/licenses/>.
*/

```

```

class CAdminLanguages extends AdminSecBaseModel
{
    //specific for this class
    private $localeManager;

    function __construct()
    {
        parent::__construct();

        //specific things for this class
        $this->localeManager = OSCLocale::newInstance();
    }

    //Business Layer...
    function doModel()
    {
        switch ($this->action) {
            case('add'):
                // calling add view
                $this->doView('languages/add.php');

                break;
            case('add_post'):
                // adding a new language
                if( defined('DEMO') ) {
                    osc_add_flash_warning_message( _m(" This action
                    can't be done because it's a demo site"),
                    'admin');
                    $this->redirectTo( osc_admin_base_url(true) .
                    '?page=languages' );
                }
                osc_csrf_check();
                $filePackage = Params::getFiles('package');

                if( isset($filePackage['size']) && $filePackage['size'] != 0 ) {
                    $path = osc_translations_path();
                    (int) $status = osc_unzip_file($filePackage['tmp-name'], $path);
                } else {
                    $status = 3;
                }

                switch ($status) {
                    case(0):
                        $msg = _m('The translation folder
                        is not writable');
                        osc_add_flash_error_message($msg,
                        'admin');

                        break;
                    case(1):
                        if( osc_checkLocales() ) {
                            $msg = _m('The language has
                            been installed correctly');
                            osc_add_flash_ok_message($msg,
                            'admin');
                        } else {
                            $msg = _m('There was a problem
                            adding the language');
                            osc_add_flash_error_message(
                            $msg, 'admin');
                        }

                        break;
                    case(2):
                        $msg = _m('The zip file is not
                        valid');
                        osc_add_flash_error_message($msg,
                        'admin');

                        break;
                    case(3):
                        $msg = _m('No file was uploaded');
                        osc_add_flash_warning_message($msg,
                        'admin');
                        $this->redirectTo(
                        osc_admin_base_url(true) . "?
                        page=languages&action=add");

                        break;
                    case(-1):
                        $msg = _m('There was a problem
                        adding the language');
                        osc_add_flash_error_message($msg,
                        'admin');

                        break;
                }

                $this->redirectTo( osc_admin_base_url(true) . "?
                page=languages' );

                break;
            case('edit'):
                // editing a language
                $sLocale = Params::getParam('id');
                if( !preg_match('/.{2}-.{2}/', $sLocale) ) {
                    osc_add_flash_error_message( _m('Language id
                    isn't in the correct format'), 'admin');
                    $this->redirectTo( osc_admin_base_url(true) .
                    '?page=languages' );
                }
            }
        }
    }
}

```

```

$ALocale = $this->localeManager->findByPrimaryKey(
    $sLocale);

if(count($ALocale) == 0) {
    osc_add_flash_error_message( _m('Language id
        doesn\'t exist'), 'admin');
    $this->redirectTo(osc_admin_base_url(true) .
        '?page=languages');
}

$this->exportVariableToView("aLocale", $ALocale);
$this->doView('languages/frm.php');

break;
case('edit_post'):
    // edit language post
    osc_csrf_check();
    $iUpdated = 0;
    $languageCode = Params::getParam('
        pk.c.code');
    $enabledWebstie = Params::getParam('
        b.enabled');
    $enabledBackoffice = Params::getParam('
        b.enabled_bo');
    $languageName = Params::getParam('s_name
        ');
    $languageShortName = Params::getParam('
        s.short_name');
    $languageDescription = Params::getParam('
        s.description');
    $languageCurrencyFormat = Params::getParam('
        s.currency_format');
    $languageDecPoint = Params::getParam('
        s.dec_point');
    $languageNumDec = Params::getParam('
        i.num_dec');
    $languageThousandsSep = Params::getParam('
        s.thousands_sep');
    $languageDateFormat = Params::getParam('
        s.date_format');
    $languageStopWords = Params::getParam('
        s.stop_words');

    // formatting variables
    if( !preg_match('/.{2}..{2}/', $languageCode) ) {
        osc_add_flash_error_message( _m('Language id
            isn\'t in the correct format'), 'admin');
        $this->redirectTo(osc_admin_base_url(true) .
            '?page=languages');
    }
    $enabledWebstie = ($enabledWebstie != '' ?
        true : false);
    $enabledBackoffice = ($enabledBackoffice !=
        '' ? true : false);
    $languageName = strip_tags($languageName
        );
    $languageName = trim($languageName);

    $languageShortName = strip_tags(
        $languageShortName);
    $languageShortName = trim($languageShortName
        );

    $languageDescription = strip_tags(
        $languageDescription);
    $languageDescription = trim(
        $languageDescription);

    $languageCurrencyFormat = strip_tags(
        $languageCurrencyFormat);
    $languageCurrencyFormat = trim(
        $languageCurrencyFormat);
    $languageDateFormat = strip_tags(
        $languageDateFormat);
    $languageDateFormat = trim($languageDateFormat
        );
    $languageStopWords = strip_tags(
        $languageStopWords);
    $languageStopWords = trim($languageStopWords
        );

    $msg = '';
    if(!osc_validate_text($languageName)) {
        $msg .= _m("Language name field is required")
            . "<br/>";
    }
    if(!osc_validate_text($languageShortName)) {
        $msg .= _m("Language short name field is
            required")."<br/>";
    }
    if(!osc_validate_text($languageDescription)) {
        $msg .= _m("Language description field is
            required")."<br/>";
    }
    if(!osc_validate_text($languageCurrencyFormat)) {
        $msg .= _m("Currency format field is required
            ")."<br/>";
    }
    if(!osc_validate_int($languageNumDec)) {

```

```

        $msg .= _m("Number of decimals must only
        contain numeric characters").<br/>";
    }
    if($msg!='') {
        osc_add_flash_error_message( $msg, 'admin');
        $this->redirectTo(osc_admin_base_url(true) .
            '?page=languages&action=edit&id=' .
            $languageCode);
    }

    $array = array('b_enabled' =>
        $enabledWebstie
        , 'b_enabled_bo' =>
        $enabledBackoffice
        , 's_name' =>
        $languageName
        , 's_short_name' =>
        $languageShortName
        , 's_description' =>
        $languageDescription
        , 's_currency_format' =>
        $languageCurrencyFormat
        , 's_dec_point' =>
        $languageDecPoint
        , 'i_num_dec' =>
        $languageNumDec
        , 's_thousands_sep' =>
        $languageThousandsSep
        , 's_date_format' =>
        $languageDateFormat
        , 's_stop_words' =>
        $languageStopWords);

    $iUpdated = $this->localeManager->update($array,
        array('pk_c_code' => $languageCode));
    if($iUpdated > 0) {
        osc_add_flash_ok_message(sprintf(_m('%s has
        been updated'), $languageShortName), '
        admin');
    }
    $this->redirectTo(osc_admin_base_url(true) . '?page=
    languages');

break;
case('enable_selected'):

    osc_csrf_check();
    $msg = _m('Selected languages have been
    enabled for the website');
    $iUpdated = 0;
    $aValues = array('b_enabled' => 1);

    $id = Params::getParam('id');

    if( !is_array($id) ) {
        osc_add_flash_warning_message( _m("The
        language ids aren't in the correct format
        "), 'admin');
        $this->redirectTo(osc_admin_base_url(true) .
            '?page=languages');
    }

    foreach( $id as $i ) {
        osc_translate_categories($i);
        $iUpdated += $this->localeManager->update(
            $aValues, array('pk_c_code' => $i));
    }

    if( $iUpdated > 0 ) {
        osc_add_flash_ok_message($msg, 'admin');
    }

    $this->redirectTo(osc_admin_base_url(true) . '?
    page=languages');

break;
case('disable_selected'):

    osc_csrf_check();
    $msg = _m('Selected languages have been
    disabled for the website');
    $msg_warning = '';
    $iUpdated = 0;
    $aValues = array('b_enabled' => 0);

    $id = Params::getParam('id');

    if( !is_array($id) ) {
        osc_add_flash_warning_message( _m("The
        language ids aren't in the correct format
        "), 'admin');
        $this->redirectTo(osc_admin_base_url(true) .
            '?page=languages');
    }

    foreach( $id as $i ) {
        if( osc_language() == $i ) {
            $msg_warning = sprintf(_m("%s can't be
            disabled because it's the default
            language"), osc_language());
            continue;
        }
        $iUpdated += $this->localeManager->update(

```

```

        $aValues , array('pk_c_code' => $i));
    }
    if( $msg_warning != '' ) {
        if( $iUpdated > 0 ) {
            osc_add_flash_warning_message($msg . '</p>
            <p>' . $msg_warning , 'admin');
        } else {
            osc_add_flash_warning_message($msg_warning
            , 'admin');
        }
    } else {
        osc_add_flash_ok_message($msg , 'admin');
    }

    $this->redirectTo(osc_admin_base_url(true) . '?
    page=languages');

break;
case('enable_bo_selected'):
    osc_csrf_check();
    $msg = _m('Selected languages have been
    enabled for the backoffice (oc-admin)');
    $iUpdated = 0;
    $aValues = array('b_enabled_bo' => 1);

    $id = Params::getParam('id');

    if( !is_array($id) ) {
        osc_add_flash_warning_message( _m("The
        language ids aren't in the correct format
        " , 'admin');
        $this->redirectTo(osc_admin_base_url(true) .
        '?page=languages');
    }

    foreach( $id as $i ) {
        osc_translate_categories($i);
        $iUpdated += $this->localeManager->update(
            $aValues , array('pk_c_code' => $i));
    }

    if( $iUpdated > 0 ) {
        osc_add_flash_ok_message($msg , 'admin');
    }

    $this->redirectTo(osc_admin_base_url(true) . '?
    page=languages');

break;
case('disable_bo_selected'):
    osc_csrf_check();
    $msg = _m('Selected languages have been
    disabled for the backoffice (oc-admin)');
    $msg_warning = '';
    $iUpdated = 0;
    $aValues = array('b_enabled_bo' => 0);

    $id = Params::getParam('id');

    if( !is_array($id) ) {
        osc_add_flash_warning_message( _m("The
        language ids aren't in the correct format
        " , 'admin');
        $this->redirectTo(osc_admin_base_url(true) .
        '?page=languages');
    }

    foreach( $id as $i ) {
        if( osc_language() == $i ) {
            $msg_warning = sprintf( _m("%s can't be
            disabled because it's the default
            language" , osc_language() );
            continue;
        }
        $iUpdated += $this->localeManager->update(
            $aValues , array('pk_c_code' => $i));
    }

    if( $msg_warning != '' ) {
        if( $iUpdated > 0 ) {
            osc_add_flash_warning_message($msg . '</p>
            <p>' . $msg_warning , 'admin');
        } else {
            osc_add_flash_warning_message($msg_warning
            , 'admin');
        }
    } else {
        osc_add_flash_ok_message($msg , 'admin');
    }

    $this->redirectTo(osc_admin_base_url(true) . '?
    page=languages');

break;
case('delete'):
    osc_csrf_check();
    if( is_array(Params::getParam('id')) ) {
        $default_lang = osc_language();
        foreach( Params::getParam('id') as $code ) {
            if( $default_lang != $code ) {
                if($this->localeManager->deleteLocale(

```



```

        b_enabled'] == 1 ? --('Disable (website)')
        : --('Enable (website)') . '</a>';
$options[] = '<a href="' . osc_admin_base_url(
true) . '?page=languages&action=' . (
    $l['b_enabled_bo'] == 1 ?
    disable_bo_selected : 'enable_bo_selected
') . '&id[]=' . $l['pk_c_code'] . '&
amp;' . osc_csrf_token_url() . '>' . ( $l
['b_enabled_bo'] == 1 ? --('Disable (oc-
admin)') : --('Enable (oc-admin)') ) . '</
a>';
$options[] = '<a onclick="return delete_dialog
(\'' . $l['pk_c_code'] . '\');" href="' .
osc_admin_base_url(true) . '?page=
languages&action=delete&id[]=' .
    $l['pk_c_code'] . '&' .
    osc_csrf_token_url() . '>' . --('Delete')
    . '</a>';

$auxOptions = '<ul>'.PHP_EOL;
foreach( $options as $actual ) {
    $auxOptions .= '<li>'. $actual . '</li>'.
        PHP_EOL;
}
$actions = '<div class="actions">'. $auxOptions
    . '</div>'.PHP_EOL;

$update = '';
// get languages to update from t_preference
if($bLanguagesToUpdate) {
    if(in_array($l['pk_c_code'],
        $aLanguagesToUpdate)){
        $update = '<a class="btn-market-
update btn-market-popup" href="#"'
            . htmlentities($l['pk_c_code']) .
            '>' . --("Update here") . '</a>';
    }
}

$row[] = $l['s_name'] . $update . $actions;
$row[] = $l['s_short_name'];
$row[] = $l['s_description'];
$row[] = ( $l['b_enabled'] ? --('Yes') : --('
No') );
$row[] = ( $l['b_enabled_bo'] ? --('Yes') : --
('No') );

$aData[] = $row;
}
// -----
$array['iTotalRecords'] = $displayRecords;
$array['iTotalDisplayRecords'] = count(
    $aLanguages);
$array['iDisplayLength'] = $limit;
$array['aData'] = $aData;

$page = (int)Params::getParam('iPage');
if(count($array['aData']) == 0 && $page!=1) {
    $total = (int)$array['iTotalDisplayRecords'];
    $maxPage = ceil( $total / (int)$array['
iDisplayLength'] );

    $url = osc_admin_base_url(true) . '?' . $_SERVER['
QUERY_STRING'];

    if($maxPage==0) {
        $url = preg_replace('/&iPage=(\d)+/', '&
iPage=1', $url);
        $this->redirectTo($url);
    }

    if($page > 1) {
        $url = preg_replace('/&iPage=(\d)+/', '&
iPage='.$maxPage, $url);
        $this->redirectTo($url);
    }
}

$this->exportVariableToView('aLanguages', $array)
;

$bulk_options = array(
    array('value' => '', 'data-dialog-content' =>
        '', 'label' => --('Bulk actions')),
    array('value' => 'enable_selected', 'data-
dialog-content' => sprintf(--('Are you
sure you want to %s the selected languages
?'), strtolower(--('Enable (Website)'))),
        'label' => --('Enable (Website)')),
    array('value' => 'disable_selected', 'data-
dialog-content' => sprintf(--('Are you
sure you want to %s the selected languages
?'), strtolower(--('Disable (Website)'))),
        'label' => --('Disable (Website)')),
    array('value' => 'enable_bo_selected', 'data-
dialog-content' => sprintf(--('Are you
sure you want to %s the selected languages
?'), strtolower(--('Enable (oc-admin)'))),
        'label' => --('Enable (oc-admin)')),

```



```

        array('value' => 'disable_bo_selected', 'data-
        dialog-content' => sprintf(--('Are you
        sure you want to %s the selected languages
        ?'), strtolower(--('Disable (oc-admin)'))
        ), 'label' => --('Disable (oc-admin)')),
        array('value' => 'delete', 'data-dialog-
        content' => sprintf(--('Are you sure you
        want to %s the selected languages?'),
        strtolower(--('Delete'))), 'label' => --('
        Delete'))
    );
    $bulk_options = osc_apply_filter("
        language_bulk_filter", $bulk_options);
    $this->exportVariableToView('bulk_options',
        $bulk_options);

    $this->doView('languages/index.php');

        break;
    }
}

//hopefully generic...
function doView($file)
{
    osc_run_hook("before_admin_html");
    osc_current_admin_theme_path($file);
    Session::newInstance()->.clearVariables();
    osc_run_hook("after_admin_html");
}

}

/* file end: ./oc-admin/languages.php */
?>

<?php if ( ! defined('ABS_PATH')) exit('ABS_PATH is not loaded. Direct access is not allowed
. ');

/*
 *      Oclass      software for creating and publishing online classified
 *                  advertising platforms
 *
 *                  Copyright (C) 2012 OSCLASS
 *
 *      This program is free software: you can redistribute it and/or
 *      modify it under the terms of the GNU Affero General Public License
 *      as published by the Free Software Foundation, either version 3 of
 *      the License, or (at your option) any later version.
 *
 *      This program is distributed in the hope that it will be useful, but
 *      WITHOUT ANY WARRANTY; without even the implied warranty of
 *      MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 *      GNU Affero General Public License for more details.
 *
 *      You should have received a copy of the GNU Affero General Public
 *      License along with this program. If not, see <http://www.gnu.org/licenses/>.
 */

class CAdminLogin extends AdminBaseModel
{
    function __construct()
    {
        parent::__construct();
    }

    //Business Layer...
    function doModel()
    {
        switch( $this->action ) {
            case('login_post'): //post execution for the login
                osc_csrf_check();
                $url_redirect = osc_get_http_referer();
                $page_redirect = '';
                if(preg_match('[\?&]page=([^&]+)|', $url_redirect
                    . '&', $match)) {
                    $page_redirect = $match[1];
                }
                if($page_redirect==' ' || $page_redirect=='login' ||
                    $url_redirect=='') {
                    $url_redirect = osc_admin_base_url();
                }

                if( Params::getParam('user') == '' ) {
                    osc_add_flash_error_message( _m('The username
                    field is empty'), 'admin');
                    $this->redirectTo( osc_admin_base_url(true)."?page
                    =login" );
                }

                if( Params::getParam('password', false, false) == '' )
                {
                    osc_add_flash_error_message( _m('The password
                    field is empty'), 'admin');
                    $this->redirectTo( osc_admin_base_url(true)."?page
                    =login" );
                }

                // fields are not empty
                $admin = Admin::newInstance()->findByUsername( Params
                    ::getParam('user') );

```

```

if( !$admin ) {
    osc_add_flash_error_message( sprintf(_m('Sorry,
        incorrect username. <a href="%s">Have you lost
        your password?</a>'), osc_admin_base_url(true)
        ). '?page=login&action=recover' ), 'admin
        ');
    $this->redirectTo( osc_admin_base_url(true).'?page
        =login' );
}

if( $admin['s_password'] !== sha1( Params::getParam('
    password', false, false) ) ) {
    osc_add_flash_error_message( sprintf(_m('Sorry,
        incorrect password. <a href="%s">Have you lost
        your password?</a>'), osc_admin_base_url(true)
        ). '?page=login&action=recover' ), 'admin
        ');
    $this->redirectTo( osc_admin_base_url(true).'?page
        =login' );
}

if( Params::getParam('remember') ) {
    // this include contains de osc_genRandomPassword
    function
    require_once osc_lib_path() . 'osclass/helpers/
        hSecurity.php';
    $secret = osc_genRandomPassword();

    Admin::newInstance()->update(
        array('s_secret' => $secret),
        array('pk.i.id' => $admin['pk.i.id'])
    );

    Cookie::newInstance()->set_expires(
        osc_time_cookie() );
    Cookie::newInstance()->push('oc_adminId', $admin['
        pk.i.id']);
    Cookie::newInstance()->push('oc_adminSecret',
        $secret);
    Cookie::newInstance()->push('oc_adminLocale',
        Params::getParam('locale'));
    Cookie::newInstance()->set();
}

// we are logged in... let's go!
Session::newInstance()->_set('adminId', $admin['
    pk.i.id']);
Session::newInstance()->_set('adminUserName', $admin['
    s_username']);
Session::newInstance()->_set('adminName', $admin['
    s_name']);
Session::newInstance()->_set('adminEmail', $admin['
    s_email']);
Session::newInstance()->_set('adminLocale', Params::
    getParam('locale'));

osc_run_hook('login_admin', $admin);

$this->redirectTo( $url_redirect );

break;
case('recover'): // form to recover the password (in this case we have
    the form in /gui/)
    $this->doView('gui/recover.php');

break;
case('recover_post'): if( defined('DEMO') ) {
    osc_add_flash_warning_message( _m("This action can
        't be done because it's a demo site"), 'admin
        ');
    $this->redirectTo( osc_admin_base_url() );
}
osc_csrf_check();

// post execution to recover the password
$admin = Admin::newInstance()->findByEmail( Params::
    getParam('email') );
if( $admin ) {
    if( (osc_recaptcha_private_key() != '') ) {
        if( !osc_check_recaptcha() ) {
            osc_add_flash_error_message( _m('The
                reCAPTCHA code is wrong'), 'admin');
            $this->redirectTo( osc_admin_base_url(true)
                ). '?page=login&action=recover' );
            return false; // BREAK THE PROCESS, THE
                RECAPTCHA IS WRONG
        }
    }

    require_once osc_lib_path() . 'osclass/helpers/
        hSecurity.php';
    $newPassword = osc_genRandomPassword(40);

    Admin::newInstance()->update(
        array('s_secret' => $newPassword),
        array('pk.i.id' => $admin['pk.i.id'])
    );
    $password_url =
        osc_forgot_admin_password_confirm_url($admin['
            pk.i.id'], $newPassword);
}

```

```

        osc_run_hook('hook_email_user_forgot_password',
                    $admin, $password_url);
    }

    osc_add_flash_ok_message( _m('A new password has been
    sent to your e-mail'), 'admin');
    $this->redirectTo(osc_admin_base_url(true) . '?page=
    login');

    break;
    case('forgot'):
        // form to recover the password (in this case we have
        the form in /gui/)
        $admin = Admin::newInstance()->findByIdSecret(Params::
        getParam('adminId'), Params::getParam('code'));
        if( !$admin ) {
            osc_add_flash_error_message( _m('Sorry, the link
            is not valid'), 'admin');
            $this->redirectTo( osc_admin_base_url() );
        }

        $this->doView( 'gui/forgot_password.php' );

    break;
    case('forgot_post'):

        osc_csrf_check();
        $admin = Admin::newInstance()->findByIdSecret(Params::
        getParam('adminId'), Params::getParam('code'));
        if( !$admin ) {
            osc_add_flash_error_message( _m('Sorry, the link
            is not valid'), 'admin');
            $this->redirectTo( osc_admin_base_url() );
        }

        if( Params::getParam('new_password', false, false) ==
        Params::getParam('new_password2', false, false) )
        {
            Admin::newInstance()->update(
                array('s_secret' => osc_genRandomPassword()
                    , 's_password' => sha1(Params::getParam('
                    new_password', false, false))
                    ), array('pk_i.id' => $admin['pk_i.id']
                )
            );
            osc_add_flash_ok_message( _m('The password has
            been changed'), 'admin');
            $this->redirectTo(osc_admin_base_url(true) . '?
            page=login');
        } else {
            osc_add_flash_error_message( _m(" Error, the
            passwords don't match"), 'admin');
            $this->redirectTo(
                osc_forgot_admin_password_confirm_url(Params::
                getParam('adminId'), Params::getParam('code'))
            );
        }

    break;
    default:

        osc_run_hook( 'init_admin' );
        Session::newInstance()->_setReferer(
            osc_get_http_referer());
        $this->doView( 'gui/login.php' );

    break;
}
}

//in this case, this function is prepared for the "recover your password" form
function doView($file)
{
    $login_admin_title = osc_apply_filter('login_admin_title', 'Osclass');
    $login_admin_url = osc_apply_filter('login_admin_url', 'http://osclass.org/');
    $login_admin_image = osc_apply_filter('login_admin_image', osc_admin_base_url() .
    'images/osclass-logo.gif');

    View::newInstance()->_exportVariableToView('login_admin_title', $login_admin_title
    );
    View::newInstance()->_exportVariableToView('login_admin_url', $login_admin_url);
    View::newInstance()->_exportVariableToView('login_admin_image', $login_admin_image
    );

    osc_run_hook("before_admin_html");
    require osc_admin_base_path() . $file;
    osc_run_hook("after_admin_html");
}
}

/* file end: ./oc-admin/login.php */

<?php if ( ! defined('ABSPATH')) exit('ABSPATH is not loaded. Direct access is not allowed
. ');

/*
*
*   Osclass      software for creating and publishing online classified
*                 advertising platforms
*
*
*               Copyright (C) 2012 OSCCLASS
*
*
*   This program is free software: you can redistribute it and/or
*   modify it under the terms of the GNU Affero General Public License
*   as published by the Free Software Foundation, either version 3 of
*   the License, or (at your option) any later version.
*/

```

```

*
*   This program is distributed in the hope that it will be useful, but
*   WITHOUT ANY WARRANTY; without even the implied warranty of
*   MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
*   GNU Affero General Public License for more details.
*
*   You should have received a copy of the GNU Affero General Public
*   License along with this program. If not, see <http://www.gnu.org/licenses/>.
*/

class CAdminMain extends AdminSecBaseModel
{
    function __construct ()
    {
        parent::__construct ();
    }

    //Business Layer...
    function doModel()
    {
        switch($this->action) {
            case('logout'): // unset only the required parameters in Session
                $this->logout();
                $this->redirectTo( osc_admin_base_url(true) );
                break;
            default: //default dashboard page (main page at oc-admin)
                $this->exportVariableToView( "numItemsPerCategory",
                    osc_get_non_empty_categories() );

                $this->exportVariableToView( "numUsers", User::
                    newInstance()->count() );
                $this->exportVariableToView( "numItems", Item::
                    newInstance()->count() );

                // stats
                $items = array();
                $stats_items = Stats::newInstance()->new_items_count(date(
                    'Y-m-d H:i:s', mktime(0, 0, 0, date("m"), date("d")
                    - 10, date("Y")) ), 'day');
                for($k = 10; $k >= 0; $k--) {
                    $items[date( 'Y-m-d', mktime(0, 0, 0, date("m"), date
                    ("d") - $k, date("Y")) )] = 0;
                }
                foreach($stats_items as $item) {
                    $items[$item['d.date']] = $item['num'];
                }
                $users = array();
                $stats_users = Stats::newInstance()->new_users_count(date(
                    'Y-m-d H:i:s', mktime(0, 0, 0, date("m"), date("d")
                    - 10, date("Y")) ), 'day');
                for($k = 10; $k >= 0; $k--) {
                    $users[date( 'Y-m-d', mktime(0, 0, 0, date("m"), date
                    ("d") - $k, date("Y")) )] = 0;
                }
                foreach($stats_users as $user) {
                    $users[$user['d.date']] = $user['num'];
                }

                if(function_exists('disk_free_space')) {
                    $freedisk = @disk_free_space(osc_uploads_path());
                    if($freedisk!==false && $freedisk < 52428800) {
                        //52428800 = 50*1024*1024
                        osc_add_flash_error_message(_m('You have very few
                        free space left, users will not be able to
                        upload pictures'), 'admin');
                    }
                }

                // show messages subscribed
                $status_subscribe = Params::getParam('subscribe-osclass');
                if( $status_subscribe != '' ) {
                    switch( $status_subscribe ) {
                        case -1:
                            osc_add_flash_error_message(_m('Entered an
                            invalid email'), 'admin');
                            break;
                        case 0:
                            osc_add_flash_warning_message(_m("You're
                            already subscribed"), 'admin');
                            break;
                        case 1:
                            osc_add_flash_ok_message(_m('Subscribed
                            correctly'), 'admin');
                            break;
                        default:
                            osc_add_flash_warning_message(_m("Error
                            subscribing"), 'admin');
                            break;
                    }
                }

                $this->exportVariableToView("item_stats", $items);
                $this->exportVariableToView("user_stats", $users);
                //calling the view...
                $this->doView('main/index.php');
            }
        }
    }

    //hopefully generic...
}

```

```

function doView($file)
{
    osc_run_hook(" before_admin.html");
    osc_current_admin_theme_path($file);
    Session::newInstance()->clearVariables();
    osc_run_hook(" after_admin.html");
}
}

/* file end: ./oc-admin/main.php */
?>

<?php if ( ! defined('ABS_PATH')) exit('ABS_PATH is not loaded. Direct access is not allowed
');

/**
 * Osclass software for creating and publishing online classified advertising
 * platforms
 *
 * Copyright (C) 2012 OSCLASS
 *
 * This program is free software: you can redistribute it and/or modify it under the terms
 * of the GNU Affero General Public License as published by the Free Software Foundation,
 * either version 3 of the License, or (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful, but WITHOUT ANY
 * WARRANTY;
 * without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
 * PURPOSE.
 * See the GNU Affero General Public License for more details.
 *
 * You should have received a copy of the GNU Affero General Public
 * License along with this program. If not, see <http://www.gnu.org/licenses/>.
 */

class CAdminMarket extends AdminSecBaseModel
{
    function __construct()
    {
        parent::__construct();
    }

    //Business Layer...
    function doModel()
    {
        parent::doModel();
        //specific things for this class
        switch ($this->action) {
            case('plugins'):
            case('themes'):
            case('languages'):
                $section = $this->action;
                $title = array(
                    'plugins' => --('Recommended plugins for You'),
                    'themes' => --('Recommended themes for You'),
                    'languages' => --('Languages for this version')
                );

                // page number
                $marketPage = Params::getParam("mPage");
                $url_actual = osc_admin_base_url(true) . '?page=market&action=' .
                    $section . '&mPage=' . $marketPage;
                if ($marketPage >= 1) $marketPage--;

                // api
                $url = osc_market_url($section) . "page/" . $marketPage . "/" . length
                    /9/';
                // default sort
                $sort_actual = '';
                $sort_download = $url_actual . '&sort=downloads&order=desc';
                $sort_updated = $url_actual . '&sort=updated&order=desc';

                // sorting options (default)
                $_order = 'desc';
                $order_download = $_order;
                $order_updated = $_order;

                $sort = Params::getParam("sort");
                $order = Params::getParam("order");

                if ($sort == '') {
                    $sort = 'updated';
                }
                if ($order == '') {
                    $order = $_order;
                }

                $aux = ($order == 'desc') ? 'asc' : 'desc';

                switch ($sort) {
                    case 'downloads':
                        $sort_actual = '&sort=downloads&order=';
                        $sort_download = $url_actual . $sort_actual . $aux;
                        $sort_updated = $url_actual . $sort_actual . $aux;
                        $order_download = $order;
                        // market api call
                        $url .= 'order/downloads/' . $order;
                        break;
                    case 'updated':

```

```

        $sort_actual    = '&sort=updated&order=';
        $sort_updated  = $url_actual.$sort_actual.$aux;
        $sort_actual   .= $order;
        $order_updated = $order;
        // market api call
        $url .= 'order/updated/'. $order;
    break;
default:
    break;
}

// pageSize or length attribute is hardcoded
$out = osc_file_get_contents($url);
$array = json_decode($out, true);

$output_pagination = '';
if( is_numeric($array['total']) && $array['total']>0 ) {
    $totalPages = ceil( $array['total'] / $array['sizePage'] );
    $pageActual = $array['page'];
    $params = array(
        'total' => $totalPages,
        'selected' => $pageActual,
        'url' => osc_admin_base_url(true).'?page=market'.'&action
            ='. $section.'&mPage={PAGE}'. $sort_actual,
        'sides' => 5
    );
    // set pagination
    $pagination = new Pagination($params);
    $output_pagination = $pagination->doPagination();
} else {
    $array['total'] = 0;
}

// export variable to view
$this->_exportVariableToView(" sort"           , $sort);
$this->_exportVariableToView(" title"          , $title);
$this->_exportVariableToView(" section"        , $section);
$this->_exportVariableToView(" array"          , $array);

$this->_exportVariableToView(" sort_download"   , $sort_download);
$this->_exportVariableToView(" sort_updated"    , $sort_updated);

$this->_exportVariableToView(" order_download"  , $order_download);
$this->_exportVariableToView(" order_updated"   , $order_updated);

$this->_exportVariableToView(' pagination' , $output_pagination);

$this->doView(" market/section.php");
break;
default:
    $aPlugins = array();
    $aThemes = array();
    $aLanguages = array();

    $out_plugin = osc_file_get_contents(osc_market_featured_url('plugins',
        6) );
    $array_plugins = json_decode($out_plugin, true);
    if(isset($array_plugins)) {
        $aPlugins = $array_plugins['plugins'];
    }

    $out_themes = osc_file_get_contents(osc_market_featured_url('themes',
        6) );
    $array_themes = json_decode($out_themes, true);
    if(isset($array_themes)) {
        $aThemes = $array_themes['themes'];
    }

    $out_languages = osc_file_get_contents(osc_market_featured_url('
        languages', 6) );
    $array_languages = json_decode($out_languages, true);
    if(isset($array_languages)) {
        $aLanguages = $array_languages['languages'];
    }

    $count = json_decode( osc_file_get_contents(osc_market_count_url(), true)
        );
    if( !isset($count['pluginsTotal']) ) {
        $count['pluginsTotal'] = 0;
    }
    if( !isset($count['themesTotal']) ) {
        $count['themesTotal'] = 0;
    }
    if( !isset($count['languagesTotal']) ) {
        $count['languagesTotal'] = 0;
    }

    $this->_exportVariableToView(" count"           , $count);
    $this->_exportVariableToView(" aPlugins"        , $aPlugins);
    $this->_exportVariableToView(" aThemes"         , $aThemes);
    $this->_exportVariableToView(" aLanguages"      , $aLanguages);

    $this->doView(" market/index.php");
break;
}
}

//hopefully generic...

```

```

function doView($file)
{
    osc_run_hook(" before_admin.html");
    osc_current_admin_theme_path($file);
    Session::newInstance()->clearVariables();
    osc_run_hook(" after_admin.html");
}
}

/* file end: ./oc-admin/market.php */
?>

<?php if ( ! defined('ABS_PATH')) exit('ABS_PATH is not loaded. Direct access is not allowed
');

/*
 *      Oclass      software for creating and publishing online classified
 *                  advertising platforms
 *
 *                  Copyright (C) 2012 OSCLASS
 *
 *      This program is free software: you can redistribute it and/or
 *      modify it under the terms of the GNU Affero General Public License
 *      as published by the Free Software Foundation, either version 3 of
 *      the License, or (at your option) any later version.
 *
 *      This program is distributed in the hope that it will be useful, but
 *      WITHOUT ANY WARRANTY; without even the implied warranty of
 *      MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 *      GNU Affero General Public License for more details.
 *
 *      You should have received a copy of the GNU Affero General Public
 *      License along with this program. If not, see <http://www.gnu.org/licenses/>.
 */

class CAdminMedia extends AdminSecBaseModel
{
    private $resourcesManager;

    function __construct()
    {
        parent::__construct();

        //specific things for this class
        $this->resourcesManager = ItemResource::newInstance();
    }

    //Business Layer...
    function doModel()
    {
        parent::doModel();

        //specific things for this class
        switch($this->action) {
            case('bulk_actions'):
                osc_csrf_check();
                switch ( Params::getParam('bulk_actions') ) {
                    case 'delete':
                        $sids = Params::getParam("id");
                        if( is_array($sids) ) {
                            foreach($sids as $sid) {
                                osc_deleteResource( $sid , true);
                            }
                            $log_ids = substr(implode(",",$sids),0,
                                250);
                            Log::newInstance()->insertLog('media', '
                                delete bulk', $log_ids, $log_ids, '
                                admin', osc_logged_admin_id());
                            $this->resourcesManager->
                                deleteResourcesIds($sids);
                        }
                        osc_add_flash_ok_message( _m('Resource deleted
                            '), 'admin');
                    break;
                    default:
                        if(Params::getParam("bulk_actions")!="") {
                            osc_run_hook(" media_bulk_", Params::
                                getParam("bulk_actions"), Params::
                                getParam('id'));
                        }
                    break;
                }
                $this->redirectTo( osc_admin_base_url(true) . "?page=
                    media' );
            break;
            case('delete'):
                osc_csrf_check();
                $sids = Params::getParam('id');
                if( is_array($sids) ) {
                    foreach($sids as $sid) {
                        osc_deleteResource( $sid , true);
                    }
                    $log_ids = substr(implode(",",$sids),0, 250);
                    Log::newInstance()->insertLog('media', 'delete',
                        $log_ids, $log_ids, 'admin',
                        osc_logged_admin_id());
                    $this->resourcesManager->deleteResourcesIds($sids);
                }
            }
        }
    }
}

```





```

    }
}
/* file end: ./oc-admin/media.php */
?>
<?php if ( ! defined('ABS_PATH')) exit('ABS_PATH is not loaded. Direct access is not allowed
');

/*
 *   Oclass      software for creating and publishing online classified
 *               advertising platforms
 *
 *               Copyright (C) 2012 OSCLASS
 *
 *   This program is free software: you can redistribute it and/or
 *   modify it under the terms of the GNU Affero General Public License
 *   as published by the Free Software Foundation, either version 3 of
 *   the License, or (at your option) any later version.
 *
 *   This program is distributed in the hope that it will be useful, but
 *   WITHOUT ANY WARRANTY; without even the implied warranty of
 *   MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 *   GNU Affero General Public License for more details.
 *
 *   You should have received a copy of the GNU Affero General Public
 *   License along with this program. If not, see <http://www.gnu.org/licenses/>.
 */

class CAdminPages extends AdminSecBaseModel
{
    //specific for this class
    private $pageManager;

    function __construct()
    {
        parent::__construct();

        //specific things for this class
        $this->pageManager = Page::newInstance();
    }

    //Business Layer...
    function doModel()
    {
        parent::doModel();

        //specific things for this class
        switch($this->action) {
            case 'edit':
                if (Params::getParam("id")=='') {
                    $this->redirectTo(osc_admin_base_url(true)."?page=pages");
                }

                $form = count(Session::newInstance()->getForm());
                $keepForm = count(Session::newInstance()->getKeepForm());
                if($form == 0 || $form == $keepForm) {
                    Session::newInstance()->dropKeepForm();
                }

                $templates = osc_apply_filter('page_templates', WebThemes::newInstance()->
                    getAvailableTemplates());
                $this->exportVariableToView('templates', $templates);
                $this->exportVariableToView("page", $this->pageManager->findByPrimaryKey(
                    Params::getParam("id")));
                $this->doView("pages/frm.php");
                break;
            case 'edit_post':
                osc_csrf_check();
                $id = Params::getParam("id");
                $b_link = (Params::getParam("b_link") != '') ? 1 : 0;
                $s_internal_name = Params::getParam("s_internal_name");
                $s_internal_name = osc_sanitizeString($s_internal_name);

                $meta = Params::getParam('meta');
                $this->pageManager->updateMeta($id, json_encode($meta));

                $aFieldsDescription = array();
                $postParams = Params::getParamsAsArray('', false);
                $not_empty = false;
                foreach ($postParams as $k => $v) {
                    if(preg_match('|(.+?)#(.+)|', $k, $m)) {
                        if ($m[2]=='s_title' && $v!='') { $not_empty = true; };
                        $aFieldsDescription[$m[1]][$m[2]] = $v;
                    }
                }
                Session::newInstance()->setForm('aFieldsDescription', $aFieldsDescription);
                ;

                if( $s_internal_name == '' ) {
                    osc_add_flash_error_message(_m('You have to set an internal name'), '
                    admin');
                    $this->redirectTo(osc_admin_base_url(true)."?page=pages&action=edit&id
                    =" . $id);
                }

                if( !WebThemes::newInstance()->isValidPage($s_internal_name) ) {
                    osc_add_flash_error_message(_m('You have to set a different internal
                    name'), 'admin');
                }
            }
        }
    }
}

```

```

        $this->redirectTo(osc_admin_base_url(true)."?page=pages&action=edit&id=" . $id);
    }
    Session::newInstance()->_setForm('s_internal_name', $s_internal_name);
    if($not_empty) {
        foreach($aFieldsDescription as $k => $data) {
            $this->pageManager->updateDescription($id, $k, $data['s_title'],
                $data['s_text']);
        }
        if(!$this->pageManager->internalNameExists($id, $s_internal_name)) {
            if(!$this->pageManager->isIndelible($id)) {
                $this->pageManager->updateInternalName($id, $s_internal_name);
                $this->pageManager->updateLink($id, $b_link);
            }
            osc_run_hook('edit_page', $id);
            Session::newInstance()->_clearVariables();
            osc_add_flash_ok_message(_m('The page has been updated'), 'admin');
            $this->redirectTo(osc_admin_base_url(true)."?page=pages");
        }
        osc_add_flash_error_message(_m("You can't repeat internal name"), 'admin');
    }
    } else {
        osc_add_flash_error_message(_m("The page couldn't be updated, at least one title should not be empty"), 'admin');
    }
    $this->redirectTo(osc_admin_base_url(true)."?page=pages&action=edit&id=" . $id);
    break;
case 'add':
    $form = count(Session::newInstance()->_getForm());
    $keepForm = count(Session::newInstance()->_getKeepForm());
    if($form == 0 || $form == $keepForm) {
        Session::newInstance()->_dropKeepForm();
    }

    $templates = osc_apply_filter('page_templates', WebThemes::newInstance()->getAvailableTemplates());
    $this->_exportVariableToView('templates', $templates);
    $this->_exportVariableToView("page", array());
    $this->doView("pages/frm.php");
    break;
case 'add_post':
    osc_csrf_check();
    $s_internal_name = Params::getParam("s_internal_name");
    $b_link = (Params::getParam("b_link") != '') ? 1 : 0;
    $s_internal_name = osc_sanitizeString($s_internal_name);

    $meta = Params::getParam('meta');

    $aFieldsDescription = array();
    $postParams = Params::getParamsAsArray('', false);
    $not_empty = false;
    foreach($postParams as $k => $v) {
        if(preg_match('!(.+?)#(.)|', $k, $m)) {
            if($m[2]='s_title' && $v!='') {
                $not_empty = true;
            }
            $aFieldsDescription[$m[1]][$m[2]] = $v;
        }
    }
    Session::newInstance()->_setForm('aFieldsDescription', $aFieldsDescription);

    if($s_internal_name == '') {
        osc_add_flash_error_message(_m('You have to set an internal name'), 'admin');
        $this->redirectTo(osc_admin_base_url(true)."?page=pages&action=add");
    }

    if(!WebThemes::newInstance()->isValidPage($s_internal_name)) {
        osc_add_flash_error_message(_m('You have to set a different internal name'), 'admin');
        $this->redirectTo(osc_admin_base_url(true)."?page=pages&action=add");
    }
    $aFields = array('s_internal_name' => $s_internal_name, 'b_indelible' => '0', 's_meta' => json_encode($meta), 'b_link' => $b_link);
    Session::newInstance()->_setForm('s_internal_name', $s_internal_name);

    $page = $this->pageManager->findByInternalName($s_internal_name);
    if(!isset($page['pk_i_id'])) {
        if($not_empty) {
            $result = $this->pageManager->insert($aFields, $aFieldsDescription);
            Session::newInstance()->_clearVariables();
            osc_add_flash_ok_message(_m('The page has been added'), 'admin');
            $this->redirectTo(osc_admin_base_url(true)."?page=pages");
        }
        } else {
            osc_add_flash_error_message(_m("The page couldn't be added, at least one title should not be empty"), 'admin');
        }
    }
    } else {
        osc_add_flash_error_message(_m("Oops! That internal name is already in use. We can't make the changes"), 'admin');
    }
    $this->redirectTo(osc_admin_base_url(true)."?page=pages&action=add");
    break;

```

```

case 'delete':
    osc_csrf_check();
    $id = Params::getParam("id");
    $page_deleted_correcty = 0;
    $page_deleted_error = 0;
    $page_indelible = 0;

    if(!is_array($id)) {
        $id = array($id);
    }

    foreach($id as $_id) {
        $result = (int) $this->pageManager->deleteByPrimaryKey($_id);
        switch ($result) {
            case -1:
                $page_indelible++;
                break;
            case 0:
                $page_deleted_error++;
                break;
            case 1:
                $page_deleted_correcty++;
        }
    }

    if($page_indelible > 0) {
        if($page_indelible == 1) {
            osc_add_flash_error_message(_m("One page can't be deleted because
            it is indelible"), 'admin');
        } else {
            osc_add_flash_error_message(sprintf(_m("%s pages couldn't be
            deleted because they are indelible"), $page_indelible), 'admin
            ');
        }
    }

    if($page_deleted_error > 0) {
        if($page_deleted_error == 1) {
            osc_add_flash_error_message(_m("One page couldn't be deleted"), '
            admin');
        } else {
            osc_add_flash_error_message(sprintf(_m("%s pages couldn't be
            deleted"), $page_deleted_error), 'admin');
        }
    }

    if($page_deleted_correcty > 0) {
        if($page_deleted_correcty == 1) {
            osc_add_flash_ok_message(_m('One page has been deleted correctly')
            , 'admin');
        } else {
            osc_add_flash_ok_message(sprintf(_m('%s pages have been deleted
            correctly'), $page_deleted_correcty), 'admin');
        }
    }

    $this->redirectTo(osc_admin_base_url(true) . "?page=pages");
    break;
default:

    if(Params::getParam("action")!="") {
        osc_run_hook("page_bulk_".Params::getParam("action"), Params::getParam
        ('id'));
    }

    require_once osc_lib_path()."osclass/classes/datatables/PagesDataTable.php
    ";

    // set default iDisplayLength
    if( Params::getParam('iDisplayLength') != '' ) {
        Cookie::newInstance()->push('listing_iDisplayLength', Params::getParam
        ('iDisplayLength'));
        Cookie::newInstance()->set();
    } else {
        // set a default value if it's set in the cookie
        if( Cookie::newInstance()->get_value('listing_iDisplayLength') != '' )
        {
            Params::setParam('iDisplayLength', Cookie::newInstance()->
            get_value('listing_iDisplayLength'));
        } else {
            Params::setParam('iDisplayLength', 10 );
        }
    }

    $this->_exportVariableToView('iDisplayLength', Params::getParam('
    iDisplayLength'));

    // Table header order by related
    if( Params::getParam('sort') == '' ) {
        Params::setParam('sort', 'date');
    }
    if( Params::getParam('direction') == '' ) {
        Params::setParam('direction', 'desc');
    }
    }

    $page = (int)Params::getParam('iPage');
    if($page==0) { $page = 1; };
    Params::setParam('iPage', $page);

    $params = Params::getParamsAsArray("get");

    $pagesDataTable = new PagesDataTable();
    $pagesDataTable->table($params);

```

```

        $aData = $pagesDataTable->getData();

        if(count($aData['aRows']) == 0 && $page!=1) {
            $total = (int)$aData['iTotalDisplayRecords'];
            $maxPage = ceil( $total / (int)$aData['iDisplayLength'] );

            $url = osc_admin_base_url(true) . '?' . $_SERVER['QUERY_STRING'];

            if($maxPage==0) {
                $url = preg_replace('/&iPage=(\d)+/', '&iPage=1', $url);
                $this->redirectTo($url);
            }

            if($page > 1) {
                $url = preg_replace('/&iPage=(\d)+/', '&iPage='.$maxPage, $url);
                $this->redirectTo($url);
            }
        }

        $this->exportVariableToView('aData', $aData);
        $this->exportVariableToView('aRawRows', $pagesDataTable->rawRows());

        $bulk_options = array(
            array('value' => '', 'data-dialog-content' => '', 'label' => __('Bulk
                actions')),
            array('value' => 'delete', 'data-dialog-content' => sprintf(__('Are
                you sure you want to %s the selected pages?'), strtolower(__('
                Delete'))), 'label' => __('Delete'))
        );
        $bulk_options = osc_apply_filter("page_bulk_filter", $bulk_options);
        $this->exportVariableToView('bulk_options', $bulk_options);

        $this->doView("pages/index.php");
        break;
    }
}

//hopefully generic...
function doView($file)
{
    osc_run_hook("before_admin_html");
    osc_current_admin_theme_path($file);
    Session::newInstance()->clearVariables();
    osc_run_hook("after_admin_html");
}

/* file end: ./oc-admin/pages.php */
?>

<?php if ( ! defined('ABS_PATH')) exit('ABS_PATH is not loaded. Direct access is not allowed
. ');

/*
 *      Oclass      software for creating and publishing online classified
 *                  advertising platforms
 *
 *      Copyright (C) 2012 OCLASS
 *
 *      This program is free software: you can redistribute it and/or
 *      modify it under the terms of the GNU Affero General Public License
 *      as published by the Free Software Foundation, either version 3 of
 *      the License, or (at your option) any later version.
 *
 *      This program is distributed in the hope that it will be useful, but
 *      WITHOUT ANY WARRANTY; without even the implied warranty of
 *      MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 *      GNU Affero General Public License for more details.
 *
 *      You should have received a copy of the GNU Affero General Public
 *      License along with this program. If not, see <http://www.gnu.org/licenses/>.
 */

class CAdminPlugins extends AdminSecBaseModel
{
    function __construct()
    {
        parent::__construct();
        //specific things for this class
    }

    // Business layer ...
    function doModel()
    {
        parent::doModel();

        //specific things for this class
        switch ($this->action)
        {
            case 'add':
                $this->doView("plugins/add.php");
                break;
            case 'add_post':
                if( defined('DEMO') ) {
                    osc_add_flash_warning_message( _m("This action can't be done because
                    it's a demo site"), 'admin');
                    $this->redirectTo(osc_admin_base_url(true) . '?page=plugins');
                }
        }
    }
}

```

```

osc_csrf_check();

$package = Params::getFiles("package");
if (isset($package['size']) && $package['size']!=0) {
    $path = osc_plugins_path();
    (int) $status = osc_unzip_file($package['tmp-name'], $path);
} else {
    $status = 3;
}
switch ($status) {
    case(0): $msg = _m('The plugin folder is not writable');
            osc_add_flash_error_message($msg, 'admin');
            break;
    case(1): $msg = _m('The plugin has been uploaded correctly');
            osc_add_flash_ok_message($msg, 'admin');
            break;
    case(2): $msg = _m('The zip file is not valid');
            osc_add_flash_error_message($msg, 'admin');
            break;
    case(3): $msg = _m('No file was uploaded');
            osc_add_flash_error_message($msg, 'admin');
            $this->redirectTo(osc_admin_base_url(true)."?page=plugins&
                action=add");
            break;
    case(-1):
    default: $msg = _m('There was a problem adding the plugin');
            osc_add_flash_error_message($msg, 'admin');
            break;
}

$this->redirectTo(osc_admin_base_url(true)."?page=plugins");
break;
case 'install':
    if ( defined('DEMO') ) {
        osc_add_flash_warning_message( _m("This action can't be done because
            it's a demo site"), 'admin');
        $this->redirectTo(osc_admin_base_url(true) . '?page=plugins');
    }
    osc_csrf_check();
    $pn = Params::getParam('plugin');

    // set header just in case it's triggered some fatal error
    header("Location: " . osc_admin_base_url(true) . "?page=plugins&error=" .
        $pn, true, '302');

    $installed = Plugins::install($pn);
    if ( is_array($installed) ) {
        switch($installed['error_code']) {
            case('error_output'):
                osc_add_flash_error_message( sprintf( _m('The plugin generated
                    %d characters of <strong>unexpected output</strong>
                    during the installation. Output: "%s" '), strlen($installed
                        ['output']), $installed['output'] ), 'admin');
                break;
            case('error_installed'):
                osc_add_flash_error_message( _m('Plugin is already installed')
                    , 'admin');
                break;
            case('error_file'):
                osc_add_flash_error_message( _m("Plugin couldn't be installed
                    because their files are missing"), 'admin');
                break;
            case('custom_error'):
                osc_add_flash_error_message( sprintf(_m("Plugin couldn't be
                    installed because of: %s"), $installed['msg']), 'admin');
                break;
            default:
                osc_add_flash_error_message( _m("Plugin couldn't be installed
                    "), 'admin');
                break;
        }
    }
    } else {
        osc_add_flash_ok_message( _m('Plugin installed'), 'admin');
    }
}

$this->redirectTo(osc_admin_base_url(true) . '?page=plugins');
break;
case 'uninstall':
    if ( defined('DEMO') ) {
        osc_add_flash_warning_message( _m("This action can't be done because
            it's a demo site"), 'admin');
        $this->redirectTo(osc_admin_base_url(true) . '?page=plugins');
    }
    osc_csrf_check();

    if ( Plugins::uninstall(Params::getParam("plugin")) ) {
        osc_add_flash_ok_message( _m('Plugin uninstalled'), 'admin');
    } else {
        osc_add_flash_error_message( _m("Plugin couldn't be uninstalled"), '
            admin');
    }
}

$this->redirectTo(osc_admin_base_url(true) . '?page=plugins');
break;
case 'enable':
    if ( defined('DEMO') ) {
        osc_add_flash_warning_message( _m("This action can't be done because
            it's a demo site"), 'admin');
        $this->redirectTo(osc_admin_base_url(true) . '?page=plugins');
    }

```

```

    }
    osc_csrf_check();

    if( Plugins::activate(Params::getParam('plugin')) ) {
        osc_add_flash_ok_message( _m('Plugin enabled'), 'admin');
    } else {
        osc_add_flash_error_message( _m('Plugin is already enabled'), 'admin')
        ;
    }
}

$this->redirectTo(osc_admin_base_url(true) . '?page=plugins');
break;
case 'disable':
    if( defined('DEMO') ) {
        osc_add_flash_warning_message( _m("This action can't be done because
            it's a demo site"), 'admin');
        $this->redirectTo(osc_admin_base_url(true) . '?page=plugins');
    }
    osc_csrf_check();

    if( Plugins::deactivate(Params::getParam('plugin')) ) {
        osc_add_flash_ok_message( _m('Plugin disabled'), 'admin');
    } else {
        osc_add_flash_error_message( _m('Plugin is already disabled'), 'admin
            ');
    }

    $this->redirectTo(osc_admin_base_url(true) . '?page=plugins');
    break;
case 'admin':
    $plugin = Params::getParam("plugin");
    if($plugin != "") {
        osc_run_hook($plugin.'_configure');
    }
    break;
case 'admin_post':
    osc_run_hook('admin_post');
    break;
case 'renderplugin':

    if(Params::existParam('route')) {
        $routes = Rewrite::newInstance()->getRoutes();
        $rid = Params::getParam('route');
        $file = '../';
        if(isset($routes[$rid]) && isset($routes[$rid]['file'])) {
            $file = $routes[$rid]['file'];
        }
    } else {
        // DEPRECATED: Disclosed path in URL is deprecated, use routes instead
        // This will be REMOVED in 3.4
        $file = Params::getParam('file');
        // We pass the GET variables (in case we have some)
        if(preg_match('|(.+?)\?(.*)|', $file, $match)) {
            $file = $match[1];
            if(preg_match_all('|&([=+])=([&]*)|', urldecode('&'. $match
                [2]. '&'), $get_vars)) {
                for($var_k=0;$var_k<count($get_vars[1]);$var_k++) {
                    Params::setParam($get_vars[1][$var_k], $get_vars[2][$var_k
                        ]);
                }
            }
        }
    } else {
        $file = $_REQUEST['file'];
    }
};

if(strpos($file, '../')===false && $file!="") {
    $this->exportVariableToView("file", osc_plugins_path() . $file);
    $this->doView("plugins/view.php");
}
break;
case 'configure':
    $plugin = Params::getParam("plugin");
    if($plugin!="") {
        $plugin_data = Plugins::getInfo($plugin);
        $this->exportVariableToView("categories", Category::newInstance()->
            toTreeAll());
        $this->exportVariableToView("selected", PluginCategory::newInstance()
            ->listSelected($plugin_data['short_name']));
        $this->exportVariableToView("plugin_data", $plugin_data);
        $this->doView("plugins/configuration.php");
    } else {
        $this->redirectTo(osc_admin_base_url(true)."?page=plugins");
    }
}
break;
case 'configure_post':
    osc_csrf_check();
    $plugin_short_name = Params::getParam("plugin_short_name");
    $categories = Params::getParam("categories");
    if( $plugin_short_name != "" ) {
        Plugins::cleanCategoryFromPlugin($plugin_short_name);
        if(isset($categories)) {
            Plugins::addToCategoryPlugin($categories, $plugin_short_name);
        }
        osc_run_hook('plugin_categories_'.$Params::getParam('plugin'),
            $categories);
        osc_add_flash_ok_message( _m('Configuration was saved'), 'admin');
        $this->redirectTo(osc_admin_base_url(true)."?page=plugins");
    }
}
}

```

```

        osc_add_flash_error_message( _m('No plugin selected'), 'admin');
        $this->doView('plugins/index.php');
    break;
case 'error_plugin':
    // force php errors and simulate plugin installation to show the errors in
    the iframe
    if( !OSC_DEBUG ) {
        error_reporting( E_ALL | E_STRICT );
    }
    @ini_set( 'display_errors', 1 );

    include( osc_plugins_path() . Params::getParam('plugin') );
    Plugins::install(Params::getParam('plugin'));
    exit;
break;
default:

    if(Params::getParam('checkUpdated') != '') {
        osc_admin_toolbar_update_plugins(true);
    }

    if( Params::getParam('iDisplayLength') == '' ) {
        Params::setParam('iDisplayLength', 10 );
    }
    // ?
    $this->_exportVariableToView('iDisplayLength', Params::getParam('iDisplayLength'));

    $p_iPage = 1;
    if( is_numeric(Params::getParam('iPage')) && Params::getParam('iPage') >= 1 ) {
        $p_iPage = Params::getParam('iPage');
    }
    Params::setParam('iPage', $p_iPage);
    $aPlugin = Plugins::listAll();
    $active_plugins = osc_get_plugins();

    // pagination
    $start = ($p_iPage-1) * Params::getParam('iDisplayLength');
    $limit = Params::getParam('iDisplayLength');
    $count = count( $aPlugin );

    $displayRecords = $limit;
    if( ($start+$limit) > $count ) {
        $displayRecords = ($start+$limit) - $count;
    }
    // -----

    $aData = array();
    $aInfo = array();
    $smax = ($start+$limit);
    if($smax > $count) $smax = $count;
    $aPluginsToUpdate = json_decode( osc_get_preference('plugins_to_update') );
    ;
    $bPluginsToUpdate = is_array($aPluginsToUpdate)?true:false;
    for($i = $start; $i < $smax; $i++) {
        $plugin = $aPlugin[$i];
        $row = array();
        $pInfo = osc_plugin_get_info($plugin);

        // prepare row 1
        $installed = 0;
        if( osc_plugin_is_installed($plugin) ) {
            $installed = 1;
        }
        $enabled = 0;
        if( osc_plugin_is_enabled($plugin) ) {
            $enabled = 1;
        }
        // prepare row 2
        $sUpdate = '';
        // get plugins to update from t_preference
        if($bPluginsToUpdate) {
            if(in_array(@$pInfo['plugin_update_uri'], $aPluginsToUpdate)) {
                $sUpdate = '<a class="market_update market_popup" href="#" .
                htmlentities($pInfo['plugin_update_uri']) . "> . --("
                There's a new update available") . '</a>';
            }
        }
        // prepare row 4
        $sConfigure = '';
        if( isset($active_plugins[$plugin . '_configure']) ) {
            $sConfigure = '<a href="" . osc_admin_base_url(true) . "?page=
            plugins&action=admin&plugin=' . $pInfo['filename'] .
            "&" . osc_csrf_token_url() . "> . --('Configure') . '</a
            >';
        }
        // prepare row 5
        $sEnable = '';
        if( $installed ) {
            if( $enabled ) {
                $sEnable = '<a href="" . osc_admin_base_url(true) . "?page=
                plugins&action=disable&plugin=' . $pInfo['filename'] .
                "&" . osc_csrf_token_url() . "> . --('Disable')
                . '</a>';
            } else {
                $sEnable = '<a href="" . osc_admin_base_url(true) . "?page=
                plugins&action=enable&plugin=' . $pInfo['filename'] .
                "&" . osc_csrf_token_url() . "> . --('Enable')
                . '</a>';
            }
        }
    }

```

```

        . '</a>';
    }
}
// prepare row 6
$sInstall = '';
if( $installed ) {
    $sInstall = '<a onclick="javascript:return uninstall_dialog(\'' .
        $pInfo['filename'] . '\', \'' . $pInfo['plugin_name'] . '\');"
        href="' . osc_admin_base_url(true) . '?page=plugins&
        action=uninstall&plugin=' . $pInfo['filename'] . "&" .
        osc_csrf_token_url() . '">' . --('Uninstall') . '</a>';
    } else {
        $sInstall = '<a href="' . osc_admin_base_url(true) . '?page=
        plugins&action=install&plugin=' . $pInfo['filename'] .
        "&" . osc_csrf_token_url() . '">' . --('Install') . '</a
        >';
    }
}

$sHelp = '';
if($pInfo['support_uri']!= '') {
    $sHelp = '<span class="plugin-support-icon plugin-tooltip" ><a
    href="' . osc_sanitize_url($pInfo['support_uri']) . '" ><img src
    ="' . osc_current_admin_theme_url('images/question.png') . '" alt
    ="' . osc_esc_html(--('Problems with this plugin? Ask for
    support.')) . '" ></a></span>';
}

$row[] = '<input type="hidden" name="installed" value="' . $installed
. '" enabled="' . $enabled . '" />' . $pInfo['plugin_name'] .
    $sHelp . '<div>' . $sUpdate . '</div>';
$row[] = $pInfo['description'];
$row[] = ($sUpdate!= '') ? $sUpdate : '&nbsp;';
$row[] = ($sConfigure!= '') ? $sConfigure : '&nbsp;';
$row[] = ($sEnable!= '') ? $sEnable : '&nbsp;';
$row[] = ($sInstall!= '') ? $sInstall : '&nbsp;';
$aData[] = $row;
if(@$pInfo['plugin_update_uri'] != '') {
    $aInfo[@$pInfo['plugin_update_uri']] = $pInfo;
} else {
    $aInfo[$i] = $pInfo;
}
}

$array['iTotalRecords'] = $displayRecords;
$array['iTotalDisplayRecords'] = count($aPlugins);
$array['iDisplayLength'] = $limit;
$array['aData'] = $aData;
$array['aInfo'] = $aInfo;

// -----
$page = (int)Params::getParam('iPage');
if(count($array['aData']) == 0 && $page!=1) {
    $total = (int)$array['iTotalDisplayRecords'];
    $maxPage = ceil( $total / (int)$array['iDisplayLength'] );

    $url = osc_admin_base_url(true) . '?' . $_SERVER['QUERY_STRING'];

    if($maxPage==0) {
        $url = preg_replace('/&iPage=(\d)+/', '&iPage=1', $url);
        $this->redirectTo($url);
    }

    if($page > 1) {
        $url = preg_replace('/&iPage=(\d)+/', '&iPage=' . $maxPage, $url);
        $this->redirectTo($url);
    }
}

$this->exportVariableToView('aPlugins', $array);
$this->doView("plugins/index.php");
break;
}
}

//hopefully generic...
function doView($file)
{
    osc_run_hook("before_admin_html");
    osc_current_admin_theme_path($file);
    Session::newInstance()->clearVariables();
    osc_run_hook("after_admin_html");
}
}

/* file end: ./oc-admin/plugins.php */
?>

<?php if ( ! defined('ABSPATH')) exit('ABSPATH is not loaded. Direct access is not allowed
. ');

/*
 *      Oclass      software for creating and publishing online classified
 *                  advertising platforms
 *
 *      Copyright (C) 2012 OSCLASS
 *
 *      This program is free software: you can redistribute it and/or
 *      modify it under the terms of the GNU Affero General Public License

```



```

*      as published by the Free Software Foundation, either version 3 of
*      the License, or (at your option) any later version.
*
*      This program is distributed in the hope that it will be useful, but
*      WITHOUT ANY WARRANTY; without even the implied warranty of
*      MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
*      GNU Affero General Public License for more details.
*
*      You should have received a copy of the GNU Affero General Public
*      License along with this program. If not, see <http://www.gnu.org/licenses/>.
*/

class CAdminSettings extends AdminSecBaseModel
{
    //Business Layer...
    function doModel()
    {
        switch($this->action) {
            case('advanced'):
            case('advanced_post'):
                require_once(osc_admin_base_path() . 'controller/settings/advanced.php');
                $do = new CAdminSettingsAdvanced();
                break;
            case('comments'):
            case('comments_post'):
                require_once(osc_admin_base_path() . 'controller/settings/comments.php');
                $do = new CAdminSettingsComments();
                break;
            case('locations'):
                require_once(osc_admin_base_path() . 'controller/settings/locations.php');
                $do = new CAdminSettingsLocations();
                break;
            case('permalinks'):
            case('permalinks_post'):
                require_once(osc_admin_base_path() . 'controller/settings/permalinks.php');
                $do = new CAdminSettingsPermalinks();
                break;
            case('spamNbots'):
            case('akismet_post'):
            case('recaptcha_post'):
                require_once(osc_admin_base_path() . 'controller/settings/spamnbots.php');
                $do = new CAdminSettingsSpamNBots();
                break;
            case('currencies'):
                require_once(osc_admin_base_path() . 'controller/settings/currencies.php');
                $do = new CAdminSettingsCurrencies();
                break;
            case('mailserver'):
            case('mailserver_post'):
                require_once(osc_admin_base_path() . 'controller/settings/mailserver.php');
                $do = new CAdminSettingsMailserver();
                break;
            case('media'):
            case('media_post'):
            case('images_post'):
                require_once(osc_admin_base_path() . 'controller/settings/media.php');
                $do = new CAdminSettingsMedia();
                break;
            case('latestsearches'):
            case('latestsearches_post'):
                require_once(osc_admin_base_path() . 'controller/settings/latestsearches.php');
                $do = new CAdminSettingsLatestSearches();
                break;
            case('update'):
            case('check_updates'):
            default:
                require_once(osc_admin_base_path() . 'controller/settings/main.php');
                $do = new CAdminSettingsMain();
                break;
        }
        $do->doModel();
    }
}

/* file end: ./oc-admin/settings.php */

<?php if ( ! defined('ABS_PATH')) exit('ABS_PATH is not loaded. Direct access is not allowed
');

/*
*      Oclass      software for creating and publishing online classified
*                  advertising platforms
*
*                  Copyright (C) 2012 OSCLASS
*
*      This program is free software: you can redistribute it and/or
*      modify it under the terms of the GNU Affero General Public License
*      as published by the Free Software Foundation, either version 3 of
*      the License, or (at your option) any later version.
*
*      This program is distributed in the hope that it will be useful, but
*      WITHOUT ANY WARRANTY; without even the implied warranty of
*      MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
*      GNU Affero General Public License for more details.
*/

```

```

*
*   You should have received a copy of the GNU Affero General Public
*   License along with this program. If not, see <http://www.gnu.org/licenses/>.
*/

class CAdminStats extends AdminSecBaseModel
{
    //specific for this class
    function __construct()
    {
        parent::__construct();

        //specific things for this class
    }

    //Business Layer...
    function doModel()
    {
        parent::doModel();

        //specific things for this class
        switch ($this->action) {
            case('reports'): // manage stats view
                $reports = array();
                if ( Params::getParam('type_stat') == 'week' ) {
                    $stats_reports = Stats::newInstance()->
                        new_reports_count(date('Y-m-d', mktime(0, 0,
                        0, date("m"), date("d") - 70, date("Y")) ), '
                        week');
                    for($k = 10; $k >= 0; $k--) {
                        $reports[date('W', mktime(0, 0, 0, date("m"),
                        date("d"), date("Y")) ) - $k]['views']
                            = 0;
                        $reports[date('W', mktime(0, 0, 0, date("m"),
                        date("d"), date("Y")) ) - $k]['spam']
                            = 0;
                        $reports[date('W', mktime(0, 0, 0, date("m"),
                        date("d"), date("Y")) ) - $k]['repeated']
                            = 0;
                        $reports[date('W', mktime(0, 0, 0, date("m"),
                        date("d"), date("Y")) ) - $k]['
                            bad-classified'] = 0;
                        $reports[date('W', mktime(0, 0, 0, date("m"),
                        date("d"), date("Y")) ) - $k]['offensive
                            '] = 0;
                        $reports[date('W', mktime(0, 0, 0, date("m"),
                        date("d"), date("Y")) ) - $k]['expired']
                            = 0;
                    }
                } else if ( Params::getParam('type_stat') == 'month' )
                {
                    $stats_reports = Stats::newInstance()->
                        new_reports_count(date('Y-m-d', mktime(0, 0,
                        0, date("m") - 10, date("d"), date("Y")) ), '
                        month');
                    for($k = 10; $k >= 0; $k--) {
                        $reports[date('F', mktime(0, 0, 0, date("m")
                        - $k, date("d"), date("Y")) )]['views']
                            = 0;
                        $reports[date('F', mktime(0, 0, 0, date("m")
                        - $k, date("d"), date("Y")) )]['spam']
                            = 0;
                        $reports[date('F', mktime(0, 0, 0, date("m")
                        - $k, date("d"), date("Y")) )]['repeated']
                            = 0;
                        $reports[date('F', mktime(0, 0, 0, date("m")
                        - $k, date("d"), date("Y")) )]['
                            bad-classified'] = 0;
                        $reports[date('F', mktime(0, 0, 0, date("m")
                        - $k, date("d"), date("Y")) )]['offensive
                            '] = 0;
                        $reports[date('F', mktime(0, 0, 0, date("m")
                        - $k, date("d"), date("Y")) )]['expired']
                            = 0;
                    }
                } else {
                    $stats_reports = Stats::newInstance()->
                        new_reports_count(date('Y-m-d', mktime(0, 0,
                        0, date("m"), date("d") - 10, date("Y")) ), '
                        day');
                    for($k = 10; $k >= 0; $k--) {
                        $reports[date('Y-m-d', mktime(0, 0, 0, date("
                        m"), date("d") - $k, date("Y")) )]['views
                            ']
                            = 0;
                        $reports[date('Y-m-d', mktime(0, 0, 0, date("
                        m"), date("d") - $k, date("Y")) )]['spam']
                            = 0;
                        $reports[date('Y-m-d', mktime(0, 0, 0, date("
                        m"), date("d") - $k, date("Y")) )]['
                            repeated']
                            = 0;
                        $reports[date('Y-m-d', mktime(0, 0, 0, date("
                        m"), date("d") - $k, date("Y")) )]['
                            bad-classified'] = 0;
                        $reports[date('Y-m-d', mktime(0, 0, 0, date("
                        m"), date("d") - $k, date("Y")) )]['
                            offensive']
                            = 0;
                        $reports[date('Y-m-d', mktime(0, 0, 0, date("
                        m"), date("d") - $k, date("Y")) )]['
                            expired']
                            = 0;
                    }
                }
            }
        }
    }
}

```

```

    }
}
$max = array();
$max['views'] = 0;
$max['other'] = 0;
foreach($stats_reports as $report) {
    $reports[$report['d_date']]['views'] =
        $report['views'];
    $reports[$report['d_date']]['spam'] =
        $report['spam'];
    $reports[$report['d_date']]['repeated'] =
        $report['repeated'];
    $reports[$report['d_date']]['bad_classified'] =
        $report['bad_classified'];
    $reports[$report['d_date']]['offensive'] =
        $report['offensive'];
    $reports[$report['d_date']]['expired'] =
        $report['expired'];
    if( $report['views']>$max['views'] ) {
        $max['views'] = $report['views'];
    }
    if( $report['spam']>$max['other'] ) {
        $max['other'] = $report['spam'];
    }
    if( $report['repeated']>$max['other'] ) {
        $max['other'] = $report['repeated'];
    }
    if( $report['bad_classified']>$max['other'] ) {
        $max['other'] = $report['bad_classified'];
    }
    if( $report['offensive']>$max['other'] ) {
        $max['other'] = $report['offensive'];
    }
    if( $report['expired']>$max['other'] ) {
        $max['other'] = $report['expired'];
    }
}
$this->_exportVariableToView("reports", $reports);
$this->_exportVariableToView("max", $max);
$this->doView("stats/reports.php");

break;
case('comments') :
    // manage stats view
    $comments = array();
    if( Params::getParam('type_stat') == 'week' ) {
        $stats_comments = Stats::newInstance()->
            new_comments_count(date('Y-m-d H:i:s',
                mktime(0, 0, 0, date("m"), date("d") - 70,
                    date("Y")) ), 'week');
        for($k = 10; $k >= 0; $k--) {
            $comments[date('W', mktime(0, 0, 0, date("m"),
                date("d"), date("Y")) ) - $k] = 0;
        }
    } else if( Params::getParam('type_stat') == 'month' )
    {
        $stats_comments = Stats::newInstance()->
            new_comments_count(date('Y-m-d H:i:s', mktime
                (0, 0, 0, date("m") - 10, date("d"), date("Y")
                ) ), 'month');
        for($k = 10; $k >= 0; $k--) {
            $comments[date('F', mktime(0, 0, 0, date("m")
                - $k, date("d"), date("Y")) )] = 0;
        }
    } else {
        $stats_comments = Stats::newInstance()->
            new_comments_count(date('Y-m-d H:i:s',
                mktime(0, 0, 0, date("m"), date("d") - 10,
                    date("Y")) ), 'day');
        for($k = 10; $k >= 0; $k--) {
            $comments[date('Y-m-d', mktime(0, 0, 0, date
                ("m"), date("d") - $k, date("Y")) )] = 0;
        }
    }
    $max = 0;
    foreach($stats_comments as $comment) {
        $comments[$comment['d_date']] = $comment['num'];
        if( $comment['num'] > $max ) {
            $max = $comment['num'];
        }
    }
    $this->_exportVariableToView("comments", $comments);
    $this->_exportVariableToView("latest_comments", Stats
        ::newInstance()->latest_comments());
    $this->_exportVariableToView("max", $max);
    $this->doView("stats/comments.php");

break;
default :
case('items') :
    // manage stats view
    $items = array();
    $reports = array();
    if( Params::getParam('type_stat') == 'week' ) {
        $stats_items = Stats::newInstance()->
            new_items_count(date('Y-m-d H:i:s', mktime
                (0, 0, 0, date("m"), date("d") - 70, date("Y")
                ) ), 'week');
        $stats_reports = Stats::newInstance()->
            new_reports_count(date('Y-m-d', mktime(0, 0,
                0, date("m"), date("d") - 70, date("Y")) ), '
            week');
        for($k = 10; $k >= 0; $k--) {

```

```

        $reports[date( 'W', mktime(0, 0, 0, date("m"),
            date("d"), date("Y")) ) - $k][ 'views' ] =
            0;
        $items[date( 'W', mktime(0, 0, 0, date("m"),
            date("d"), date("Y")) ) - $k] = 0;
    }
} else if( Params::getParam( 'type-stat' ) == 'month' )
{
    $stats_items = Stats::newInstance()->
        new_items_count(date( 'Y-m-d H:i:s', mktime
            (0, 0, 0, date("m") - 10, date("d"), date("Y"))
            ), 'month');
    $stats_reports = Stats::newInstance()->
        new_reports_count(date( 'Y-m-d', mktime(0, 0,
            0, date("m") - 10, date("d"), date("Y")) ), '
            month');
    for($k = 10; $k >= 0; $k--) {
        $reports[date( 'F', mktime(0, 0, 0, date("m")
            - $k, date("d"), date("Y")) )][ 'views' ] =
            0;
        $items[date( 'F', mktime(0, 0, 0, date("m") -
            $k, date("d"), date("Y")) )] = 0;
    }
} else {
    $stats_items = Stats::newInstance()->
        new_items_count(date( 'Y-m-d H:i:s', mktime
            (0, 0, 0, date("m"), date("d") - 10, date("Y"))
            ), 'day');
    $stats_reports = Stats::newInstance()->
        new_reports_count(date( 'Y-m-d', mktime(0, 0,
            0, date("m"), date("d") - 10, date("Y")) ), '
            day');
    for($k = 10; $k >= 0; $k--) {
        $reports[date( 'Y-m-d', mktime(0, 0, 0, date("
            m"), date("d") - $k, date("Y")) )][ 'views
            ' ] = 0;
        $items[date( 'Y-m-d', mktime(0, 0, 0, date("m
            "), date("d") - $k, date("Y")) )] = 0;
    }
}
$max = 0;
foreach($stats_items as $item) {
    $items[$item['d_date']] = $item['num'];
    if( $item['num'] > $max ) {
        $max = $item['num'];
    }
}
$max_views = 0;
foreach($stats_reports as $report) {
    $reports[$report['d_date']][ 'views' ] = $report[ '
        views' ];
    if( $report[ 'views' ] > $max_views ) {
        $max_views = $report[ 'views' ];
    }
}

$alerts = array();
$subscribers = array();
if( Params::getParam( 'type-stat' ) == 'week' ) {
    $stats_alerts = Stats::newInstance()->
        new_alerts_count(date( 'Y-m-d H:i:s', mktime
            (0, 0, 0, date("m"), date("d") - 70, date("Y"))
            ), 'week');
    $stats_subscribers = Stats::newInstance()->
        new_subscribers_count(date( 'Y-m-d', mktime
            (0, 0, 0, date("m"), date("d") - 70, date("Y"))
            ), 'week');
    for($k = 10; $k >= 0; $k--) {
        $subscribers[date( 'W', mktime(0, 0, 0, date("
            m"), date("d"), date("Y")) ) - $k] = 0;
        $alerts[date( 'W', mktime(0, 0, 0, date("m"),
            date("d"), date("Y")) ) - $k] = 0;
    }
} else if( Params::getParam( 'type-stat' ) == 'month' )
{
    $stats_alerts = Stats::newInstance()->
        new_alerts_count(date( 'Y-m-d H:i:s', mktime
            (0, 0, 0, date("m") - 10, date("d"), date("Y"))
            ), 'month');
    $stats_subscribers = Stats::newInstance()->
        new_subscribers_count(date( 'Y-m-d', mktime
            (0, 0, 0, date("m") - 10, date("d"), date("Y"))
            ), 'month');
    for($k = 10; $k >= 0; $k--) {
        $subscribers[date( 'F', mktime(0, 0, 0, date("
            m") - $k, date("d"), date("Y")) )] = 0;
        $alerts[date( 'F', mktime(0, 0, 0, date("m") -
            $k, date("d"), date("Y")) )] = 0;
    }
} else {
    $stats_alerts = Stats::newInstance()->
        new_alerts_count(date( 'Y-m-d H:i:s', mktime
            (0, 0, 0, date("m"), date("d") - 10, date("Y"))
            ), 'day');
    $stats_subscribers = Stats::newInstance()->
        new_subscribers_count(date( 'Y-m-d', mktime(0,
            0, 0, date("m"), date("d") - 10, date("Y")) )
            , 'day');
}

```

```

        for($k = 10; $k >= 0; $k--) {
            $subscribers[date('Y-m-d', mktime(0, 0, 0,
                date("m"), date("d") - $k, date("Y")))] =
                0;
            $alerts[date('Y-m-d', mktime(0, 0, 0, date("m")
                ), date("d") - $k, date("Y")))] = 0;
        }
    }
    $max = 0;
    $max_alerts = 0;
    foreach($stats_alerts as $alert) {
        $alerts[$alert['d_date']] = $alert['num'];
        if( $alert['num'] > $max ) {
            $max_alerts = $alert['num'];
        }
    }
    $max_subs = 0;
    foreach($stats_subscribers as $subscriber) {
        $subscribers[$subscriber['d_date']] = $subscriber
            ['num'];
        if( $subscriber['num'] > $max_subs ) {
            $max_subs = $subscriber['num'];
        }
    }
}

$this->_exportVariableToView(" reports", $reports);
$this->_exportVariableToView(" items", $items);
$this->_exportVariableToView(" latest_items", Stats::
    newInstance()->latest_items());
$this->_exportVariableToView(" max", $max);
$this->_exportVariableToView(" max_views", $max_views);

$this->_exportVariableToView(" subscribers",
    $subscribers);
$this->_exportVariableToView(" alerts", $alerts);
$this->_exportVariableToView(" max_alerts", $max_alerts
);
$this->_exportVariableToView(" max_subs", $max_subs);

$this->doView(" stats/items.php");

break;
case('users') :
    // manage stats view
    $users = array();
    if( Params::getParam('type_stat') == 'week' ) {
        $stats_users = Stats::newInstance()->
            new_users_count(date('Y-m-d H:i:s', mktime
                (0, 0, 0, date("m"), date("d") - 70, date("Y")
                )), 'week');
        for($k = 10; $k >= 0; $k--) {
            $users[date('W', mktime(0,0,0, date("m"), date
                ("d"), date("Y")))-$k] = 0;
        }
    } else if( Params::getParam('type_stat') == 'month' )
    {
        $stats_users = Stats::newInstance()->
            new_users_count(date('Y-m-d H:i:s', mktime
                (0, 0, 0, date("m") - 10, date("d"), date("Y")
                )), 'month');
        for($k = 10; $k >= 0; $k--) {
            $users[date('F', mktime(0,0,0, date("m")-$k,
                date("d"), date("Y")))] = 0;
        }
    } else {
        $stats_users = Stats::newInstance()->
            new_users_count(date('Y-m-d H:i:s', mktime
                (0, 0, 0, date("m"), date("d") - 10, date("Y")
                )), 'day');
        for($k = 10; $k >= 0; $k--) {
            $users[date('Y-m-d', mktime(0, 0, 0, date("m")
                ), date("d") - $k, date("Y")))] = 0;
        }
    }
    $max = 0;
    foreach($stats_users as $user) {
        $users[$user['d_date']] = $user['num'];
        if($user['num'] > $max) {
            $max = $user['num'];
        }
    }
    $item = Stats::newInstance()->items_by_user();
    $this->_exportVariableToView(" users-by-country", Stats
        ::newInstance()->users_by_country());
    $this->_exportVariableToView(" users-by-region", Stats
        ::newInstance()->users_by_region());
    $this->_exportVariableToView(" item", (!isset($item
        [0]['avg']) || !is-numeric($item[0]['avg'])) ? 0 :
        $item[0]['avg']);
    $this->_exportVariableToView(" latest_users", Stats::
        newInstance()->latest_users());
    $this->_exportVariableToView(" users", $users);
    $this->_exportVariableToView(" max", $max);
    $this->doView(" stats/users.php");

break;
}
}

//hopefully generic...
function doView($file)

```

```

    {
        osc_run_hook(" before_admin.html");
        osc_current_admin_theme_path($file);
        Session::newInstance()->_clearVariables();
        osc_run_hook(" after_admin.html");
    }
}

?> /* file end: ./oc-admin/stats.php */

<?php if ( ! defined('ABS_PATH')) exit('ABS_PATH is not loaded. Direct access is not allowed
. ');

/*
 *      Oclass      software for creating and publishing online classified
 *                  advertising platforms
 *
 *                  Copyright (C) 2012 OSCLASS
 *
 *      This program is free software: you can redistribute it and/or
 *      modify it under the terms of the GNU Affero General Public License
 *      as published by the Free Software Foundation, either version 3 of
 *      the License, or (at your option) any later version.
 *
 *      This program is distributed in the hope that it will be useful, but
 *      WITHOUT ANY WARRANTY; without even the implied warranty of
 *      MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 *      GNU Affero General Public License for more details.
 *
 *      You should have received a copy of the GNU Affero General Public
 *      License along with this program. If not, see <http://www.gnu.org/licenses/>.
 */

class CAdminTools extends AdminSecBaseModel
{
    function __construct()
    {
        parent::__construct();
    }

    //Business Layer...
    function doModel()
    {
        parent::doModel();

        switch($this->action) {
            case('import'): // calling import view
                $this->doView('tools/import.php');

                break;
            case('import_post'): if( defined('DEMO') ) {
                osc_add_flash_warning_message( _m("This action
                    cannot be done because it is a demo site"),
                    'admin');
                $this->redirectTo(osc_admin_base_url(true) . '?
                    page=tools&action=import');
            }
            // calling
            osc_csrf_check();
            $sql = Params::getFiles('sql');
            if( isset($sql['size']) && $sql['size'] != 0 ) {
                $content_file = file_get_contents($sql['tmp_name
                    ']);

                $conn = DBConnectionClass::newInstance();
                $c_db = $conn->getOclassDb();
                $comm = new DBCommandClass($c_db);
                if ( $comm->importSQL($content_file) ) {
                    osc_calculate_location_slug(osc_subdomain_type
                        ());
                    osc_add_flash_ok_message( _m('Import complete
                        '), 'admin');
                } else {
                    osc_add_flash_error_message( _m('There was a
                        problem importing data to the database'),
                        'admin');
                }
            } else {
                osc_add_flash_warning_message( _m('No file was
                    uploaded'), 'admin');
            }
            $this->redirectTo(osc_admin_base_url(true) . '?page=
                tools&action=import');

            break;
            case('images'): // calling images view
                $this->doView('tools/images.php');

                break;
            case('images_post'): if( defined('DEMO') ) {
                osc_add_flash_warning_message( _m("This action
                    cannot be done because it is a demo site"),
                    'admin');
                $this->redirectTo(osc_admin_base_url(true) . '?
                    page=tools&action=images');
            }
        }

        osc_csrf_check();
        $preferences = Preference::newInstance()->toArray();

        $wat = new Watermark();

```

```

$aResources = ItemResource::newInstance()->
    getAllResources();
foreach($aResources as $resource) {
    osc_run_hook('regenerate_image', $resource);

    $spath = osc_uploads_path();
    // comprobar que no haya original
    $img_original = $spath . $resource['pk_i_id'] . "
        _original*";
    $aImages = glob($img_original);
    // there is original image
    if( count($aImages) == 1 ) {
        $image_tmp = $aImages[0];
    } else {
        $img_normal = $spath . $resource['pk_i_id'] .
            ".*";
        $aImages = glob( $img_normal );
        if( count($aImages) == 1 ) {
            $image_tmp = $aImages[0];
        } else {
            $img_thumbnail = $spath . $resource['
                pk_i_id'] . "_thumbnail*";
            $aImages = glob( $img_thumbnail );
            $image_tmp = $aImages[0];
        }
    }

    // extension
    preg_match('/\.(.*)$/', $image_tmp, $matches);
    if( isset($matches[1]) ) {
        $extension = $matches[1];

        // Create normal size
        $path_normal = $spath = osc_uploads_path() .
            $resource['pk_i_id'] . '.jpg';
        $size = explode('x', osc_normal_dimensions());
        ImageResizer::fromFile($image_tmp)->resizeTo(
            $size[0], $size[1])->saveToFile($path);

        if( osc_is_watermark_text() ) {
            $swat->doWatermarkText( $path ,
                osc_watermark_text_color(),
                osc_watermark_text() , 'image/jpeg' );
        } elseif ( osc_is_watermark_image() ){
            $swat->doWatermarkImage( $path , 'image/jpeg
                ');
        }

        // Create preview
        $path = osc_uploads_path() . $resource['
            pk_i_id'] . '_preview.jpg';
        $size = explode('x', osc_preview_dimensions()
            );
        ImageResizer::fromFile($path_normal)->resizeTo
            ($size[0], $size[1])->saveToFile($path);

        // Create thumbnail
        $path = osc_uploads_path() . $resource['
            pk_i_id'] . '_thumbnail.jpg';
        $size = explode('x', osc_thumbnail_dimensions
            ());
        ImageResizer::fromFile($path_normal)->resizeTo
            ($size[0], $size[1])->saveToFile($path);

        // update resource info
        ItemResource::newInstance()->update(
            array(
                's_path' => str_replace(
                    osc_base_path(), '',
                    osc_uploads_path() )
                , 's_name' =>
                    osc_genRandomPassword()
                , 's_extension' => 'jpg'
                , 's_content_type' => 'image/jpeg'
            )
            , array(
                'pk_i_id' => $resource['pk_i_id
                    ']
            )
        );
        osc_run_hook('regenerated_image', ItemResource
            ::newInstance()->findByPrimaryKey(
                $resource['pk_i_id']));
        // si extension es diferente a jpg, eliminar
        // las imagenes con $extension si hay
        if( $extension != 'jpg' ) {
            $files_to_remove = osc_uploads_path() .
                $resource['pk_i_id'] . "*" .
                $extension;
            $fs = glob( $files_to_remove );
            if( is_array($fs) ) {
                array_map( "unlink", $fs );
            }
        }
        // ...
    } else {
        // no es imagen o imagen sin extensi n
    }
}

```

```

    }
    osc_add_flash_ok_message( _m('Re-generation complete')
        , 'admin');
    $this->redirectTo(osc_admin_base_url(true) . '?page=
        tools&action=images');
break;
case('category'): $this->doView('tools/category.php');
break;
case('category_post'): if( defined('DEMO') ) {
    osc_add_flash_warning_message( _m("This action
        cannot be done because it is a demo site"), '
        admin');
    $this->redirectTo(osc_admin_base_url(true) . '?
        page=tools&action=category');
}
osc_update_cat_stats();
osc_add_flash_ok_message(_m("Recount category stats
    has been successful"), 'admin');
$this->redirectTo(osc_admin_base_url(true) . '?page=
    tools&action=category');
break;
case('locations'): $this->doView('tools/locations.php');
break;
case('locations_post'): if( defined('DEMO') ) {
    osc_add_flash_warning_message( _m("This action
        cannot be done because it is a demo site"), '
        admin');
    $this->redirectTo(osc_admin_base_url(true) . '?
        page=tools&action=locations');
}
osc_update_location_stats(true);
$this->redirectTo( osc_admin_base_url(true) . '?page=
    tools&action=locations' );
break;
case('upgrade'): $this->doView('tools/upgrade.php');
break;
case('backup'): $this->doView('tools/backup.php');
break;
case('backup-sql'): if( defined('DEMO') ) {
    osc_add_flash_warning_message( _m("This action
        cannot be done because it is a demo site"), '
        admin');
    $this->redirectTo(osc_admin_base_url(true) . '?
        page=tools&action=backup');
}
osc_csrf_check();
//databasse dump...
if( Params::getParam('bck_dir') != '' ) {
    $path = trim( Params::getParam('bck_dir') );
    if(substr($path, -1, 1) != "/" ) {
        $path .= '/';
    }
} else {
    $path = osc_base_path();
}
$filename = 'Osclass_mysqlbackup.' . date('YmdHis') .
    '.sql';
switch ( osc_dbdump($path, $filename) ) {
    case(-1): $msg = _m('Path is empty');
        osc_add_flash_error_message( $msg, '
            admin');
        break;
    case(-2): $msg = sprintf(_m('Could not connect
        with the database. Error: %s'), mysql_error())
        ;
        osc_add_flash_error_message( $msg, '
            admin');
        break;
    case(-3): $msg = _m('There are no tables to back
        up');
        osc_add_flash_error_message( $msg, '
            admin');
        break;
    case(-4): $msg = _m('The folder is not writable
        ');
        osc_add_flash_error_message( $msg, '
            admin');
        break;
    default: $msg = _m('Backup completed
        successfully');
        osc_add_flash_ok_message( $msg, 'admin
            ');
        break;
}
$this->redirectTo( osc_admin_base_url(true) . '?page=
    tools&action=backup' );
break;
case('backup-sql_file'): if( defined('DEMO') ) {
    osc_add_flash_warning_message( _m("This action
        cannot be done because it is a demo site"), '
        admin');
    $this->redirectTo(osc_admin_base_url(true) . '?'

```



```

        page=tools&action=backup');
    }
    //databasse dump...
    $filename = 'Osclass_mysqlbackup.' . date('YmdHis') .
        '.sql';
    $path = sys_get_temp_dir() . "/";

    switch ( osc_dbdump($path, $filename) ) {
        case(-1):  $msg = _m('Path is empty');
                  osc_add_flash_error_message( $msg, '
                    admin');
                break;
        case(-2):  $msg = sprintf(_m('Could not connect
                    with the database. Error: %s'), mysql_error());
                  ;
                  osc_add_flash_error_message( $msg, '
                    admin');
                break;
        case(-3):  $msg = sprintf(_m('Could not select
                    the database. Error: %s'), mysql_error());
                  osc_add_flash_error_message( $msg, '
                    admin');
                break;
        case(-4):  $msg = _m('There are no tables to back
                    up');
                  osc_add_flash_error_message( $msg, '
                    admin');
                break;
        case(-5):  $msg = _m('The folder is not writable
                    ');
                  osc_add_flash_error_message( $msg, '
                    admin');
                break;
        default:   $msg = _m('Backup completed
                    successfully');
                  osc_add_flash_ok_message( $msg, 'admin
                    ');
                  header('Content-Description: File
                    Transfer');
                  header('Content-Type: application/
                    octet-stream');
                  header('Content-Disposition:
                    attachment; filename='.basename(
                    $filename));
                  header('Content-Transfer-Encoding:
                    binary');
                  header('Expires: 0');
                  header('Cache-Control: must-revalidate
                    , post-check=0, pre-check=0');
                  header('Pragma: public');
                  header('Content-Length: ' . filesize(
                    $path.$filename));
                  flush();
                  readfile($path.$filename);
                  exit;
                break;
    }
    $this->redirectTo( osc_admin_base_url(true) . '?page=
        tools&action=backup' );
break;
case('backup-zip-file'):
    if( defined('DEMO') ) {
        osc_add_flash_warning_message( _m("This action
            cannot be done because it is a demo site"), '
            admin');
        $this->redirectTo(osc_admin_base_url(true) . '?
            page=tools&action=backup');
    }
    $filename = "Osclass_backup." . date('YmdHis') . ".zip
        ";
    $path = sys_get_temp_dir() . "/";

    if ( osc_zip_folder(osc_base_path(),$path. $filename)
        ) {
        $msg = _m('Archived successfully!');
        osc_add_flash_ok_message( $msg, 'admin');
        header('Content-Description: File Transfer');
        header('Content-Type: application/octet-stream');
        header('Content-Disposition: attachment; filename
            ='.basename($filename));
        header('Content-Transfer-Encoding: binary');
        header('Expires: 0');
        header('Cache-Control: must-revalidate , post-check
            =0, pre-check=0');
        header('Pragma: public');
        header('Content-Length: ' . filesize($path.
            $filename));
        flush();
        readfile($path.$filename);
        exit;
    } else {
        $msg = _m('Error, the zip file was not created in
            the specified directory');
        osc_add_flash_error_message( $msg, 'admin');
    }
    $this->redirectTo( osc_admin_base_url(true) . '?page=
        tools&action=backup' );
break;

```

```

case('backup-zip'):
    if( defined('DEMO') ) {
        osc_add_flash_warning_message( _m("This action
            cannot be done because it is a demo site"),
            'admin');
        $this->redirectTo( osc_admin_base_url(true) . '?'
            page=tools&action=backup');
    }
    //zip of the code just to back it up
    osc_csrf_check();
    if( Params::getParam('bck_dir') != '' ) {
        $archive_name = trim( Params::getParam('bck_dir')
            );
        if(substr(trim($archive_name), -1, 1) != '/') {
            $archive_name .= '/';
        }
        $archive_name = Params::getParam('bck_dir') . '/'
            . 'Osclass_backup.' . date('YmdHis') . '.zip';
    } else {
        $archive_name = osc_base_path() . "Osclass_backup
            ." . date('YmdHis') . ".zip";
    }
    $archive_folder = osc_base_path();

    if ( osc_zip_folder($archive_folder, $archive_name) )
    {
        $msg = _m('Archived successfully!');
        osc_add_flash_ok_message( $msg, 'admin');
    } else {
        $msg = _m('Error, the zip file was not created in
            the specified directory');
        osc_add_flash_error_message( $msg, 'admin');
    }
    $this->redirectTo( osc_admin_base_url(true) . '?page=
        tools&action=backup' );

break;
case('backup-post'):
    $this->doView('tools/backup.php');

break;
case('maintenance'):
    if( defined('DEMO') ) {
        osc_add_flash_warning_message( _m("This action
            cannot be done because it is a demo site"),
            'admin');
        $this->doView('tools/maintenance.php');
        break;
    }
    $mode = Params::getParam('mode');
    if( $mode == 'on' ) {
        osc_csrf_check();
        $maintenance_file = osc_base_path() . '.
            maintenance';
        $fileHandler = @fopen($maintenance_file, 'w');
        if( $fileHandler ) {
            osc_add_flash_ok_message( _m('Maintenance mode
                is ON'), 'admin');
        } else {
            osc_add_flash_error_message( _m('There was an
                error creating the .maintenance file ,
                please create it manually at the root
                folder'), 'admin');
        }
        fclose($fileHandler);
        $this->redirectTo( osc_admin_base_url(true) . '?'
            page=tools&action=maintenance' );
    } else if( $mode == 'off' ) {
        osc_csrf_check();
        $deleted = @unlink( osc_base_path() . '.maintenance
            ');
        if( $deleted ) {
            osc_add_flash_ok_message( _m('Maintenance mode
                is OFF'), 'admin');
        } else {
            osc_add_flash_error_message( _m('There was an
                error removing the .maintenance file ,
                please remove it manually from the root
                folder'), 'admin');
        }
        $this->redirectTo( osc_admin_base_url(true) . '?'
            page=tools&action=maintenance' );
    }
    $this->doView('tools/maintenance.php');

break;
default:
}
}
//hopefully generic...
function doView($file)
{
    osc_run_hook(" before_admin.html");
    osc_current_admin_theme_path($file);
    Session::newInstance()->_clearVariables();
    osc_run_hook(" after_admin.html");
}
}
/* file end: ./oc-admin/tools.php */
?>
<?php if ( ! defined('ABSPATH')) exit('ABSPATH is not loaded. Direct access is not allowed

```

```

.);
/*
 *      Oclass      software for creating and publishing online classified
 *                  advertising platforms
 *
 *                  Copyright (C) 2012 OSCLASS
 *
 *      This program is free software: you can redistribute it and/or
 *      modify it under the terms of the GNU Affero General Public License
 *      as published by the Free Software Foundation, either version 3 of
 *      the License, or (at your option) any later version.
 *
 *      This program is distributed in the hope that it will be useful, but
 *      WITHOUT ANY WARRANTY; without even the implied warranty of
 *      MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 *      GNU Affero General Public License for more details.
 *
 *      You should have received a copy of the GNU Affero General Public
 *      License along with this program. If not, see <http://www.gnu.org/licenses/>.
 */

class CAdminUpgrade extends AdminSecBaseModel
{
    function __construct() {
        parent::__construct();
    }

    //Business Layer...
    function doModel() {
        parent::doModel();

        //specific things for this class
        switch ($this->action)
        {
            case 'upgrade-funcs':    require(LIB_PATH.'osclass/upgrade-funcs.php');
            break;
            default:                 $this->doView('upgrade/index.php');
        }
    }

    //hopefully generic...
    function doView($file) {
        osc_run_hook("before_admin_html");
        osc_current_admin_theme_path($file);
        Session::newInstance()->clearVariables();
        osc_run_hook("after_admin_html");
    }
}

?>

<?php if ( ! defined('ABS_PATH')) exit('ABS_PATH is not loaded. Direct access is not allowed
.);

/*
 *      Oclass      software for creating and publishing online classified
 *                  advertising platforms
 *
 *                  Copyright (C) 2012 OSCLASS
 *
 *      This program is free software: you can redistribute it and/or
 *      modify it under the terms of the GNU Affero General Public License
 *      as published by the Free Software Foundation, either version 3 of
 *      the License, or (at your option) any later version.
 *
 *      This program is distributed in the hope that it will be useful, but
 *      WITHOUT ANY WARRANTY; without even the implied warranty of
 *      MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 *      GNU Affero General Public License for more details.
 *
 *      You should have received a copy of the GNU Affero General Public
 *      License along with this program. If not, see <http://www.gnu.org/licenses/>.
 */

class CAdminUsers extends AdminSecBaseModel
{
    //specific for this class
    private $userManager;

    function __construct()
    {
        parent::__construct();

        //specific things for this class
        $this->userManager = User::newInstance();
    }

    //Business Layer...
    function doModel()
    {
        parent::doModel();

        //specific things for this class
        switch ($this->action) {
            case('create'):        // callign create view
                $aCountries = array();
                $aRegions    = array();
                $aCities     = array();
        }
    }
}

```

```

        $aCountries = Country::newInstance()->listAll();

        if( isset($aCountries[0]['pk-c-code']) ) {
            $aRegions = Region::newInstance()->findByCountry(
                $aCountries[0]['pk-c-code']);
        }

        if( isset($aRegions[0]['pk-i-id']) ) {
            $aCities = City::newInstance()->findByRegion(
                $aRegions[0]['pk-i-id']);
        }

        $this->_exportVariableToView( 'user', null );
        $this->_exportVariableToView( 'countries', $aCountries
            );
        $this->_exportVariableToView( 'regions', $aRegions );
        $this->_exportVariableToView( 'cities', $aCities );
        $this->_exportVariableToView( 'locales', OSCLocale::
            newInstance()->listAllEnabled() );

        $this->doView(" users/frm.php");

break;
case('create_post') : // creating the user...
                    osc_csrf_check();
                    require_once LIB_PATH . 'osclass/UserActions.php';
                    $userActions = new UserActions(true);
                    $success      = $userActions->add();

                    switch($success) {
                        case 1: osc_add_flash_ok_message( _m("The user has
                            been created. We've sent an activation e-mail
                            "), 'admin');
                            break;
                        case 2: osc_add_flash_ok_message( _m('The user has
                            been created successfully'), 'admin');
                            break;
                        case 3: osc_add_flash_warning_message( _m('Sorry,
                            but that e-mail is already in use'), 'admin');
                            break;
                        case 5: osc_add_flash_warning_message( _m('The
                            specified e-mail is not valid'), 'admin');
                            break;
                        case 6: osc_add_flash_warning_message( _m('Sorry,
                            the password cannot be empty'), 'admin');
                            break;
                        case 7: osc_add_flash_warning_message( _m("Sorry,
                            passwords don't match"), 'admin');
                            break;
                        case 8: osc_add_flash_warning_message( _m("
                            Username is already taken"), 'admin');
                            break;
                        case 9: osc_add_flash_warning_message( _m("The
                            specified username is not valid, it contains
                            some invalid words"), 'admin');
                            break;
                        case 10: osc_add_flash_warning_message( _m("The
                            name cannot be empty"), 'admin');
                            break;
                        default: osc_add_flash_error_message( _m("There
                            was some error"), 'admin');
                            break;
                    }

                    $this->redirectTo(osc_admin_base_url(true) . '?page=
                        users');

break;
case('edit') : // calling the edit view
                $aUser      = array();
                $aCountries = array();
                $aRegions    = array();
                $aCities     = array();

                $aUser = $this->userManager->findByPrimaryKey(Params::
                    getParam("id"));
                $aCountries = Country::newInstance()->listAll();
                $aRegions    = array();
                if( $aUser['fk-c-country-code'] != '' ) {
                    $aRegions = Region::newInstance()->findByCountry(
                        $aUser['fk-c-country-code']);
                } else if( count($aCountries) > 0 ) {
                    $aRegions = Region::newInstance()->findByCountry(
                        $aCountries[0]['pk-c-code']);
                }
                $aCities = array();
                if( $aUser['fk-i-region-id'] != '' ) {
                    $aCities = City::newInstance()->findByRegion(
                        $aUser['fk-i-region-id']);
                } else if( count($aRegions) > 0 ) {
                    $aCities = City::newInstance()->findByRegion(
                        $aRegions[0]['pk-i-id']);
                }

                $this->_exportVariableToView(" user", $aUser);
                $this->_exportVariableToView(" countries", $aCountries)
                    ;
                $this->_exportVariableToView(" regions", $aRegions);
                $this->_exportVariableToView(" cities", $aCities);
                $this->_exportVariableToView(" locales", OSCLocale::

```

```

        newInstance()->listAllEnabled();
$this->doView("users/frm.php");
break;
case('edit_post'):
    // edit post
    osc_csrf_check();
    require_once LIB_PATH . 'osclass/UserActions.php';
    $userActions = new UserActions(true);
    $success = $userActions->edit(Params::getParam("id")
    );

    switch($success) {
        case 1:
            osc_add_flash_error_message( _m("Passwords don
            't match"), 'admin');
            $this->redirectTo(osc_admin_base_url(true) .
            '?page=users&action=edit&id=' . Params::
            getParam('id'));
            break;
        case 2: osc_add_flash_ok_message( _m('The user
            has been updated and activated'), 'admin');
            break;
        case 3:
            osc_add_flash_error_message( _m('E-mail is
            already in used by another user'), 'admin
            ');
            $this->redirectTo(osc_admin_base_url(true) .
            '?page=users&action=edit&id=' . Params::
            getParam('id'));
            break;
        case 10: osc_add_flash_error_message( _m("The name
            cannot be empty"), 'admin');
            $this->redirectTo(osc_admin_base_url(true) .
            '?page=users&action=edit&id=' . Params::
            getParam('id'));
            break;
        default: osc_add_flash_ok_message( _m('The user
            has been updated'), 'admin');
            break;
    }

    $this->redirectTo(osc_admin_base_url(true) . '?page=
    users');
break;
case('resend_activation'):
    // activate
    osc_csrf_check();
    require_once LIB_PATH . 'osclass/UserActions.php';
    $iUpdated = 0;
    $userId = Params::getParam('id');
    if(!is_array($userId)) {
        osc_add_flash_error_message( _m("User id isn't in
        the correct format"), 'admin');
        $this->redirectTo(osc_admin_base_url(true) . '?
        page=users');
    }

    $userActions = new UserActions(true);
    foreach($userId as $id) {
        $iUpdated += $userActions->resend_activation($id
        );
    }

    if($iUpdated==0) {
        osc_add_flash_error_message(_m('No users have been
        selected'), 'admin');
    } else {
        osc_add_flash_ok_message(sprintf( _mn('Activation
        email sent to one user', 'Activation email
        sent to %s users', $iUpdated), $iUpdated), '
        admin');
    }

    $this->redirectTo(osc_admin_base_url(true) . '?page=
    users');
break;
case('activate'):
    // activate
    osc_csrf_check();
    require_once LIB_PATH . 'osclass/UserActions.php';
    $iUpdated = 0;
    $userId = Params::getParam('id');
    if( !is_array($userId) ) {
        osc_add_flash_error_message( _m("User id isn't in
        the correct format"), 'admin');
        $this->redirectTo(osc_admin_base_url(true) . '?
        page=users');
    }

    $userActions = new UserActions(true);
    foreach($userId as $id) {
        $iUpdated += $userActions->activate($id);
    }

    if( $iUpdated == 0 ) {
        $msg = _m('No users have been activated');
    } else {
        $msg = sprintf( _mn('One user has been activated',
        '%s users have been activated', $iUpdated),
        $iUpdated );
    }
}

```

```

osc_add_flash_ok_message($msg, 'admin');
$this->redirectTo(osc_admin_base_url(true) . '?page=
users ');
break;
case('deactivate'):
//deactivate
osc_csrf_check();
require_once LIB_PATH . 'osclass/UserActions.php';
$iUpdated = 0;
$userId = Params::getParam('id');

if( !is_array($userId) ) {
osc_add_flash_error_message( _m("User id isn't in
the correct format"), 'admin');
$this->redirectTo(osc_admin_base_url(true) . '?
page=users ');
}

$userActions = new UserActions(true);
foreach($userId as $id) {
$iUpdated += $userActions->deactivate($id);
}

if( $iUpdated == 0 ) {
$msg = _m('No users have been deactivated');
} else {
$msg = sprintf( _mn('One user has been deactivated
', '%s users have been deactivated', $iUpdated
), $iUpdated );
}

osc_add_flash_ok_message($msg, 'admin');
$this->redirectTo(osc_admin_base_url(true) . '?page=
users ');

break;
case('enable'):

osc_csrf_check();
require_once LIB_PATH . 'osclass/UserActions.php';
$iUpdated = 0;
$userId = Params::getParam('id');
if( !is_array($userId) ) {
osc_add_flash_error_message( _m("User id isn't in
the correct format"), 'admin');
$this->redirectTo(osc_admin_base_url(true) . '?
page=users ');
}

$userActions = new UserActions(true);
foreach($userId as $id) {
$iUpdated += $userActions->enable($id);
}

if( $iUpdated == 0 ) {
$msg = _m('No users have been enabled');
} else {
$msg = sprintf( _mn('One user has been unblocked',
'%s users have been unblocked', $iUpdated),
$iUpdated );
}

osc_add_flash_ok_message($msg, 'admin');
$this->redirectTo(osc_admin_base_url(true) . '?page=
users ');

break;
case('disable'):

osc_csrf_check();
require_once LIB_PATH . 'osclass/UserActions.php';
$iUpdated = 0;
$userId = Params::getParam('id');
if( !is_array($userId) ) {
osc_add_flash_error_message( _m("User id isn't in
the correct format"), 'admin');
$this->redirectTo(osc_admin_base_url(true) . '?
page=users ');
}

$userActions = new UserActions(true);
foreach($userId as $id) {
$iUpdated += $userActions->disable($id);
}

if( $iUpdated == 0 ) {
$msg = _m('No users have been disabled');
} else {
$msg = sprintf( _mn('One user has been blocked',
'%s users have been blocked', $iUpdated),
$iUpdated );
}

osc_add_flash_ok_message($msg, 'admin');
$this->redirectTo(osc_admin_base_url(true) . '?page=
users ');

break;
case('delete'):
//delete
osc_csrf_check();
$iDeleted = 0;
$userId = Params::getParam('id');

if( !is_array($userId) ) {

```

```

        osc_add_flash_error_message( _m("User id isn't in
        the correct format"), 'admin');
        $this->redirectTo(osc_admin_base_url(true) . '?
        page=users ');
    }

    foreach($UserId as $id) {
        $user = $this->userManager->findByPrimaryKey($id);
        Log::newInstance()->insertLog('user', 'delete ',
            $id, $user['s_email'], 'admin',
            osc_logged_admin_id());
        if( $this->userManager->deleteUser($id) ) {
            $iDeleted++;
        }
    }

    if( $iDeleted == 0 ) {
        $msg = _m('No users have been deleted');
    } else {
        $msg = sprintf( _mn('One user has been deleted',
            '%s users have been deleted', $iDeleted),
            $iDeleted );
    }

    osc_add_flash_ok_message($msg, 'admin');
    $this->redirectTo(osc_admin_base_url(true) . '?page=
    users ');

break;
case('delete_alerts') :                //delete

    $iDeleted = 0;
    $alertId = Params::getParam('alert_id');
    if( !is_array($alertId) ) {
        osc_add_flash_error_message( _m("Alert id isn't in
        the correct format"), 'admin');
        if(Params::getParam('user_id')=='' ) {
            $this->redirectTo(osc_admin_base_url(true) .
            '?page=users&action=alerts ');
        } else {
            $this->redirectTo(osc_admin_base_url(true) .
            '?page=users&action=edit&id='.Params::
            getParam('user_id '));
        }
    }

    $mAlerts = new Alerts();
    foreach($alertId as $id) {
        Log::newInstance()->insertLog('user', '
        delete_alerts ', $id, $id, 'admin',
            osc_logged_admin_id());
        $iDeleted += $mAlerts->delete(array('pk_i_id' =>
            $id));
    }

    if( $iDeleted == 0 ) {
        $msg = _m('No alerts have been deleted');
    } else {
        $msg = sprintf( _mn('One alert has been deleted',
            '%s alerts have been deleted', $iDeleted),
            $iDeleted );
    }

    osc_add_flash_ok_message($msg, 'admin');
    if(Params::getParam('user_id')=='' ) {
        $this->redirectTo(osc_admin_base_url(true) . '?
        page=users&action=alerts ');
    } else {
        $this->redirectTo(osc_admin_base_url(true) . '?
        page=users&action=edit&id='.Params::getParam('
        user_id '));
    }

break;
case('status_alerts') :                //delete

    $status = Params::getParam("status");
    $iUpdated = 0;
    $alertId = Params::getParam('alert_id');

    if( !is_array($alertId) ) {
        osc_add_flash_error_message( _m("Alert id isn't in
        the correct format"), 'admin');
        if(Params::getParam('user_id')=='' ) {
            $this->redirectTo(osc_admin_base_url(true) .
            '?page=users&action=alerts ');
        } else {
            $this->redirectTo(osc_admin_base_url(true) .
            '?page=users&action=edit&id='.Params::
            getParam('user_id '));
        }
    }

    $mAlerts = new Alerts();
    foreach($alertId as $id) {
        if($status==1) {
            $iUpdated += $mAlerts->activate($id);
        } else {
            $iUpdated += $mAlerts->deactivate($id);
        }
    }
}

```

```

if($status==1) {
    if( $iUpdated == 0 ) {
        $msg = _m('No alerts have been activated');
    } else {
        $msg = sprintf( _mn('One alert has been
activated', '%s alerts have been activated
', $iUpdated), $iUpdated );
    }
} else {
    if( $iUpdated == 0 ) {
        $msg = _m('No alerts have been deactivated');
    } else {
        $msg = sprintf( _mn('One alert has been
deactivated', '%s alerts have been
deactivated', $iUpdated), $iUpdated );
    }
}

osc_add_flash_ok_message($msg, 'admin');
if(Params::getParam('user_id')=='' ) {
    $this->redirectTo(osc_admin_base_url(true) . '?
page=users&action=alerts');
} else {
    $this->redirectTo(osc_admin_base_url(true) . '?
page=users&action=edit&id='.Params::getParam('
user_id'));
}

break;
case('settings') : // calling the users settings view
$this->doView('users/settings.php');

break;
case('settings_post') : // updating users
osc_csrf_check();
$iUpdated = 0;
$enabledUserValidation = Params::getParam('
enabled_user_validation');
$enabledUserValidation = (($enabledUserValidation !=
'' ) ? true : false);
$enabledUserRegistration = Params::getParam('
enabled_user_registration');
$enabledUserRegistration = (($enabledUserRegistration
!= '' ) ? true : false);
$enabledUsers = Params::getParam('
enabled_users');
$enabledUsers = (($enabledUsers != '' ) ?
true : false);
$notifyNewUser = Params::getParam('
notify_new_user');
$notifyNewUser = (($notifyNewUser != '' ) ?
true : false);
$usernameBlacklistTmp = explode(", ", Params::
getParam('username_blacklist'));
foreach($usernameBlacklistTmp as $k => $v) {
    $usernameBlacklistTmp[$k] = strtolower(trim($v));
}
$usernameBlacklist = implode(", ",
$usernameBlacklistTmp);

$iUpdated += osc_set_preference('
enabled_user_validation', $enabledUserValidation);
$iUpdated += osc_set_preference('
enabled_user_registration',
$enabledUserRegistration);
$iUpdated += osc_set_preference('enabled_users',
$enabledUsers);
$iUpdated += osc_set_preference('notify_new_user',
$notifyNewUser);
$iUpdated += osc_set_preference('username_blacklist',
$usernameBlacklist);

if( $iUpdated > 0 ) {
    osc_add_flash_ok_message( _m("User settings have
been updated"), 'admin');
}
$this->redirectTo(osc_admin_base_url(true) . '?page=
users&action=settings');

break;
case('alerts') : // manage alerts view
require_once osc_lib_path()."osclass/classes/
datatables/AlertsDataTable.php";

// set default iDisplayLength
if( Params::getParam('iDisplayLength') != '' ) {
    Cookie::newInstance()->push('
listing_iDisplayLength', Params::getParam('
iDisplayLength'));
    Cookie::newInstance()->set();
} else {
    // set a default value if it's set in the cookie
    if( Cookie::newInstance()->get_value('
listing_iDisplayLength') != '' ) {
        Params::setParam('iDisplayLength', Cookie::
newInstance()->get_value('
listing_iDisplayLength'));
    } else {
        Params::setParam('iDisplayLength', 10 );
    }
}

```



```

}
$this->exportVariableToView('iDisplayLength', Params
::getParam('iDisplayLength'));

// Table header order by related
if( Params::getParam('sort') == '' ) {
    Params::setParam('sort', 'date');
}
if( Params::getParam('direction') == '' ) {
    Params::setParam('direction', 'desc');
}

$page = (int)Params::getParam('iPage');
if($page==0) { $page = 1; };
Params::setParam('iPage', $page);

$params = Params::getParamsAsArray("get");

$alertsDataTable = new AlertsDataTable();
$alertsDataTable->table($params);
$aData = $alertsDataTable->getData();

if(count($aData['aRows']) == 0 && $page!=1) {
    $total = (int)$aData['iTotalDisplayRecords'];
    $maxPage = ceil( $total / (int)$aData['iDisplayLength'] );

    $url = osc_admin_base_url(true) . '?' . $_SERVER['QUERY_STRING'];

    if($maxPage==0) {
        $url = preg_replace('/&iPage=(\d)+/', '&iPage=1', $url);
        $this->redirectTo($url);
    }

    if($page > 1) {
        $url = preg_replace('/&iPage=(\d)+/', '&iPage='.$maxPage, $url);
        $this->redirectTo($url);
    }
}

$this->exportVariableToView('aData', $aData);
$this->exportVariableToView('aRawRows', $alertsDataTable->rawRows());

$this->doView("users/alerts.php");

break;
case('ban'):
    // manage ban rules view

    if(Params::getParam("action")!="") {
        osc_run_hook("ban_rules_bulk_" . Params::getParam("action"), Params::getParam('id'));
    }

    require_once osc_lib_path()."osclass/classes/datatables/BanRulesDataTable.php";

    // set default iDisplayLength
    if( Params::getParam('iDisplayLength') != '' ) {
        Cookie::newInstance()->push('listing-iDisplayLength', Params::getParam('iDisplayLength'));
        Cookie::newInstance()->set();
    } else {
        // set a default value if it's set in the cookie
        if( Cookie::newInstance()->get_value('listing-iDisplayLength') != '' ) {
            Params::setParam('iDisplayLength', Cookie::newInstance()->get_value('listing-iDisplayLength'));
        } else {
            Params::setParam('iDisplayLength', 10 );
        }
    }
    $this->exportVariableToView('iDisplayLength', Params::getParam('iDisplayLength'));

    // Table header order by related
    if( Params::getParam('sort') == '' ) {
        Params::setParam('sort', 'date');
    }
    if( Params::getParam('direction') == '' ) {
        Params::setParam('direction', 'desc');
    }

    $page = (int)Params::getParam('iPage');
    if($page==0) { $page = 1; };
    Params::setParam('iPage', $page);

    $params = Params::getParamsAsArray("get");

    $banRulesDataTable = new BanRulesDataTable();
    $banRulesDataTable->table($params);
    $aData = $banRulesDataTable->getData();

    if(count($aData['aRows']) == 0 && $page!=1) {

```

```

$total = (int)$aData['iTotalDisplayRecords'];
$maxPage = ceil( $total / (int)$aData['iDisplayLength'] );

$url = osc_admin_base_url(true) . '?' . $_SERVER['QUERY_STRING'];

if($maxPage==0) {
    $url = preg_replace('/&iPage=(\d)+/', '&iPage=1', $url);
    $this->redirectTo($url);
}

if($page > 1) {
    $url = preg_replace('/&iPage=(\d)+/', '&iPage='.$maxPage, $url);
    $this->redirectTo($url);
}
}

$this->_exportVariableToView('aData', $aData);
$this->_exportVariableToView('aRawRows', $banRulesDataTable->rawRows());

$bulk_options = array(
    array('value' => '', 'data-dialog-content' => '', 'label' => __('Bulk actions')),
    array('value' => 'delete_ban_rule', 'data-dialog-content' => sprintf(__('Are you sure you want to %s the selected ban rules?'), strtolower(__('Delete'))), 'label' => __('Delete'))
);

$bulk_options = osc_apply_filter("ban_rule_bulk_filter", $bulk_options);
$this->_exportVariableToView('bulk_options', $bulk_options);

//calling the view...
$this->doView('users/ban.php');

break;
case('edit_ban_rule'):
    $this->_exportVariableToView('rule', BanRule::newInstance()->findByPrimaryKey(Params::getParam('id')));
    $this->doView('users/ban_frm.php');

break;
case('edit_ban_rule_post'):
    osc_csrf_check();
    if(Params::getParam('s_ip')=='' && Params::getParam('s_email')=='' ) {
        osc_add_flash_warning_message( _m("Both rules can not be empty"), 'admin');
        $this->redirectTo(osc_admin_base_url(true) . '?page=users&action=ban');
    }

    BanRule::newInstance()->update(array('s_name' => Params::getParam('s_name'), 's_ip' => Params::getParam('s_ip'), 's_email' => Params::getParam('s_email')), array('pk_i_id' => Params::getParam('id')));
    osc_add_flash_ok_message( _m('Rule updated correctly'), 'admin');
    $this->redirectTo(osc_admin_base_url(true) . '?page=users&action=ban');

break;
case('create_ban_rule'):
    $this->_exportVariableToView('rule', null);
    $this->doView('users/ban_frm.php');

break;
case('create_ban_rule_post'):
    osc_csrf_check();
    if(Params::getParam('s_ip')=='' && Params::getParam('s_email')=='' ) {
        osc_add_flash_warning_message( _m("Both rules can not be empty"), 'admin');
        $this->redirectTo(osc_admin_base_url(true) . '?page=users&action=ban');
    }

    BanRule::newInstance()->insert(array('s_name' => Params::getParam('s_name'), 's_ip' => Params::getParam('s_ip'), 's_email' => Params::getParam('s_email')));
    osc_add_flash_ok_message( _m('Rule saved correctly'), 'admin');
    $this->redirectTo(osc_admin_base_url(true) . '?page=users&action=ban');

break;
case('delete_ban_rule'):
    //delete ban rules
    osc_csrf_check();
    $iDeleted = 0;
    $ruleId = Params::getParam('id');

    if( !is_array($ruleId) ) {

```

```

        osc_add_flash_error_message( _m("User id isn't in
        the correct format"), 'admin');
        $this->redirectTo(osc_admin_base_url(true) . '?'
        page=users&action=ban');
    }

    $ruleMgr = BanRule::newInstance();
    foreach($ruleId as $id) {
        if($ruleMgr->deleteByPrimaryKey($id)) {
            $iDeleted++;
        }
    }

    if( $iDeleted == 0 ) {
        $msg = _m('No rules have been deleted');
    } else {
        $msg = sprintf( _mn('One ban rule has been deleted
        ', '%s ban rules have been deleted', $iDeleted
        ), $iDeleted );
    }

    osc_add_flash_ok_message($msg, 'admin');
    $this->redirectTo(osc_admin_base_url(true) . '?page=
    users&action=ban');

break;
default:

    // manage users view

    if(Params::getParam("action")!="") {
        osc_run_hook("user_bulk_", Params::getParam("action
        "), Params::getParam('id'));
    }

    require_once osc_lib_path()."osclass/classes/
    datatables/UsersDataTable.php";

    // set default iDisplayLength
    if( Params::getParam('iDisplayLength') != '' ) {
        Cookie::newInstance()->push('
        listing-iDisplayLength', Params::getParam('
        iDisplayLength'));
        Cookie::newInstance()->set();
    } else {
        // set a default value if it's set in the cookie
        if( Cookie::newInstance()->get_value('
        listing-iDisplayLength') != '' ) {
            Params::setParam('iDisplayLength', Cookie::
            newInstance()->get_value('
            listing-iDisplayLength'));
        } else {
            Params::setParam('iDisplayLength', 10 );
        }
    }
    $this->exportVariableToView('iDisplayLength', Params
    ::getParam('iDisplayLength'));

    // Table header order by related
    if( Params::getParam('sort') == '' ) {
        Params::setParam('sort', 'date');
    }
    if( Params::getParam('direction') == '' ) {
        Params::setParam('direction', 'desc');
    }
    }

    $page = (int)Params::getParam('iPage');
    if($page==0) { $page = 1; };
    Params::setParam('iPage', $page);

    $params = Params::getParamsAsArray("get");

    $usersDataTable = new UsersDataTable();
    $usersDataTable->table($params);
    $aData = $usersDataTable->getData();

    if(count($aData['aRows']) == 0 && $page!=1) {
        $total = (int)$aData['iTotalDisplayRecords'];
        $maxPage = ceil( $total / (int)$aData['
        iDisplayLength'] );

        $url = osc_admin_base_url(true) . '?' . $_SERVER['
        QUERY_STRING'];

        if($maxPage==0) {
            $url = preg_replace('/&iPage=(\d)+/', '&iPage
            =1', $url);
            $this->redirectTo($url);
        }

        if($page > 1) {
            $url = preg_replace('/&iPage=(\d)+/', '&iPage
            ='. $maxPage, $url);
            $this->redirectTo($url);
        }
    }

    $this->exportVariableToView('aData', $aData);
    $this->exportVariableToView('aRawRows',
    $usersDataTable->rawRows());

```

```

        $bulk_options = array(
            array('value' => '', 'data-dialog-content' => '',
                'label' => __('Bulk actions')),
            array('value' => 'activate', 'data-dialog-content'
                => sprintf(__('Are you sure you want to %s
                the selected users?'), strtolower(__('Activate
                '))), 'label' => __('Activate')),
            array('value' => 'deactivate', 'data-dialog-
                content' => sprintf(__('Are you sure you want
                to %s the selected users?'), strtolower(__('
                Deactivate'))), 'label' => __('Deactivate')),
            array('value' => 'enable', 'data-dialog-content'
                => sprintf(__('Are you sure you want to %s the
                selected users?'), strtolower(__('Unblock'))),
                'label' => __('Unblock')),
            array('value' => 'disable', 'data-dialog-content'
                => sprintf(__('Are you sure you want to %s the
                selected users?'), strtolower(__('Block'))),
                'label' => __('Block')),
            array('value' => 'delete', 'data-dialog-content'
                => sprintf(__('Are you sure you want to %s the
                selected users?'), strtolower(__('Delete'))),
                'label' => __('Delete'))
        );
        if( osc_user_validation_enabled() ) {
            $bulk_options[] = array('value' => '
                resend.activation', 'data-dialog-content' =>
                sprintf(__('Are you sure you want to %s the
                selected users?'), strtolower(__('Resend the
                activation to'))), 'label' => __('Resend
                activation'));
        }

        $bulk_options = osc_apply_filter("user_bulk_filter",
            $bulk_options);
        $this->exportVariableToView('bulk_options',
            $bulk_options);

        //calling the view...
        $this->doView('users/index.php');
    }
}

break;
}

//hopefully generic...
function doView($file)
{
    osc_run_hook("before_admin_html");
    osc_current_admin_theme_path($file);
    Session::newInstance()->.clearVariables();
    osc_run_hook("after_admin_html");
}

}

/* file end: ./oc-admin/users.php */
?>

<?php if ( ! defined('ABSPATH')) exit('ABSPATH is not loaded. Direct access is not allowed
.');
```

```

/*
 *   Oclass      software for creating and publishing online classified
 *               advertising platforms
 *
 *               Copyright (C) 2012 OSCLASS
 *
 *   This program is free software: you can redistribute it and/or
 *   modify it under the terms of the GNU Affero General Public License
 *   as published by the Free Software Foundation, either version 3 of
 *   the License, or (at your option) any later version.
 *
 *   This program is distributed in the hope that it will be useful, but
 *   WITHOUT ANY WARRANTY; without even the implied warranty of
 *   MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 *   GNU Affero General Public License for more details.
 *
 *   You should have received a copy of the GNU Affero General Public
 *   License along with this program. If not, see <http://www.gnu.org/licenses/>.
 */

class AdminThemes extends Themes
{
    private static $instance;

    public static function newInstance()
    {
        if(!self::$instance instanceof self) {
            self::$instance = new self;
        }
        return self::$instance;
    }

    public function __construct()
    {
        parent::__construct();
        $this->setCurrentTheme( osc_admin_theme() );
    }
}

```

```

public function setCurrentThemeUrl()
{
    if ($this->theme_exists) {
        $this->theme_url = osc_admin_base_url() . 'themes/' . $this->theme . '/';
    } else {
        $this->theme_url = osc_admin_base_url() . 'gui/';
    }
}

public function setCurrentThemePath()
{
    if (file_exists(osc_admin_base_path() . 'themes/' . $this->theme . '/')) {
        $this->theme_exists = true;
        $this->theme_path = osc_admin_base_path() . 'themes/' . $this->theme . '/';
    } else {
        $this->theme_exists = false;
        $this->theme_path = osc_admin_base_path() . 'gui/';
    }
}
}

/* file end: ./oc-includes/osclass/AdminThemes.php */

<?php
/*
 *      Osclass      software for creating and publishing online classified
 *                  advertising platforms
 *
 *                  Copyright (C) 2012 OSCCLASS
 *
 *      This program is free software: you can redistribute it and/or
 *      modify it under the terms of the GNU Affero General Public License
 *      as published by the Free Software Foundation, either version 3 of
 *      the License, or (at your option) any later version.
 *
 *      This program is distributed in the hope that it will be useful, but
 *      WITHOUT ANY WARRANTY; without even the implied warranty of
 *      MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 *      GNU Affero General Public License for more details.
 *
 *      You should have received a copy of the GNU Affero General Public
 *      License along with this program. If not, see <http://www.gnu.org/licenses/>.
 */

function osc_runAlert($type = null) {
    if ( !in_array($type, array('HOURLY', 'DAILY', 'WEEKLY', 'INSTANT')) ) {
        return;
    }

    $internal_name = 'alert_email_hourly';
    switch ($type) {
        case 'HOURLY':
            $internal_name = 'alert_email_hourly';
            break;
        case 'DAILY':
            $internal_name = 'alert_email_daily';
            break;
        case 'WEEKLY':
            $internal_name = 'alert_email_weekly';
            break;
        case 'INSTANT':
            $internal_name = 'alert_email_instant';
            break;
    }

    $active = TRUE;
    $searches = Alerts::newInstance()->findByType($type, $active);
    foreach($searches as $s_search) {
        // Get if there're new ads on this search
        $json = base64_decode($s_search['s_search']);
        $array_conditions = (array)json_decode($json);

        $new_search = Search::newInstance();
        $new_search->setJsonAlert($array_conditions);

        $cron = Cron::newInstance()->getCronByType($type);
        if( is_array($cron) ) {
            $last_exec = $cron['d_last_exec'];
        } else {
            $last_exec = '0000-00-00 00:00:00';
        }

        $new_search->addConditions(sprintf(" %st_item.dt_pub_date > '%s' ",
            DB_TABLE_PREFIX, $last_exec));

        $items = $new_search->doSearch();
        $totalItems = $new_search->count();

        if( count($items) > 0 ) {
            // If we have new items from last check
            // Catch the user subscribed to this search
            $users = Alerts::newInstance()->findUsersBySearchAndType($s_search['s_search'],
                $type, $active);

            if( count($users) > 0 ) {
                $sads = '';
                foreach($items as $item) {
                    $sads .= '<a href="'. osc_item_url_ns($item['pk_i_id']).">' . $item['s_title'] . '</a><br/>';
                }
            }
        }
    }
}

```



```

/*
 *      Oclass      software for creating and publishing online classified
 *                  advertising platforms
 *
 *                  Copyright (C) 2012 OSCLASS
 *
 *      This program is free software: you can redistribute it and/or
 *      modify it under the terms of the GNU Affero General Public License
 *      as published by the Free Software Foundation, either version 3 of
 *      the License, or (at your option) any later version.
 *
 *      This program is distributed in the hope that it will be useful, but
 *      WITHOUT ANY WARRANTY; without even the implied warranty of
 *      MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 *      GNU Affero General Public License for more details.
 *
 *      You should have received a copy of the GNU Affero General Public
 *      License along with this program. If not, see <http://www.gnu.org/licenses/>.
 */

$d_now = date('Y-m-d H:i:s');
$i_now = strtotime($d_now);

// Hourly crons
$cron = Cron::newInstance()->getCronByType('HOURLY');
if( is_array($cron) ) {
    $i_next = strtotime($cron['d_next_exec']);

    if( (CLI && (Params::getParam('cron-type') == 'hourly')) || ((( $i_now - $i_next ) >=
        0) && !CLI) ) {
        // update the next execution time in t_cron
        $d_next = date('Y-m-d H:i:s', $i_now + 3600);
        Cron::newInstance()->update(array('d_last_exec' => $d_now, 'd_next_exec' =>
            $d_next),
            array('e-type' => 'HOURLY'));

        // Run cron AFTER updating the next execution time to avoid double run of cron
        $purge = osc_purge_latest_searches();
        if( $purge == 'hour' ) {
            LatestSearches::newInstance()->purgeDate( date('Y-m-d H:i:s', ( time() - 3600)
                ) );
        } else if( !in_array($purge, array('forever', 'day', 'week')) ) {
            LatestSearches::newInstance()->purgeNumber($purge);
        }
        osc_update_location_stats(true, 'auto');

        // WARN EXPIRATION EACH HOUR (COMMENT TO DISABLE)
        // NOTE: IF THIS IS ENABLE, SAME CODE SHOULD BE DISABLE ON CRON DAILY
        if(is_numeric(osc_warn_expiration()) && osc_warn_expiration() > 0) {
            $items = Item::newInstance()->findByHourExpiration(24*osc_warn_expiration());
            foreach($items as $item) {
                osc_run_hook('hook_email_warn_expiration', $item);
            }
        }

        osc_run_hook('cron-hourly');
    }
}

// Daily crons
$cron = Cron::newInstance()->getCronByType('DAILY');
if( is_array($cron) ) {
    $i_next = strtotime($cron['d_next_exec']);

    if( (CLI && (Params::getParam('cron-type') == 'daily')) || ((( $i_now - $i_next ) >= 0)
        && !CLI) ) {
        // before update, d_last_exec
        osc_runAlert('DAILY');

        // update the next execution time in t_cron
        $d_next = date('Y-m-d H:i:s', $i_now + (24 * 3600));
        Cron::newInstance()->update(array('d_last_exec' => $d_now, 'd_next_exec' =>
            $d_next),
            array('e-type' => 'DAILY'));

        // Run cron AFTER updating the next execution time to avoid double run of cron
        $purge = osc_purge_latest_searches();
        if( $purge == 'day' ) {
            LatestSearches::newInstance()->purgeDate( date('Y-m-d H:i:s', ( time() - (24 *
                3600) ) ) );
        }
        osc_update_cat_stats();

        // WARN EXPIRATION EACH DAY (UNCOMMENT TO ENABLE)
        // NOTE: IF THIS IS ENABLE, SAME CODE SHOULD BE DISABLE ON CRON HOURLY
        // *if(is_numeric(osc_warn_expiration()) && osc_warn_expiration() > 0) {
        $items = Item::newInstance()->findByDayExpiration(osc_warn_expiration());
        foreach($items as $item) {
            osc_run_hook('hook_email_warn_expiration', $item);
        }
        // */

        osc_run_hook('cron-daily');
    }
}

// Weekly crons
$cron = Cron::newInstance()->getCronByType('WEEKLY');

```

```

if( is_array($cron) ) {
    $i_next = strtotime($cron['d_next_exec']);

    if( ( CLI && ( Params::getParam('cron-type') == 'weekly' ) ) || ((( $i_now - $i_next ) >=
        0) && !CLI) ) {
        // update the next execution time in t_cron
        $d_next = date('Y-m-d H:i:s', $i_now + (7 * 24 * 3600));
        Cron::newInstance()->update(array('d_last_exec' => $d_now, 'd_next_exec' =>
            $d_next),
            array('e_type' => 'WEEKLY'));

        // Run cron AFTER updating the next execution time to avoid double run of cron
        $purge = osc_purge_latest_searches();
        if( $purge == 'week' ) {
            LatestSearches::newInstance()->purgeDate( date('Y-m-d H:i:s', ( time() - (7 *
                24 * 3600) ) ) );
        }
        osc_run_hook('cron_weekly');
    }
}

osc_run_hook('cron');
/* file end: ./oc-includes/osclass/cron.php */

<?php if ( ! defined('ABSPATH')) exit('ABSPATH is not loaded. Direct access is not allowed
. ');

/*
 *      Oclass      software for creating and publishing online classified
 *                  advertising platforms
 *
 *                  Copyright (C) 2012 OSCLASS
 *
 *      This program is free software: you can redistribute it and/or
 *      modify it under the terms of the GNU Affero General Public License
 *      as published by the Free Software Foundation, either version 3 of
 *      the License, or (at your option) any later version.
 *
 *      This program is distributed in the hope that it will be useful, but
 *      WITHOUT ANY WARRANTY; without even the implied warranty of
 *      MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 *      GNU Affero General Public License for more details.
 *
 *      You should have received a copy of the GNU Affero General Public
 *      License along with this program. If not, see <http://www.gnu.org/licenses/>.
 */

class DB
{
    private $db = null;
    private $db_errno = 0;
    private $db_error = 0;
    private $dbHost = null;
    private $dbUser = null;
    private $dbPassword = null;
    private $dbName = null;
    private $msg = "";

    function __construct($dbHost, $dbUser, $dbPassword, $dbName) {
        $this->dbHost = $dbHost;
        $this->dbUser = $dbUser;
        $this->dbPassword = $dbPassword;
        $this->dbName = $dbName;
        $this->db_errno = 0;

        $this->osc_dbConnect();
    }

    function __destruct() {
        $this->osc_dbClose();
    }

    function debug($msg, $ok = true)
    {
        if( OSC_DEBUG_DB ) {
            $this->msg .= date("d/m/Y - H:i:s") . " ";

            if( $ok ) {
                $this->msg .= "<span style='background-color: #D0F5A9;' >[ OPERATION OK ]
                ";
            } else {
                $this->msg .= "<span style='background-color: #F5A9A9;' >[ OPERATION
                FAILED ] ";
            }

            $this->msg .= str_replace("\n", " ", $msg);
            $this->msg .= '</span><br />';
            $this->msg .= "\n";
        }
    }

    function print_debug()
    {
        if( OSC_DEBUG_DB && !defined('IS AJAX') ) {
            echo $this->msg;
        }
    }
}

/**

```



```

* Establish a connection to the MySQL database.
*
* @param string server ip or name
* @param string database user
* @param string database password
* @param string databasename
*/
function osc_dbConnect() {
    $this->db = @new mysqli($this->dbHost, $this->dbUser, $this->dbPassword, $this->
    dbName);
    if ($this->db->connect_error) {
        if( !defined('OSC_INSTALLING') ) {
            require_once LIB_PATH . 'osclass/helpers/hErrors.php';
            $title = 'Osclass &raquo; Error connecting to database';
            $message = 'Cannot connect to database. Check your configuration in <code>
            config.php</code> file.';
            osc_die($title, $message);
        }
        $this->debug('Error connecting to \'' . $this->dbName . '\', (' . $this->db->
        connect_errno . ': ' . $this->db->connect_error . ')', false);
    }

    $this->db_erro = $this->db->connect_errno;
    $this->debug('Connected to \'' . $this->dbName . '\': [DBHOST] = ' . $this->dbHost
    . ' | [DBUSER] = ' . $this->dbUser);
    $this->db->set_charset('utf8');
}

/**
* Close the database connection.
*/
function osc_dbClose() {
    if (!@$this->db->close()) {
        $this->debug('Error releasing the connection to \'' . $this->dbName . '\',
        false);
    }

    $this->debug('Connection with \'' . $this->dbName . '\', released properly');
    $this->print_debug();
}

/**
* Executes a SQL statement in the database.
*/
function osc_dbExec()
{
    $sql = null;
    $argv = func_get_args();
    switch(func_num_args()) {
        case 0: return; break;
        case 1: $sql = $argv[0]; break;
        default:
            $format = array_shift($argv);
            foreach($argv as &$arg)
                $arg = $this->db->real_escape_string($arg);
            unset($arg);

            $sql = vsprintf($format, $argv);
            break;
    }

    $result = $this->db->query($sql);
    if(!$result) {
        $this->debug($sql . ' | ' . $this->db->error . ' (' . $this->db->errno . ')',
        false);
    } else {
        $this->debug($sql);
    }

    $this->db_erro = $this->db->errno;
    return $result;
}

function osc_dbFetchValue() {
    $result = null;

    $sql = null;
    $argv = func_get_args();
    switch(func_num_args()) {
        case 0: return $result; break;
        case 1: $sql = $argv[0]; break;
        default:
            $format = array_shift($argv);
            $sql = vsprintf($format, $argv);
            break;
    }

    if($qry = $this->db->query($sql)) {
        $this->debug($sql);
        $row = $qry->fetch_array();
        $result = $row[0];
        $qry->free();
    } else {
        $this->debug($sql . ' | ' . $this->db->error . ' (' . $this->db->errno . ')',
        false);
    }

    $this->db_erro = $this->db->errno;
    return $result;
}

```

```

/**
 * @return array with values resulting of execution of query passed by parameter.
 */
function osc_dbFetchValues() {
    $results = array();

    $sql = null;
    $argv = func_get_args();
    switch(func_num_args()) {
        case 0: return $results; break;
        case 1: $sql = $argv[0]; break;
        default:
            $format = array_shift($argv);
            $sql = vsprintf($format, $argv);
            break;
    }

    if($qry = $this->db->query($sql)) {
        $this->debug($sql);
        while($result = $qry->fetch_array())
            $results[] = $result[0];
        $qry->free();
    } else {
        $this->debug($sql . ' | ' . $this->db->error . ' (' . $this->db->errno . ')',
            false);
    }
    $this->db->errno = $this->db->errno;
    return $results;
}

function osc_dbFetchResult() {
    $result = null;

    $sql = null;
    $argv = func_get_args();
    switch(func_num_args()) {
        case 0: return $results; break;
        case 1: $sql = $argv[0]; break;
        default:
            $format = array_shift($argv);
            $sql = vsprintf($format, $argv);
            break;
    }

    $qry = $this->db->query($sql);
    if($qry) {
        $this->debug($sql);
        $result = $qry->fetch_assoc();
        $qry->free();
    } else {
        $this->debug($sql . ' | ' . $this->db->error . ' (' . $this->db->errno . ')',
            false);
    }
    $this->db->errno = $this->db->errno;
    return $result;
}

function osc_dbFetchResults() {
    $results = array();

    $sql = null;
    $argv = func_get_args();
    switch(func_num_args()) {
        case 0: return $results; break;
        case 1: $sql = $argv[0]; break;
        default:
            $format = array_shift($argv);
            $sql = vsprintf($format, $argv);
            break;
    }

    if($qry = $this->db->query($sql)) {
        $this->debug($sql);
        while($result = $qry->fetch_assoc())
            $results[] = $result;
        $qry->free();
    } else {
        $this->debug($sql . ' | ' . $this->db->error . ' (' . $this->db->errno . ')',
            false);
    }
    $this->db->errno = $this->db->errno;
    return $results;
}

/**
 * Import (executes) the SQL passed as parameter making some proper adaptations.
 */
function osc_dbImportSQL($sql, $needle = '')
{
    $sql = str_replace('/*TABLE_PREFIX*/', DB_TABLE_PREFIX, $sql);
    $sentences = explode($needle . ';', $sql);
    // PREPARE THE QUERIES
    $var_l = count($sentences);
    $s_temp = '';
    for($var_k=0;$var_k<$var_l;$var_k++) {
        $s = $s_temp.$sentences[$var_k];
        if(!empty($s) && trim($s)!='') {
            $s .= $needle;
        }
    }
}

```

```

        $simple_comma = substr_count($s, ",");
        $escaped_simple_comma = substr_count($s, "\\,");
        if (($simple_comma - $escaped_simple_comma) % 2 == 0) {
            $sentences[$var_k] = $s;
            $s_temp = ',';
            //echo "[OK] ". $s. " <br />";
        } else {
            unset($sentences[$var_k]);
            $s_temp = $s. ",";
            //echo "[FAIL] ". $s. " <br />";
        }
    } else {
        unset($sentences[$var_k]);
    }
}

foreach($sentences as $s) {
    $s = trim($s);
    if( !empty($s) ) {
        $s = trim($s); // . $needle;
        if( $this->db->query($s) ) {
            $this->debug($s);
        } else {
            $this->debug($s . ' | ' . $this->db->error . ' (' . $this->db->errno .
                ')', false);
        }
    }
}

$this->db->errno = $this->db->errno;

if ($this->db->errno != 0) return false;
return true;
}

function autocommit($b_value) {
    $this->db->autocommit($b_value);
}

function commit() {
    $this->db->commit();
}

function rollback() {
    $this->db->rollback();
}

function get_last_id() {
    return($this->db->insert_id);
}

function get_affected_rows() {
    return($this->db->affected_rows);
}

function get_errno() {
    return($this->db->errno);
}

/**
 * Given some queries, it will check against the installed database if the information
 * is the same
 *
 * @param mixed array or string with the SQL queries.
 * @return BOOLEAN true on success, false on fail
 */
function osc_updateDB($queries = '') {
    if(!is_array($queries)) {
        $queries = explode(";", $queries);
    }

    // Prepare and separate the queries
    $struct_queries = array();
    $data_queries = array();
    foreach($queries as $query) {
        if(preg_match('|CREATE DATABASE ([^ ]*)|', $query, $match)) {
            array_unshift($struct_queries, $query);
        } else if(preg_match('|CREATE TABLE ([^ ]*)|', $query, $match)) {
            $struct_queries[trim(strtolower($match[1]), ' ')] = $query;
        } else if(preg_match('|INSERT INTO ([^ ]*)|', $query, $match)) {
            $data_queries[] = $query;
        } else if(preg_match('|UPDATE ([^ ]*)|', $query, $match)) {
            $data_queries[] = $query;
        }
    }

    // Get tables from DB (already installed)
    $tables = $this->osc_dbFetchResults('SHOW TABLES');
    foreach($tables as $v) {
        $table = current($v);
        if(array_key_exists(strtolower($table), $struct_queries)) {
            // Get the fields from the query
            if(preg_match('|\\((.*)\\)|ms', $struct_queries[strtolower($table)], $match)) {
                $fields = explode("\n", trim($match[1]));
            }
            // Detect if it's a "normal field definition" or a index one
            $normal_fields = $indexes = array();

```



```

        } else {
            $var_index = array_search(str_replace(' ', ' ', $string),
                $indexes);
            if (!$var_index) {
                unset($indexes[$var_index]);
            }
        }
    }
}
// For the rest of the indexes (they are in the new definition but not
// in the table installed
foreach($indexes as $index) {
    if(strtolower(substr(trim($index),0,2))!='on') { // && strtolower(
        substr(trim($index),0,7))!='foreign' ) {
        $struct_queries [] = "ALTER TABLE ".$table." ADD ".$index;
    } else {
        // $struct_queries [] = "ALTER TABLE ".$table." ".$index;
    }
}
// No need to create the table, so we delete it SQL
unset($struct_queries[strtolower($table)]);
}
}
}

$queries = array_merge($struct_queries, $data_queries);
$ok = true;
$error_queries = array();
foreach($queries as $query) {
    $res = $this->osc_dbExec($query);
    if (!$res) {
        $ok = false;
        $error_queries [] = $query;
    }
}

return array($ok, $queries, $error_queries);
}

}

function getConnection($dbHost = null, $dbUser = null, $dbPassword = null, $dbName = null,
    $dbLogLevel = null)
{
    static $instance;

    if (defined('DB_HOST') && $dbHost == null) $dbHost = osc_db_host();
    if (defined('DB_USER') && $dbUser == null) $dbUser = osc_db_user();
    if (defined('DB_PASSWORD') && $dbPassword == null) $dbPassword =
        osc_db_password();
    if (defined('DB_NAME') && $dbName == null) $dbName = osc_db_name();

    if (!isset($instance[$dbName . "-" . $dbHost])) {
        if (!isset($instance)) {
            $instance = array();
        }

        $instance[$dbName . "-" . $dbHost] = new DB($dbHost, $dbUser, $dbPassword, $dbName
        );
    }

    return ($instance[$dbName . "-" . $dbHost]);
}

?>

<?php

/*
 *   Osclass   software for creating and publishing online classified
 *             advertising platforms
 *
 *   Copyright (C) 2012 OSCLASS
 *
 *   This program is free software: you can redistribute it and/or
 *   modify it under the terms of the GNU Affero General Public License
 *   as published by the Free Software Foundation, either version 3 of
 *   the License, or (at your option) any later version.
 *
 *   This program is distributed in the hope that it will be useful, but
 *   WITHOUT ANY WARRANTY; without even the implied warranty of
 *   MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 *   GNU Affero General Public License for more details.
 *
 *   You should have received a copy of the GNU Affero General Public
 *   License along with this program. If not, see <http://www.gnu.org/licenses/>.
 */

if( !defined('MULTISITE') ) {
    define('MULTISITE', 0);
}

if( !defined('OC_ADMIN') ) {
    define('OC_ADMIN', false);
}

if( !defined('LIB_PATH') ) {
    define('LIB_PATH', ABS_PATH . 'oc-includes/');
}

```

```

if( !defined('CONTENT_PATH') ) {
    define('CONTENT_PATH', ABS_PATH . 'oc-content/');
}

if( !defined('THEMES_PATH') ) {
    define('THEMES_PATH', CONTENT_PATH . 'themes/');
}

if( !defined('PLUGINS_PATH') ) {
    define('PLUGINS_PATH', CONTENT_PATH . 'plugins/');
}

if( !defined('TRANSLATIONS_PATH') ) {
    define('TRANSLATIONS_PATH', CONTENT_PATH . 'languages/');
}

if( !defined('UPLOADS_PATH') ) {
    define('UPLOADS_PATH', CONTENT_PATH . 'uploads/');
}

if( !defined('OSC_DEBUG_DB') ) {
    define('OSC_DEBUG_DB', false);
}

if( !defined('OSC_DEBUG_DB_LOG') ) {
    define('OSC_DEBUG_DB_LOG', false);
}

if( !defined('OSC_DEBUG_DB_EXPLAIN') ) {
    define('OSC_DEBUG_DB_EXPLAIN', false);
}

if( !defined('OSC_DEBUG') ) {
    define('OSC_DEBUG', false);
}

if( !defined('OSC_DEBUG_LOG') ) {
    define('OSC_DEBUG_LOG', false);
}

if( !defined('OSC_MEMORY_LIMIT') ) {
    define('OSC_MEMORY_LIMIT', '32M');
}

if( function_exists('memory_get_usage') && ( (int) @ini_get('memory_limit') < abs(intval(
    OSC_MEMORY_LIMIT)) ) ) {
    @ini_set('memory_limit', OSC_MEMORY_LIMIT);
}

if( !defined('CLI') ) {
    define('CLI', false);
}
?>

<?php if ( ! defined('ABS_PATH')) exit('ABS_PATH is not loaded. Direct access is not allowed
. ');

/**
 * Osclass software for creating and publishing online classified advertising
 * platforms
 *
 * Copyright (C) 2012 OSCLASS
 *
 * This program is free software: you can redistribute it and/or modify it under the terms
 * of the GNU Affero General Public License as published by the Free Software Foundation,
 * either version 3 of the License, or (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful, but WITHOUT ANY
 * WARRANTY;
 * without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
 * PURPOSE.
 * See the GNU Affero General Public License for more details.
 *
 * You should have received a copy of the GNU Affero General Public
 * License along with this program. If not, see <http://www.gnu.org/licenses/>.
 */

function fn_email_alert_validation($alert, $email, $secret) {
    $user['s_name'] = "";

    // send alert validation email
    $prefLocale = osc_language();
    $page = Page::newInstance()->findByName('email_alert_validation');
    $page_description = $page['locale'];

    $s_title = osc_apply_filter('email_title', osc_apply_filter('
        email_alert_validation_title', $page_description[$prefLocale]['s_title']));
    $s_body = osc_apply_filter('email_description', osc_apply_filter('
        email_alert_validation_description', $page_description[$prefLocale]['s_text']));

    $validation_link = osc_user_activate_alert_url( $alert['pk_i_id'], $secret, $email );

    $words = array();
    $words[] = array(
        'USER_NAME',
        'USER_EMAIL',
        'VALIDATION_LINK'
    );
}

```

```

$words[] = array(
    $user['s_name'],
    $email,
    $validation_link
);
$title = osc_mailBeauty($_title, $words);
$body = osc_mailBeauty($_body, $words);

$params = array(
    'subject' => $title,
    'from' => osc_contact_email(),
    'to' => $email,
    'to_name' => $user['s_name'],
    'body' => $body,
    'alt_body' => $body
);

    osc_sendMail($params);
}
osc_add_hook('hook_email_alert_validation', 'fn_email_alert_validation');

function fn_alert_email_hourly($user, $ads, $s_search) {
    $prefLocale = osc_language();
    $page = Page::newInstance()->findByName('alert_email_hourly');
    $page_description = $page['locale'];

    $_title = osc_apply_filter('email_title', osc_apply_filter('alert_email_hourly_title',
        $page_description[$prefLocale]['s_title']));
    $_body = osc_apply_filter('email_description', osc_apply_filter('
        alert_email_hourly_description', $page_description[$prefLocale]['s_text']));

    if( $user['fk_i_user_id'] != 0 ) {
        $user = User::newInstance()->findByPrimaryKey($user['fk_i_user_id']);
    } else {
        $user['s_name'] = $user['s_email'];
    }

    $unsubscribe_link = osc_user_unsubscribe_alert_url($s_search['pk_i_id'], $user['s_email'],
        $s_search['s_secret']);
    $unsubscribe_link = '<a href="' . $unsubscribe_link . '">' . __('unsubscribe alert') . '</a>';

    $words = array();
    $words[] = array(
        '{USER_NAME}',
        '{USER_EMAIL}',
        '{ADS}',
        '{UNSUBLINK}'
    );
    $words[] = array(
        $user['s_name'],
        $user['s_email'],
        $ads,
        $unsubscribe_link
    );
    $title = osc_mailBeauty($_title, $words);
    $body = osc_mailBeauty($_body, $words);

    $params = array(
        'subject' => $title,
        'from' => osc_contact_email(),
        'to' => $user['s_email'],
        'to_name' => $user['s_name'],
        'body' => $body,
        'alt_body' => $body
    );

    osc_sendMail($params);
}
osc_add_hook('hook_alert_email_hourly', 'fn_alert_email_hourly');

function fn_alert_email_daily($user, $ads, $s_search) {
    $prefLocale = osc_language();
    $page = Page::newInstance()->findByName('alert_email_daily');
    $page_description = $page['locale'];

    $_title = osc_apply_filter('email_title', osc_apply_filter('alert_email_daily_title',
        $page_description[$prefLocale]['s_title']));
    $_body = osc_apply_filter('email_description', osc_apply_filter('
        alert_email_daily_description', $page_description[$prefLocale]['s_text']));

    if( $user['fk_i_user_id'] != 0 ) {
        $user = User::newInstance()->findByPrimaryKey($user['fk_i_user_id']);
    } else {
        $user['s_name'] = $user['s_email'];
    }

    $unsubscribe_link = osc_user_unsubscribe_alert_url($s_search['pk_i_id'], $user['s_email'],
        $s_search['s_secret']);
    $unsubscribe_link = '<a href="' . $unsubscribe_link . '">' . __('unsubscribe alert') . '</a>';

    $words = array();
    $words[] = array(
        '{USER_NAME}',
        '{USER_EMAIL}',
        '{ADS}',
        '{UNSUBLINK}'
    );
    $words[] = array(
        $user['s_name'],

```

```

        $user['s_email'],
        $ads,
        $unsub_link
    );
    $title = osc_mailBeauty($_title, $words);
    $body = osc_mailBeauty($_body, $words);

    $params = array(
        'subject' => $title,
        'from' => osc_contact_email(),
        'to' => $user['s_email'],
        'to_name' => $user['s_name'],
        'body' => $body,
        'alt_body' => $body
    );

    osc_sendMail($params);
}
osc_add_hook('hook_alert_email_daily', 'fn_alert_email_daily');

function fn_alert_email_weekly($user, $ads, $s_search) {
    $prefLocale = osc_language();
    $page = Page::newInstance()->findByInternalName('alert_email_weekly');
    $page_description = $page['locale'];

    $_title = osc_apply_filter('email_title', osc_apply_filter('alert_email_weekly_title',
        $page_description[$prefLocale]['s_title']));
    $_body = osc_apply_filter('email_description', osc_apply_filter('
        alert_email_weekly_description', $page_description[$prefLocale]['s_text']));

    if( $user['fk_i_user_id'] != 0 ) {
        $user = User::newInstance()->findByPrimaryKey($user['fk_i_user_id']);
    } else {
        $user['s_name'] = $user['s_email'];
    }

    $unsub_link = osc_user_unsubscribe_alert_url($s_search['pk_i_id'], $user['s_email'],
        $s_search['s_secret']);
    $unsub_link = '<a href="' . $unsub_link . '">' . __('unsubscribe alert') . '</a>';

    $words = array();
    $words[] = array(
        '{USER_NAME}',
        '{USER_EMAIL}',
        '{ADS}',
        '{UNSUB_LINK}'
    );
    $words[] = array(
        $user['s_name'],
        $user['s_email'],
        $ads,
        $unsub_link
    );
    $title = osc_mailBeauty($_title, $words);
    $body = osc_mailBeauty($_body, $words);

    $params = array(
        'subject' => $title,
        'from' => osc_contact_email(),
        'to' => $user['s_email'],
        'to_name' => $user['s_name'],
        'body' => $body,
        'alt_body' => $body
    );

    osc_sendMail($params);
}
osc_add_hook('hook_alert_email_weekly', 'fn_alert_email_weekly');

function fn_alert_email_instant($user, $ads, $s_search) {
    $prefLocale = osc_language();
    $page = Page::newInstance()->findByInternalName('alert_email_instant');
    $page_description = $page['locale'];

    $_title = osc_apply_filter('email_title', osc_apply_filter('alert_email_instant_title',
        $page_description[$prefLocale]['s_title']));
    $_body = osc_apply_filter('email_description', osc_apply_filter('
        alert_email_instant_description', $page_description[$prefLocale]['s_text']));

    if( $user['fk_i_user_id'] != 0 ) {
        $user = User::newInstance()->findByPrimaryKey($user['fk_i_user_id']);
    } else {
        $user['s_name'] = $user['s_email'];
    }

    $unsub_link = osc_user_unsubscribe_alert_url($s_search['pk_i_id'], $user['s_email'],
        $s_search['s_secret']);
    $unsub_link = '<a href="' . $unsub_link . '">' . __('unsubscribe alert') . '</a>';

    $words = array();
    $words[] = array(
        '{USER_NAME}',
        '{USER_EMAIL}',
        '{ADS}',
        '{UNSUB_LINK}'
    );
    $words[] = array(
        $user['s_name'],
        $user['s_email'],

```



```

        $ads,
        $unsubscribe
    );
    $title = osc_mailBeauty($_title, $words);
    $body = osc_mailBeauty($_body, $words);

    $params = array(
        'subject' => $title,
        'from' => osc_contact_email(),
        'to' => $user['s_email'],
        'to_name' => $user['s_name'],
        'body' => $body,
        'alt_body' => $body
    );

    osc_sendMail($params);
}
osc_add_hook('hook_alert_email_instant', 'fn_alert_email_instant');

function fn_email_comment_validated($aComment) {
    $mPages = new Page();
    $locale = osc_current_user_locale();
    $aPage = $mPages->findByInternalName('email_comment_validated');

    $content = array();
    if(isset($aPage['locale'][$locale]['s_title'])) {
        $content = $aPage['locale'][$locale];
    } else {
        $content = current($aPage['locale']);
    }

    if(!is_null($content)) {
        $words = array();
        $words[] = array(
            '{COMMENT.AUTHOR}',
            '{COMMENT.EMAIL}',
            '{COMMENT.TITLE}',
            '{COMMENT.BODY}',
            '{ITEM.URL}',
            '{ITEM.LINK}',
            '{ITEM.TITLE}'
        );

        $words[] = array(
            $aComment['s_author_name'],
            $aComment['s_author_email'],
            $aComment['s_title'],
            $aComment['s_body'],
            osc_item_url(),
            '<a href="'. osc_item_url() . '>' . osc_item_url() . '</a>',
            osc_item_title()
        );

        $title = osc_mailBeauty(osc_apply_filter('email_title', osc_apply_filter('
            email_comment_validated_title', $content['s_title'])), $words);
        $body = osc_mailBeauty(osc_apply_filter('email_description', osc_apply_filter('
            email_comment_validated_description', $content['s_text'])), $words);

        $emailParams = array(
            'subject' => $title,
            'from' => osc_contact_email(),
            'to' => $aComment['s_author_email'],
            'to_name' => $aComment['s_author_name'],
            'body' => $body,
            'alt_body' => $body
        );

        osc_sendMail($emailParams);
    }
}
osc_add_hook('hook_email_comment_validated', 'fn_email_comment_validated');

function fn_email_new_item_non_register_user($item) {
    $mPages = new Page();
    $aPage = $mPages->findByInternalName('email_new_item_non_register_user');
    $locale = osc_current_user_locale();

    $content = array();
    if(isset($aPage['locale'][$locale]['s_title'])) {
        $content = $aPage['locale'][$locale];
    } else {
        $content = current($aPage['locale']);
    }

    $item_url = osc_item_url();
    $item_url = '<a href="'. $item_url . '>' . $item_url . '</a>';
    $edit_url = osc_item_edit_url($item['s_secret'], $item['pk_i_id']);
    $delete_url = osc_item_delete_url($item['s_secret'], $item['pk_i_id']);

    $words = array();
    $words[] = array(
        '{ITEM.ID}',
        '{USER.NAME}',
        '{USER.EMAIL}',
        '{ITEM.TITLE}',
        '{ITEM.URL}',
        '{ITEM.LINK}',
        '{EDIT.LINK}',
        '{EDIT.URL}',
        '{DELETE.LINK}',
        '{DELETE.URL}'
    );
}

```

```

$words[] = array(
    $item['pk.i.id'],
    $item['s.contact.name'],
    $item['s.contact.email'],
    $item['s.title'],
    osc_item_url(),
    $item_url,
    '<a href="' . $edit_url . '>' . $edit_url . '</a>',
    $edit_url,
    '<a href="' . $delete_url . '>' . $delete_url . '</a>',
    $delete_url
);
$title = osc_mailBeauty(osc_apply_filter('email.title', osc_apply_filter('
    email_new_item_non_register_user_title', $content['s.title'])), $words);
$body = osc_mailBeauty(osc_apply_filter('email.description', osc_apply_filter('
    email_new_item_non_register_user_description', $content['s.text'])), $words);

$emailParams = array(
    'subject' => $title,
    'from' => osc_contact_email(),
    'to' => $item['s.contact.email'],
    'to_name' => $item['s.contact.name'],
    'body' => $body,
    'alt_body' => $body
);

    osc_sendMail($emailParams);
}
osc_add_hook('hook_email_new_item_non_register_user', 'fn_email_new_item_non_register_user
');

function fn_email_user_forgot_password($user, $password_url) {
    $aPage = Page::newInstance()->findByInternalName('email_user_forgot_password');
    $locale = osc_current_user_locale();

    $content = array();
    if (isset($aPage['locale'][$locale]['s.title'])) {
        $content = $aPage['locale'][$locale];
    } else {
        $content = current($aPage['locale']);
    }

    if (!is_null($content)) {
        $words = array();
        $words[] = array(
            '{USER_NAME}',
            '{USER_EMAIL}',
            '{PASSWORD_LINK}',
            '{PASSWORD_URL}',
            '{DATE.TIME}'
        );
        $words[] = array(
            $user['s.name'],
            $user['s.email'],
            '<a href="' . $password_url . '>' . $password_url . '</a>',
            $password_url,
            date('Y-m-d H:i:') . '00'
        );
        $title = osc_mailBeauty(osc_apply_filter('email.title', osc_apply_filter('
            email_user_forgot_password_title', $content['s.title'])), $words);
        $body = osc_mailBeauty(osc_apply_filter('email.description', osc_apply_filter('
            email_user_forgot_password_description', $content['s.text'])), $words);

        $emailParams = array(
            'subject' => $title,
            'from' => osc_contact_email(),
            'to' => $user['s.email'],
            'to_name' => $user['s.name'],
            'body' => $body,
            'alt_body' => $body
        );

        osc_sendMail($emailParams);
    }
}
osc_add_hook('hook_email_user_forgot_password', 'fn_email_user_forgot_password');

function fn_email_user_registration($user) {
    $pageManager = new Page();
    $locale = osc_current_user_locale();
    $aPage = $pageManager->findByInternalName('email_user_registration');

    $content = array();
    if (isset($aPage['locale'][$locale]['s.title'])) {
        $content = $aPage['locale'][$locale];
    } else {
        $content = current($aPage['locale']);
    }

    if (!is_null($content)) {
        $words = array();
        $words[] = array(
            '{USER_NAME}',
            '{USER_EMAIL}'
        );
        $words[] = array(
            $user['s.name'],
            $user['s.email']
        );
    }
}

```

```

$title = osc_mailBeauty(osc_apply_filter('email_title', osc_apply_filter('
email_user_registration_title', $content['s_title'])), $words);
$body = osc_mailBeauty(osc_apply_filter('email_description', osc_apply_filter('
email_user_registration_description', $content['s_text'])), $words);

$emailParams = array(
    'subject' => $title,
    'from' => osc_contact_email(),
    'to' => $user['s_email'],
    'to_name' => $user['s_name'],
    'body' => $body,
    'alt_body' => $body
);

    osc_sendMail($emailParams);
}
}
osc_add_hook('hook_email_user_registration', 'fn_email_user_registration');

function fn_email_new_email($new_email, $validation_url) {
    $locale = osc_current_user_locale();
    $aPage = Page::newInstance()->findByInternalName('email_new_email');

    $content = array();
    if (isset($aPage['locale'][$locale]['s_title'])) {
        $content = $aPage['locale'][$locale];
    } else {
        $content = current($aPage['locale']);
    }

    if (!is_null($content)) {
        $words = array();
        $words[] = array(
            '{USER_NAME}',
            '{USER_EMAIL}',
            '{VALIDATION_LINK}',
            '{VALIDATION_URL}'
        );
        $words[] = array(
            Session::newInstance()->get('userName'),
            Params::getParam('new_email'),
            '<a href="' . $validation_url . '" >' . $validation_url . '</a>',
            $validation_url
        );
    };
    $title = osc_mailBeauty(osc_apply_filter('email_title', osc_apply_filter('
email_new_email_title', $content['s_title'])), $words);
    $body = osc_mailBeauty(osc_apply_filter('email_description', osc_apply_filter('
email_new_email_description', $content['s_text'])), $words);

    $params = array(
        'subject' => $title,
        'from' => osc_contact_email(),
        'to' => $new_email,
        'to_name' => Session::newInstance()->get('userName'),
        'body' => $body,
        'alt_body' => $body
    );
    osc_sendMail($params);
    osc_add_flash_ok_message(_m("We've sent you an e-mail. Follow its instructions to
    validate the changes"));
} else {
    osc_add_flash_error_message(_m("We tried to sent you an e-mail, but it failed.
    Please, contact an administrator"));
}
}
osc_add_hook('hook_email_new_email', 'fn_email_new_email');

function fn_email_user_validation($user, $input) {
    $mPages = new Page();
    $locale = osc_current_user_locale();
    $aPage = $mPages->findByInternalName('email_user_validation');

    $content = array();
    if (isset($aPage['locale'][$locale]['s_title'])) {
        $content = $aPage['locale'][$locale];
    } else {
        $content = current($aPage['locale']);
    }

    if (!is_null($content)) {
        $validation_url = osc_user_activate_url($user['pk_id'], $input['s_secret']);
        $words = array();
        $words[] = array(
            '{USER_NAME}',
            '{USER_EMAIL}',
            '{VALIDATION_LINK}',
            '{VALIDATION_URL}'
        );
        $words[] = array(
            $user['s_name'],
            $user['s_email'],
            '<a href="' . $validation_url . '" >' . $validation_url . '</a>',
            $validation_url
        );
    };
    $title = osc_mailBeauty(osc_apply_filter('email_title', osc_apply_filter('
email_user_validation_title', $content['s_title'])), $words);
    $body = osc_mailBeauty(osc_apply_filter('email_description', osc_apply_filter('
email_user_validation_description', $content['s_text'])), $words);

```

```

        $emailParams = array(
            'subject' => $title,
            'from' => osc_contact_email(),
            'to' => $user['s_email'],
            'to_name' => $user['s_name'],
            'body' => $body,
            'alt_body' => $body
        );
        osc_sendMail($emailParams);
    }
}
osc_add_hook('hook_email_user_validation', 'fn_email_user_validation');

function fn_email_send_friend($aItem) {
    $mPages = new Page();
    $aPage = $mPages->findByInternalName('email_send_friend');
    $locale = osc_current_user_locale();

    $content = array();
    if (isset($aPage['locale'][$locale]['s_title'])) {
        $content = $aPage['locale'][$locale];
    } else {
        $content = current($aPage['locale']);
    }

    $item_url = osc_item_url();
    $item_url = '<a href="'. $item_url .'" >'. $item_url . '</a>';

    $words = array();
    $words[] = array(
        '{FRIEND_NAME}',
        '{USER_NAME}',
        '{USER_EMAIL}',
        '{FRIEND_EMAIL}',
        '{ITEM_TITLE}',
        '{COMMENT}',
        '{ITEM_URL}',
        '{ITEM_LINK}'
    );
    $words[] = array(
        $aItem['friendName'],
        $aItem['yourName'],
        $aItem['yourEmail'],
        $aItem['friendEmail'],
        $aItem['s_title'],
        $aItem['message'],
        osc_item_url(),
        $item_url
    );
};

$title = osc_mailBeauty(osc_apply_filter('email_title', osc_apply_filter('
    email_send_friend_title', $content['s_title'])), $words);
$body = osc_mailBeauty(osc_apply_filter('email_description', osc_apply_filter('
    email_send_friend_description', $content['s_text'])), $words);

$params = array(
    'from' => osc_contact_email(),
    'from_name' => osc_page_title(),
    'reply_to' => $aItem['yourEmail'],
    'subject' => $title,
    'to' => $aItem['friendEmail'],
    'to_name' => $aItem['friendName'],
    'body' => $body
);

if ( osc_notify_contact_friends() ) {
    $params['add_bcc'] = osc_contact_email();
}

    osc_sendMail($params);
}
osc_add_hook('hook_email_send_friend', 'fn_email_send_friend');

function fn_email_item_inquiry($aItem) {
    $id = $aItem['id'];
    $yourEmail = $aItem['yourEmail'];
    $yourName = $aItem['yourName'];
    $phoneNumber = $aItem['phoneNumber'];
    // contact - strip_tags + nl2br
    $message = nl2br( strip_tags( $aItem['message'] ) );

    $path = null;
    $item = Item::newInstance()->findByPrimaryKey( $id );
    View::newInstance()->exportVariableToView('item', $item);

    $mPages = new Page();
    $aPage = $mPages->findByInternalName('email_item_inquiry');
    $locale = osc_current_user_locale();

    $content = array();
    if (isset($aPage['locale'][$locale]['s_title'])) {
        $content = $aPage['locale'][$locale];
    } else {
        $content = current($aPage['locale']);
    }

    $item_url = osc_item_url();
    $item_url = '<a href="'. $item_url .'" >'. $item_url . '</a>';
}

```

```

$words = array();
$words[] = array(
    '{CONTACT_NAME}',
    '{USER_NAME}',
    '{USER_EMAIL}',
    '{USER_PHONE}',
    '{ITEM_TITLE}',
    '{ITEM_URL}',
    '{ITEM_LINK}',
    '{COMMENT}'
);

$words[] = array(
    $item['s_contact_name'],
    $yourName,
    $yourEmail,
    $phoneNumber,
    $item['s_title'],
    osc_item_url(),
    $item_url,
    $message
);

$title = osc_mailBeauty(osc_apply_filter('email_title', osc_apply_filter('
email_item_inquiry_title', $content['s_title'])), $words);
$body = osc_mailBeauty(osc_apply_filter('email_description', osc_apply_filter('
email_item_inquiry_description', $content['s_text'])), $words);

$from = osc_contact_email();
$from_name = osc_page_title();

$emailParams = array(
    'from' => $from,
    'from_name' => $from_name,
    'subject' => $title,
    'to' => $item['s_contact_email'],
    'to_name' => $item['s_contact_name'],
    'body' => $body,
    'alt_body' => $body,
    'reply_to' => $yourEmail
);

if( osc_notify_contact_item() ) {
    $emailParams['add_bcc'] = osc_contact_email();
}

if( osc_item_attachment() ) {
    $attachment = Params::getFiles('attachment');
    $resourceName = $attachment['name'];
    $tmpName = $attachment['tmp_name'];
    $path = osc_uploads_path() . time() . '_' . $resourceName;

    if( !is_writable(osc_uploads_path()) ) {
        osc_add_flash_error_message( _m('There has been some errors sending the
message') );
    }

    if( !move_uploaded_file($tmpName, $path) ) {
        unset($path);
    }
}

if( isset($path) ) {
    $emailParams['attachment'] = $path;
}

osc_sendMail($emailParams);

@unlink($path);
}
osc_add_hook('hook_email_item_inquiry', 'fn_email_item_inquiry');

function fn_email_new_comment_admin($aItem) {
    $authorName = trim($aItem['authorName']);
    $authorName = strip_tags($authorName);
    $authorEmail = trim($aItem['authorEmail']);
    $authorEmail = strip_tags($authorEmail);
    $body = trim($aItem['body']);
    // only \n -> <br/>
    $body = nl2br(strip_tags($body));
    $title = $aItem['title'];
    $itemId = $aItem['id'];
    $admin_email = osc_contact_email();

    $item = Item::newInstance()->findByPrimaryKey($itemId);
    View::newInstance()->exportVariableToView('item', $item);
    $itemURL = osc_item_url();
    $itemURL = '<a href="' . $itemURL . '" >' . $itemURL . '</a>';

    $mPages = new Page();
    $aPage = $mPages->findByPrimaryKey('email_new_comment_admin');
    $locale = osc_current_user_locale();

    $content = array();
    if( isset($aPage['locale'][$locale]['s_title']) ) {
        $content = $aPage['locale'][$locale];
    } else {
        $content = current($aPage['locale']);
    }
}

```

```

$words = array();
$words[] = array(
    '{COMMENT.AUTHOR}',
    '{COMMENT.EMAIL}',
    '{COMMENT.TITLE}',
    '{COMMENT.TEXT}',
    '{ITEM.TITLE}',
    '{ITEM.ID}',
    '{ITEM.URL}',
    '{ITEM.LINK}'
);
$words[] = array(
    $authorName,
    $authorEmail,
    $title,
    $body,
    $item['s_title'],
    $itemId,
    osc_item_url(),
    $itemURL
);
$title_email = osc_mailBeauty(osc_apply_filter('email_title', osc_apply_filter('
email_new_comment_admin_title', $content['s_title'])), $words);
$body_email = osc_mailBeauty(osc_apply_filter('email_description', osc_apply_filter('
email_new_comment_admin_description', $content['s_text'])), $words);

$emailParams = array(
    'from' => osc_contact_email(),
    'to' => $admin_email,
    'to_name' => --('Admin mail system'),
    'subject' => $title_email,
    'body' => $body_email,
    'alt_body' => $body_email
);
osc_sendMail($emailParams);
}
osc_add_hook('hook_email_new_comment_admin', 'fn_email_new_comment_admin');

function fn_email_item_validation($item) {
    View::newInstance()->exportVariableToView('item', $item);
    $title = osc_item_title();
    $contactEmail = $item['s_contact_email'];
    $contactName = $item['s_contact_name'];
    $mPages = new Page();
    $locale = osc_current_user_locale();
    $aPage = $mPages->findByInternalName('email_item_validation');

    $content = array();
    if (isset($aPage['locale'][$locale]['s_title'])) {
        $content = $aPage['locale'][$locale];
    } else {
        $content = current($aPage['locale']);
    }

    $item_url = osc_item_url();
    $item_url = '<a href="'. $item_url .'" >'. $item_url . '</a>';

    $all = '';

    if (isset($item['locale'])) {
        foreach ($item['locale'] as $locale => $data) {
            $locale_name = OSCLocale::newInstance()->findByCode($locale);
            $all .= '<br/>';
            if (isset($locale_name[0] ['s_name'])) {
                $all .= --('Language') . ': ' . $locale_name[0] ['s_name'] . '<br/>';
            } else {
                $all .= --('Language') . ': ' . $locale . '<br/>';
            }
            $all .= --('Title') . ': ' . $data['s_title'] . '<br/>';
            $all .= --('Description') . ': ' . $data['s_description'] . '<br/>';
            $all .= '<br/>';
        }
    } else {
        $all .= --('Title') . ': ' . $item['s_title'] . '<br/>';
        $all .= --('Description') . ': ' . $item['s_description'] . '<br/>';
    }

    // Format activation URL
    $validation_url = osc_item_activate_url($item['s_secret'], $item['pk-i-id']);

    $words = array();
    $words[] = array(
        '{ITEM.DESCRPTION.ALLLANGUAGES}',
        '{ITEM.DESCRPTION}',
        '{ITEM.COUNTRY}',
        '{ITEM.PRICE}',
        '{ITEM.REGION}',
        '{ITEM.CITY}',
        '{ITEM.ID}',
        '{USER.NAME}',
        '{USER.EMAIL}',
        '{ITEM.TITLE}',
        '{ITEM.URL}',
        '{VALIDATION.LINK}',
        '{VALIDATION.URL}'
    );
    $words[] = array(
        $all,

```

```

        $item['s_description'],
        $item['s_country'],
        osc_format_price($item['i_price']),
        $item['s_region'],
        $item['s_city'],
        $item['pk_i_id'],
        $item['s_contact_name'],
        $item['s_contact_email'],
        $item['s_title'],
        $item_url,
        '<a href="' . $validation_url . '" >' . $validation_url . '</a>',
        $validation_url
    );
    $title = osc_mailBeauty(osc_apply_filter('email_title', osc_apply_filter('
        email_item_validation_title', $content['s_title'])), $words);
    $body = osc_mailBeauty(osc_apply_filter('email_description', osc_apply_filter('
        email_item_validation_description', $content['s_text'])), $words);

    $emailParams = array (
        'subject' => $title,
        'from' => osc_contact_email(),
        'to' => $contactEmail,
        'to_name' => $contactName,
        'body' => $body,
        'alt_body' => $body
    );
    osc_sendMail($emailParams);
}
osc_add_hook('hook_email_item_validation', 'fn_email_item_validation');

function fn_email_admin_new_item($item) {
    View::newInstance()->exportVariableToView('item', $item);
    $title = osc_item_title();
    $mPages = new Page();
    $locale = osc_current_user_locale();
    $aPage = $mPages->findByInternalName('email_admin_new_item');

    $content = array();
    if (isset($aPage['locale'][$locale]['s_title'])) {
        $content = $aPage['locale'][$locale];
    } else {
        $content = current($aPage['locale']);
    }

    $item_url = osc_item_url();
    $item_url = '<a href="' . $item_url . '" >' . $item_url . '</a>';

    $all = '';

    if (isset($item['locale'])) {
        foreach ($item['locale'] as $locale => $data) {
            $locale_name = OSCLocale::newInstance()->findByCode($locale);
            $all .= '<br/>';
            if (isset($locale_name[0] && isset($locale_name[0]['s_name']))) {
                $all .= --('Language') . ': ' . $locale_name[0]['s_name'] . '<br/>';
            } else {
                $all .= --('Language') . ': ' . $locale . '<br/>';
            }
            $all .= --('Title') . ': ' . $data['s_title'] . '<br/>';
            $all .= --('Description') . ': ' . $data['s_description'] . '<br/>';
            $all .= '<br/>';
        }
    } else {
        $all .= --('Title') . ': ' . $item['s_title'] . '<br/>';
        $all .= --('Description') . ': ' . $item['s_description'] . '<br/>';
    }

    // Format activation URL
    $validation_url = osc_item_activate_url($item['s_secret'], $item['pk_i_id']);

    // Format admin edit URL
    $admin_edit_url = osc_item_admin_edit_url($item['pk_i_id']);

    $words = array();
    $words[] = array(
        '{EDIT_LINK}',
        '{EDIT_URL}',
        '{ITEM_DESCRIPTION_ALLLANGUAGES}',
        '{ITEM_DESCRIPTION}',
        '{ITEM_COUNTRY}',
        '{ITEM_PRICE}',
        '{ITEM_REGION}',
        '{ITEM_CITY}',
        '{ITEM_ID}',
        '{USER_NAME}',
        '{USER_EMAIL}',
        '{ITEM_TITLE}',
        '{ITEM_URL}',
        '{VALIDATION_LINK}',
        '{VALIDATION_URL}'
    );
    $words[] = array(
        '<a href="' . $admin_edit_url . '" >' . $admin_edit_url . '</a>',
        $admin_edit_url,
        $all,
        $item['s_description'],
        $item['s_country'],
        osc_format_price($item['i_price']),
        $item['s_region'],

```

```

        $item['s_city'],
        $item['pk_i_id'],
        $item['s_contact_name'],
        $item['s_contact_email'],
        $item['s_title'],
        $item_url,
        '<a href="' . $validation_url . '" >' . $validation_url . '</a>',
        $validation_url
    );
    $title = osc_mailBeauty(osc_apply_filter('email_title', osc_apply_filter('
        email_admin_new_item_title', $content['s_title'])), $words);
    $body = osc_mailBeauty(osc_apply_filter('email_description', osc_apply_filter('
        email_admin_new_item_description', $content['s_text'])), $words);

    $emailParams = array(
        'subject' => $title,
        'from' => osc_contact_email(),
        'to' => osc_contact_email(),
        'to_name' => 'admin',
        'body' => $body,
        'alt_body' => $body
    );
    osc_sendMail($emailParams);
}
osc_add_hook('hook_email_admin_new_item', 'fn_email_admin_new_item');

function fn_email_item_validation_non_register_user($item) {
    View::newInstance()->exportVariableToView('item', $item);

    $mPages = new Page();
    $aPage = $mPages->findByInternalName('email_item_validation_non_register_user');
    $locale = osc_current_user_locale();

    $content = array();
    if (isset($aPage['locale'][$locale]['s_title'])) {
        $content = $aPage['locale'][$locale];
    } else {
        $content = current($aPage['locale']);
    }

    $item_url = osc_item_url();
    $item_url = '<a href="' . $item_url . '" >' . $item_url . '</a>';
    $edit_url = osc_item_edit_url($item['s_secret'], $item['pk_i_id']);
    $delete_url = osc_item_delete_url($item['s_secret'], $item['pk_i_id']);

    $all = '';

    if (isset($item['locale'])) {
        foreach ($item['locale'] as $locale => $data) {
            $locale_name = OSCLocale::newInstance()->findByCode($locale);
            $all .= '<br/>';
            if (isset($locale_name[0]) && isset($locale_name[0]['s_name'])) {
                $all .= --('Language') . ': ' . $locale_name[0]['s_name'] . '<br/>';
            } else {
                $all .= --('Language') . ': ' . $locale . '<br/>';
            }
            $all .= --('Title') . ': ' . $data['s_title'] . '<br/>';
            $all .= --('Description') . ': ' . $data['s_description'] . '<br/>';
            $all .= '<br/>';
        }
    } else {
        $all .= --('Title') . ': ' . $item['s_title'] . '<br/>';
        $all .= --('Description') . ': ' . $item['s_description'] . '<br/>';
    }

    // Format activation URL
    $validation_url = osc_item_activate_url($item['s_secret'], $item['pk_i_id']);

    $words = array();
    $words[] = array(
        '{ITEM_DESCRIPTION_ALLLANGUAGES}',
        '{ITEM_DESCRIPTION}',
        '{ITEM_COUNTRY}',
        '{ITEM_PRICE}',
        '{ITEM_REGION}',
        '{ITEM_CITY}',
        '{ITEM_ID}',
        '{USER_NAME}',
        '{USER_EMAIL}',
        '{ITEM_TITLE}',
        '{ITEM_URL}',
        '{VALIDATION_LINK}',
        '{VALIDATION_URL}',
        '{EDIT_LINK}',
        '{EDIT_URL}',
        '{DELETE_LINK}',
        '{DELETE_URL}'
    );
    $words[] = array(
        $all,
        $item['s_description'],
        $item['s_country'],
        osc_format_price($item['i_price']),
        $item['s_region'],
        $item['s_city'],
        $item['pk_i_id'],
        $item['s_contact_name'],
        $item['s_contact_email'],
        $item['s_title'],
    );
}

```



```

        $item_url,
        '<a href="' . $validation_url . '" >' . $validation_url . '</a>',
        $validation_url,
        '<a href="' . $edit_url . '">' . $edit_url . '</a>',
        $edit_url,
        '<a href="' . $delete_url . '">' . $delete_url . '</a>',
        $delete_url
    );
    $title = osc_mailBeauty(osc_apply_filter('email_title', osc_apply_filter('
        email_item_validation_non_register_user_title', $content['s_title'])), $words);
    $body = osc_mailBeauty(osc_apply_filter('email_description', osc_apply_filter('
        email_item_validation_non_register_user_description', $content['s_text'])), $words
    );

    $emailParams = array (
        'subject' => $title,
        'from' => osc_contact_email(),
        'to' => $item['s_contact_email'],
        'to_name' => $item['s_contact_name'],
        'body' => $body,
        'alt_body' => $body
    );

    osc_sendMail($emailParams);
}
osc_add_hook('hook_email_item_validation_non_register_user', '
    fn_email_item_validation_non_register_user');

function fn_email_admin_new_user($user) {
    $pageManager = new Page();
    $locale = osc_current_user_locale();
    $aPage = $pageManager->findByInternalName('email_admin_new_user');
    $content = array();

    if (isset($aPage['locale'][$locale]['s_title'])) {
        $content = $aPage['locale'][$locale];
    } else {
        $content = current($aPage['locale']);
    }

    if (!is_null($content)) {
        $words = array();
        $words[] = array(
            '{USER_NAME}',
            '{USER_EMAIL}'
        );
        $words[] = array(
            $user['s_name'],
            $user['s_email']
        );
        $title = osc_mailBeauty(osc_apply_filter('email_title', osc_apply_filter('
            email_user_registration_title', $content['s_title'])), $words);
        $body = osc_mailBeauty(osc_apply_filter('email_description', osc_apply_filter('
            email_user_registration_description', $content['s_text'])), $words);

        $emailParams = array (
            'subject' => $title,
            'from' => osc_contact_email(),
            'to' => osc_contact_email(),
            'to_name' => osc_page_title(),
            'body' => $body,
            'alt_body' => $body,
        );
        osc_sendMail($emailParams);
    }
}
osc_add_hook('hook_email_admin_new_user', 'fn_email_admin_new_user');

function fn_email_contact_user($id, $yourEmail, $yourName, $phoneNumber, $message) {
    $mPages = new Page();
    $aPage = $mPages->findByInternalName('email_contact_user');
    $locale = osc_current_user_locale();

    $content = array();
    if (isset($aPage['locale'][$locale]['s_title'])) {
        $content = $aPage['locale'][$locale];
    } else {
        $content = current($aPage['locale']);
    }

    $words = array();
    $words[] = array(
        '{CONTACT_NAME}',
        '{USER_NAME}',
        '{USER_EMAIL}',
        '{USER_PHONE}',
        '{COMMENT}'
    );
    $words[] = array(
        osc_user_name(),
        $yourName,
        $yourEmail,
        $phoneNumber,
        $message
    );

    $title = osc_mailBeauty(osc_apply_filter('email_title', osc_apply_filter('
        email_item_inquiry_title', $content['s_title'])), $words);
    $body = osc_mailBeauty(osc_apply_filter('email_description', osc_apply_filter('

```

```

        email_item_inquiry_description ', $content['s_text'])); $words);

$emailParams = array (
    'from'      => osc_contact_email(),
    'subject'   => $title,
    'to'        => osc_user_email(),
    'to_name'   => osc_user_name(),
    'body'      => $body,
    'alt_body'  => $body,
    'reply_to' => $yourEmail
);

if( osc_notify_contact_item() ) {
    $emailParams['add_bcc'] = osc_contact_email();
}

osc_sendMail($emailParams);
}
osc_add_hook('hook_email_contact_user', 'fn_email_contact_user');

function fn_email_new_comment_user($aItem) {
    $authorName = trim($aItem['authorName']);
    $authorName = strip_tags($authorName);
    $authorEmail = trim($aItem['authorEmail']);
    $authorEmail = strip_tags($authorEmail);
    $body = trim($aItem['body']);
    // only \n -> <br/>
    $body = nl2br(strip_tags($body));
    $title = $aItem['title'];
    $itemId = $aItem['id'];
    $admin_email = osc_contact_email();

    $item = Item::newInstance()->findByPrimaryKey($itemId);
    View::newInstance()->exportVariableToView('item', $item);
    $itemURL = osc_item_url();
    $itemURL = '<a href="'. $itemURL. '" >'. $itemURL. '</a>';

    $mPages = new Page();
    $aPage = $mPages->findByName('email_new_comment_user');
    $locale = osc_current_user_locale();

    $content = array();
    if (isset($aPage['locale'][$locale]['s_title'])) {
        $content = $aPage['locale'][$locale];
    } else {
        $content = current($aPage['locale']);
    }

    $words = array();
    $words[] = array(
        '{COMMENT.AUTHOR}',
        '{COMMENT.EMAIL}',
        '{COMMENT.TITLE}',
        '{COMMENT.TEXT}',
        '{ITEM.TITLE}',
        '{ITEM.ID}',
        '{ITEM.URL}',
        '{ITEM.LINK}',
        '{SELLER.NAME}',
        '{SELLER.EMAIL}'
    );

    $words[] = array(
        $authorName,
        $authorEmail,
        $title,
        $body,
        $item['s_title'],
        $itemId,
        osc_item_url(),
        $itemURL,
        $item['s_contact_name'],
        $item['s_contact_email']
    );

    $title_email = osc_mailBeauty(osc_apply_filter('email_title', osc_apply_filter('email_new_comment_user_title', $content['s_title'])), $words);
    $body_email = osc_mailBeauty(osc_apply_filter('email_description', osc_apply_filter('email_new_comment_user_description', $content['s_text'])), $words);

    $emailParams = array(
        'from'      => $admin_email,
        'subject'   => $title_email,
        'to'        => $item['s_contact_email'],
        'to_name'   => $item['s_contact_name'],
        'body'      => $body_email,
        'alt_body'  => $body_email
    );
    osc_sendMail($emailParams);
}
osc_add_hook('hook_email_new_comment_user', 'fn_email_new_comment_user');

function fn_email_new_admin($data) {

    $name = trim($data['s_name']);
    $name = strip_tags($name);
    $username = trim($data['s_username']);
    $username = strip_tags($username);

    $mPages = new Page();
    $aPage = $mPages->findByName('email_new_admin');
}

```

```

$locale = osc_current_user_locale();

$content = array();
if (isset($aPage['locale'][$locale]['s_title'])) {
    $content = $aPage['locale'][$locale];
} else {
    $content = current($aPage['locale']);
}

$words = array();
$words[] = array(
    '{ADMIN_NAME}',
    '{USERNAME}',
    '{PASSWORD}',
    '{WEB_ADMIN_LINK}'
);
$words[] = array(
    $data['s_name'],
    $data['s_username'],
    $data['s_password'],
    '<a href="' . osc_admin_base_url() . '">' . osc_page_title() . '</a>',
);
$title_email = osc_mailBeauty(osc_apply_filter('email_title', osc_apply_filter('email_new_admin_title', $content['s_title'])), $words);
$body_email = osc_mailBeauty(osc_apply_filter('email_description', osc_apply_filter('email_new_admin_description', $content['s_text'])), $words);

$emailParams = array(
    'from' => osc_contact_email(),
    'subject' => $title_email,
    'to' => $data['s_email'],
    'to_name' => $data['s_name'],
    'body' => $body_email,
    'alt_body' => $body_email
);
osc_sendMail($emailParams);
}
osc_add_hook('hook_email_new_admin', 'fn_email_new_admin');

function fn_email_warn_expiration($aItem) {
    $itemId = $aItem['pk_i_id'];
    $admin_email = osc_contact_email();

    View::newInstance()->exportVariableToView('item', $aItem);
    $itemURL = osc_item_url();
    $itemURL = '<a href="' . $itemURL . '" >' . $itemURL . '</a>';

    $mPages = new Page();
    $aPage = $mPages->findByInternalName('email_warn_expiration');
    $locale = osc_current_user_locale();

    $content = array();
    if (isset($aPage['locale'][$locale]['s_title'])) {
        $content = $aPage['locale'][$locale];
    } else {
        $content = current($aPage['locale']);
    }

    $words = array();
    $words[] = array(
        '{USER_NAME}',
        '{ITEM_TITLE}',
        '{ITEM_ID}',
        '{ITEM_EXPIRATION_DATE}',
        '{ITEM_URL}',
        '{ITEM_LINK}',
        '{SELLER_NAME}',
        '{SELLER_EMAIL}'
    );
    $words[] = array(
        $aItem['s_contact_name'],
        $aItem['s_title'],
        $itemId,
        $aItem['dt_expiration'],
        osc_item_url(),
        $itemURL,
        $aItem['s_contact_name'],
        $aItem['s_contact_email']
    );
    $title_email = osc_mailBeauty(osc_apply_filter('email_title', osc_apply_filter('email_warn_expiration_title', $content['s_title'])), $words);
    $body_email = osc_mailBeauty(osc_apply_filter('email_description', osc_apply_filter('email_warn_expiration_description', $content['s_text'])), $words);

    $emailParams = array(
        'from' => $admin_email,
        'subject' => $title_email,
        'to' => $aItem['s_contact_email'],
        'to_name' => $aItem['s_contact_name'],
        'body' => $body_email,
        'alt_body' => $body_email
    );
    osc_sendMail($emailParams);
}
osc_add_hook('hook_email_warn_expiration', 'fn_email_warn_expiration');

/* file end: ./oc-includes/osclass/emails.php */
?>

```

```

<?php
/**
 * Oclass      software for creating and publishing online classified advertising platforms
 *
 * Copyright (C) 2012 OSCLASS
 *
 * This program is free software: you can redistribute it and/or modify it under the terms
 * of the GNU Affero General Public License as published by the Free Software Foundation,
 * either version 3 of the License, or (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY;
 * without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
 * See the GNU Affero General Public License for more details.
 *
 * You should have received a copy of the GNU Affero General Public
 * License along with this program. If not, see <http://www.gnu.org/licenses/>.
 */

/**
 * Escape all the values of an array.
 *
 * @param array $array Array used to apply addslashes.
 * @return array $array after apply addslashes.
 */
function add_slashes_extended( $array ) {
    if(get_magic_quotes_gpc()) {
        return $array;
    }

    foreach ( (array) $array as $k => $v ) {
        if( is_array($v) ) {
            $array[$k] = add_slashes_extended($v);
        } else {
            $array[$k] = addslashes($v);
        }
    }

    return $array;
}

function osc_sanitizeString($string) {
    $string = strip_tags($string);
    $string = preg_replace('/%([a-fA-F0-9][a-fA-F0-9])/','--$1--', $string);
    $string = str_replace('%','',$string);
    $string = preg_replace('/--[a-fA-F0-9][a-fA-F0-9]--/','%$1', $string);

    $string = remove_accents($string);

    $string = strtolower($string);
    $string = preg_replace('/&.+?;/','',$string);
    $string = str_replace('.', '-', $string);
    $string = preg_replace('/\s+/', '-', $string);
    $string = preg_replace('|[\p{Ps}\p{Pe}\p{Pi}\p{Pf}\p{Po}\p{S}\p{Z}\p{C}\p{No}]|+|u','',$string);

    if( is_utf8($string) ) {
        $string = urlencode($string);
        // mdash & ndash
        $string = str_replace(array('%e2%80%93','%e2%80%94'), '-', strtolower($string));
    }

    $string = preg_replace('/-+/', '-', $string);
    $string = trim($string, '-');

    return $string;
}

function remove_accents($string) {
    if ( !preg_match('/[\x80-\xff]/', $string) )
        return $string;

    if (is_utf8($string)) {
        $chars = array(
            // Decompositions for Latin-1 Supplement
            chr(195).chr(128) => 'A', chr(195).chr(129) => 'A',
            chr(195).chr(130) => 'A', chr(195).chr(131) => 'A',
            chr(195).chr(132) => 'A', chr(195).chr(133) => 'A',
            chr(195).chr(135) => 'C', chr(195).chr(136) => 'E',
            chr(195).chr(137) => 'E', chr(195).chr(138) => 'E',
            chr(195).chr(139) => 'E', chr(195).chr(140) => 'I',
            chr(195).chr(141) => 'I', chr(195).chr(142) => 'I',
            chr(195).chr(143) => 'I', chr(195).chr(145) => 'N',
            chr(195).chr(146) => 'O', chr(195).chr(147) => 'O',
            chr(195).chr(148) => 'O', chr(195).chr(149) => 'O',
            chr(195).chr(150) => 'O', chr(195).chr(153) => 'U',
            chr(195).chr(154) => 'U', chr(195).chr(155) => 'U',
            chr(195).chr(156) => 'U', chr(195).chr(157) => 'Y',
            chr(195).chr(159) => 's', chr(195).chr(160) => 'a',
            chr(195).chr(161) => 'a', chr(195).chr(162) => 'a',
            chr(195).chr(163) => 'a', chr(195).chr(164) => 'a',
            chr(195).chr(165) => 'a', chr(195).chr(167) => 'c',
            chr(195).chr(168) => 'e', chr(195).chr(169) => 'e',
            chr(195).chr(170) => 'e', chr(195).chr(171) => 'e',
            chr(195).chr(172) => 'i', chr(195).chr(173) => 'i',
            chr(195).chr(174) => 'i', chr(195).chr(175) => 'i',
            chr(195).chr(177) => 'n', chr(195).chr(178) => 'o',
            chr(195).chr(179) => 'o', chr(195).chr(180) => 'o',
            chr(195).chr(181) => 'o', chr(195).chr(182) => 'o',
            chr(195).chr(182) => 'o', chr(195).chr(185) => 'u',
        );
    }
}

```

```

chr(195).chr(186) => 'u', chr(195).chr(187) => 'u',
chr(195).chr(188) => 'u', chr(195).chr(189) => 'y',
chr(195).chr(191) => 'y',
// Decompositions for Latin Extended-A
chr(196).chr(128) => 'A', chr(196).chr(129) => 'a',
chr(196).chr(130) => 'A', chr(196).chr(131) => 'a',
chr(196).chr(132) => 'A', chr(196).chr(133) => 'a',
chr(196).chr(134) => 'C', chr(196).chr(135) => 'c',
chr(196).chr(136) => 'C', chr(196).chr(137) => 'c',
chr(196).chr(138) => 'C', chr(196).chr(139) => 'c',
chr(196).chr(140) => 'C', chr(196).chr(141) => 'c',
chr(196).chr(142) => 'D', chr(196).chr(143) => 'd',
chr(196).chr(144) => 'D', chr(196).chr(145) => 'd',
chr(196).chr(146) => 'E', chr(196).chr(147) => 'e',
chr(196).chr(148) => 'E', chr(196).chr(149) => 'e',
chr(196).chr(150) => 'E', chr(196).chr(151) => 'e',
chr(196).chr(152) => 'E', chr(196).chr(153) => 'e',
chr(196).chr(154) => 'E', chr(196).chr(155) => 'e',
chr(196).chr(156) => 'G', chr(196).chr(157) => 'g',
chr(196).chr(158) => 'G', chr(196).chr(159) => 'g',
chr(196).chr(160) => 'G', chr(196).chr(161) => 'g',
chr(196).chr(162) => 'G', chr(196).chr(163) => 'g',
chr(196).chr(164) => 'H', chr(196).chr(165) => 'h',
chr(196).chr(166) => 'H', chr(196).chr(167) => 'h',
chr(196).chr(168) => 'I', chr(196).chr(169) => 'i',
chr(196).chr(170) => 'I', chr(196).chr(171) => 'i',
chr(196).chr(172) => 'I', chr(196).chr(173) => 'i',
chr(196).chr(174) => 'I', chr(196).chr(175) => 'i',
chr(196).chr(176) => 'I', chr(196).chr(177) => 'i',
chr(196).chr(178) => 'I', chr(196).chr(179) => 'ij',
chr(196).chr(180) => 'J', chr(196).chr(181) => 'j',
chr(196).chr(182) => 'K', chr(196).chr(183) => 'k',
chr(196).chr(184) => 'k', chr(196).chr(185) => 'L',
chr(196).chr(186) => 'l', chr(196).chr(187) => 'L',
chr(196).chr(188) => 'l', chr(196).chr(189) => 'L',
chr(196).chr(190) => 'l', chr(196).chr(191) => 'L',
chr(197).chr(128) => 'l', chr(197).chr(129) => 'L',
chr(197).chr(130) => 'l', chr(197).chr(131) => 'N',
chr(197).chr(132) => 'n', chr(197).chr(133) => 'N',
chr(197).chr(134) => 'n', chr(197).chr(135) => 'N',
chr(197).chr(136) => 'n', chr(197).chr(137) => 'N',
chr(197).chr(138) => 'n', chr(197).chr(139) => 'N',
chr(197).chr(140) => 'O', chr(197).chr(141) => 'o',
chr(197).chr(142) => 'O', chr(197).chr(143) => 'o',
chr(197).chr(144) => 'O', chr(197).chr(145) => 'o',
chr(197).chr(146) => 'OE', chr(197).chr(147) => 'oe',
chr(197).chr(148) => 'R', chr(197).chr(149) => 'r',
chr(197).chr(150) => 'R', chr(197).chr(151) => 'r',
chr(197).chr(152) => 'R', chr(197).chr(153) => 'r',
chr(197).chr(154) => 'S', chr(197).chr(155) => 's',
chr(197).chr(156) => 'S', chr(197).chr(157) => 's',
chr(197).chr(158) => 'S', chr(197).chr(159) => 's',
chr(197).chr(160) => 'S', chr(197).chr(161) => 's',
chr(197).chr(162) => 'T', chr(197).chr(163) => 't',
chr(197).chr(164) => 'T', chr(197).chr(165) => 't',
chr(197).chr(166) => 'T', chr(197).chr(167) => 't',
chr(197).chr(168) => 'U', chr(197).chr(169) => 'u',
chr(197).chr(170) => 'U', chr(197).chr(171) => 'u',
chr(197).chr(172) => 'U', chr(197).chr(173) => 'u',
chr(197).chr(174) => 'U', chr(197).chr(175) => 'u',
chr(197).chr(176) => 'U', chr(197).chr(177) => 'u',
chr(197).chr(178) => 'U', chr(197).chr(179) => 'u',
chr(197).chr(180) => 'W', chr(197).chr(181) => 'w',
chr(197).chr(182) => 'Y', chr(197).chr(183) => 'y',
chr(197).chr(184) => 'Y', chr(197).chr(185) => 'Z',
chr(197).chr(186) => 'z', chr(197).chr(187) => 'Z',
chr(197).chr(188) => 'z', chr(197).chr(189) => 'Z',
chr(197).chr(190) => 'z', chr(197).chr(191) => 's',
// Euro Sign
chr(226).chr(130).chr(172) => 'E',
// GBP (Pound) Sign
chr(194).chr(163) => '' );
} else {
    $string = strtr($string, $chars);
    // Assume ISO-8859-1 if not UTF-8
    $chars['in'] = chr(128).chr(131).chr(138).chr(142).chr(154).chr(158)
        .chr(159).chr(162).chr(165).chr(181).chr(192).chr(193).chr(194)
        .chr(195).chr(196).chr(197).chr(199).chr(200).chr(201).chr(202)
        .chr(203).chr(204).chr(205).chr(206).chr(207).chr(209).chr(210)
        .chr(211).chr(212).chr(213).chr(214).chr(216).chr(217).chr(218)
        .chr(219).chr(220).chr(221).chr(224).chr(225).chr(226).chr(227)
        .chr(228).chr(229).chr(231).chr(232).chr(233).chr(234).chr(235)
        .chr(236).chr(237).chr(238).chr(239).chr(241).chr(242).chr(243)
        .chr(244).chr(245).chr(246).chr(248).chr(249).chr(250).chr(251)
        .chr(252).chr(253).chr(255);
    $chars['out'] = "
        EfSZszYcYuAAAAACEEEEEHIIINOOUUUUYaaaaaceeeiiiinoouuuuy";
    $string = strtr($string, $chars['in'], $chars['out']);
    $double_chars['in'] = array(chr(140), chr(156), chr(198), chr(208), chr(222),
        chr(223), chr(230), chr(240), chr(254));
    $double_chars['out'] = array('OE', 'oe', 'AE', 'DH', 'TH', 'ss', 'ae', 'dh',
        'th');
    $string = str_replace($double_chars['in'], $double_chars['out'], $string);
}
return $string;

```

```

}

function is_utf8($string) {
    return preg_match('%^(?:
        [\x09\x0A\x0D\x20-\x7E]          # ASCII
        | [\xC2-\xDF][\x80-\xBF]       # non-overlong 2-byte
        | \xE0[\xA0-\xBF][\x80-\xBF]   # excluding overlongs
        | [\xE1-\xEC\xEE\xEF][\x80-\xBF]{2} # straight 3-byte
        | \xED[\x80-\x9F][\x80-\xBF]   # excluding surrogates
        | \xF0[\x90-\xBF][\x80-\xBF]{2} # planes 1-3
        | [\xF1-\xF3][\x80-\xBF]{3}    # planes 4-15
        | \xF4[\x80-\xBF]{2}          # plane 16
    )*%xs', $string);
}

?>

<?php
/*
 *      Oclass      software for creating and publishing online classified
 *                  advertising platforms
 *
 *                  Copyright (C) 2012 OSCLASS
 *
 *      This program is free software: you can redistribute it and/or
 *      modify it under the terms of the GNU Affero General Public License
 *      as published by the Free Software Foundation, either version 3 of
 *      the License, or (at your option) any later version.
 *
 *      This program is distributed in the hope that it will be useful, but
 *      WITHOUT ANY WARRANTY; without even the implied warranty of
 *      MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 *      GNU Affero General Public License for more details.
 *
 *      You should have received a copy of the GNU Affero General Public
 *      License along with this program. If not, see <http://www.gnu.org/licenses/>.
 */

function osc_meta_publish($catid = null) {
    osc_enqueue_script('php-date');
    echo '<div class="row">';
        FieldForm::meta_fields_input($catid);
    echo '</div>';
}

function osc_meta_edit($catid = null, $item_id = null) {
    osc_enqueue_script('php-date');
    echo '<div class="row">';
        FieldForm::meta_fields_input($catid, $item_id);
    echo '</div>';
}

osc_add_hook('item_form', 'osc_meta_publish');
osc_add_hook('item_edit', 'osc_meta_edit');

/**
 * All CF will be searchable
 */
function osc_meta_search($catid = null) {
    FieldForm::meta_fields_search($catid);
}
osc_add_hook('search_form', 'osc_meta_search');

function osc_customfields_search_conditions($params) {

    if (isset($params['meta']) && is_array($params['meta'])) {
        $stable = DB_TABLE_PREFIX.'t_item_meta';

        $addTable = false;
        foreach($params['meta'] as $key => $aux) {
            if (is_numeric($key)) {
                $field = Field::newInstance()->findByPrimaryKey($key);
                switch ($field['e_type']) {
                    case 'TEXTAREA':
                    case 'TEXT':
                    case 'URL':
                        if ($aux!='') {
                            $aux = "%$aux%";
                            $sql = "SELECT fk_i_item_id FROM $stable WHERE ";
                            $str_escaped = Search::newInstance()->dao->escape($aux);
                            $sql .= $stable.'.fk_i_field_id = '.$key.' AND ';
                            $sql .= $stable.'.s_value LIKE ".$str_escaped;
                            Search::newInstance()->addConditions('pk_i_id IN ( '.$sql.' )');
                            $addTable = true;
                        }
                        break;
                    case 'DROPDOWN':
                    case 'RADIO':
                        if ($aux!='') {
                            $sql = "SELECT fk_i_item_id FROM $stable WHERE ";
                            $str_escaped = Search::newInstance()->dao->escape($aux);
                            $sql .= $stable.'.fk_i_field_id = '.$key.' AND ';
                            $sql .= $stable.'.s_value = ".$str_escaped;
                            Search::newInstance()->addConditions('pk_i_id IN ( '.$sql.' )');
                            $addTable = true;
                        }
                }
            }
        }
    }
}

```



```

        $result .= $region;
    }
}

return $result;
}

function meta_title() {
    $location = Rewrite::newInstance()->get_location();
    $section = Rewrite::newInstance()->get_section();

    switch ($location) {
        case ('item'):
            switch ($section) {
                case 'item_add': $text = --('Publish a listing'); break;
                case 'item_edit': $text = --('Edit your listing'); break;
                case 'send_friend': $text = --('Send to a friend') . ' - ' . osc_item_title();
                    break;
                case 'contact': $text = --('Contact seller') . ' - ' . osc_item_title();
                    break;
                default: $text = osc_item_title() . ' ' . osc_item_city(); break;
            }
        break;
        case('page'):
            $text = osc_static_page_title();
        break;
        case('error'):
            $text = --('Error');
        break;
        case('search'):
            $region = osc_search_region();
            $city = osc_search_city();
            $pattern = osc_search_pattern();
            $category = osc_search_category_id();
            $s_page = '';
            $i_page = Params::getParam('iPage');

            if($i_page != '' && $i_page > 1) {
                $s_page = ' - ' . --('page') . ' ' . $i_page;
            }

            $b_show_all = ($region == '' && $city == '' & $pattern == '' && $category == '');
            $b_category = ($category != '');
            $b_pattern = ($pattern != '');
            $b_city = ($city != '');
            $b_region = ($region != '');

            if($b_show_all) {
                $text = --('Show all listings') . ' - ' . $s_page . osc_page_title();
            }

            $result = '';
            if($b_pattern) {
                $result .= $pattern . ' &raquo; ';
            }

            if($b_category && is_array($category) && count($category) > 0) {
                $cat = Category::newInstance()->findByPrimaryKey($category[0]);
                if($cat) {
                    $result .= $cat['s_name'] . ' ';
                }
            }

            if($b_city && $b_region) {
                $result .= $city . ' &raquo; ';
            } else if($b_city) {
                $result .= $city . ' &raquo; ';
            } else if($b_region) {
                $result .= $region . ' &raquo; ';
            }

            $result = preg_replace('|\\s?&raquo;\\s$|', '', $result);

            if($result == '') {
                $result = --('Search results');
            }

            $text = '';
            if( osc_get_preference('seo_title_keyword') != '' ) {
                $text .= osc_get_preference('seo_title_keyword') . ' ';
            }
            $text .= $result . $s_page;
        break;
        case('login'):
            switch ($section) {
                case('recover'): $text = --('Recover your password');
            }
            default: $text = --('Login');
        }
        break;
        case('register'):
            $text = --('Create a new account');
        break;
        case('user'):
            switch ($section) {
                case('dashboard'): $text = --('Dashboard'); break;
                case('items'): $text = --('Manage my listings'); break;
                case('alerts'): $text = --('Manage my alerts'); break;
                case('profile'): $text = --('Update my profile'); break;
                case('pub_profile'): $text = --('Public profile') . ' - ' . osc_user_name
                    (); break;
            }
        break;
    }
}

```



```

        case('change_email'): $text = _('Change my email'); break;
        case('change_username'): $text = _('Change my username'); break;
        case('change_password'): $text = _('Change my password'); break;
        case('forgot'): $text = _('Recover my password'); break;
    }
    break;
    case('contact'):
        $text = _('Contact', 'modern');
    break;
    default:
        $text = osc_page_title();
    break;
}

if( !osc_is_home_page() ) {
    $text .= ' - ' . osc_page_title();
}

return (osc_apply_filter('meta_title_filter', $text));
}

function meta_description( ) {
    $text = '';
    // home page
    if( osc_is_home_page() ) {
        $text = osc_page_description();
    }
    // static page
    if( osc_is_static_page() ) {
        $text = osc_highlight(osc_static_page_text(), 140, '', '');
    }
    // search
    if( osc_is_search_page() ) {
        if( osc_has_items() ) {
            $text = osc_item_category() . ' ' . osc_item_city() . ' ' . osc_highlight(
                osc_item_description(), 120) . ' ' . osc_item_category() . ' ' .
                osc_item_city();
        }
        osc_reset_items();
    }
    // listing
    if( osc_is_ad_page() ) {
        $text = osc_item_category() . ' ' . osc_item_city() . ' ' . osc_highlight(
            osc_item_description(), 120) . ' ' . osc_item_category() . ' ' . osc_item_city();
    }

    return (osc_apply_filter('meta_description_filter', $text));
}

function meta_keywords( ) {
    $text = '';
    // search
    if( osc_is_search_page() ) {
        if( osc_has_items() ) {
            $keywords = array();
            $keywords[] = osc_item_category();
            if( osc_item_city() != '' ) {
                $keywords[] = osc_item_city();
                $keywords[] = sprintf('%s %s', osc_item_category(), osc_item_city());
            }
            if( osc_item_region() != '' ) {
                $keywords[] = osc_item_region();
                $keywords[] = sprintf('%s %s', osc_item_category(), osc_item_region());
            }
            if( (osc_item_city() != '') && (osc_item_region() != '') ) {
                $keywords[] = sprintf('%s %s %s', osc_item_category(), osc_item_region(),
                    osc_item_city());
                $keywords[] = sprintf('%s %s', osc_item_region(), osc_item_city());
            }
            $text = implode(' ', $keywords);
        }
        osc_reset_items();
    }
    // listing
    if( osc_is_ad_page() ) {
        $keywords = array();
        $keywords[] = osc_item_category();
        if( osc_item_city() != '' ) {
            $keywords[] = osc_item_city();
            $keywords[] = sprintf('%s %s', osc_item_category(), osc_item_city());
        }
        if( osc_item_region() != '' ) {
            $keywords[] = osc_item_region();
            $keywords[] = sprintf('%s %s', osc_item_category(), osc_item_region());
        }
        if( (osc_item_city() != '') && (osc_item_region() != '') ) {
            $keywords[] = sprintf('%s %s %s', osc_item_category(), osc_item_region(),
                osc_item_city());
            $keywords[] = sprintf('%s %s', osc_item_region(), osc_item_city());
        }
        $text = implode(' ', $keywords);
    }

    return (osc_apply_filter('meta_keywords_filter', $text));
}

function osc_search_footer_links( ) {
    if( !osc_rewrite_enabled() ) {
        return array();
    }
}

```

```

}

$categoryID = '';
if( osc_search_category_id() ) {
    $categoryID = osc_search_category_id();

    if( Category::newInstance()->isRoot( current($categoryID) ) ) {
        $cat = Category::newInstance()->findSubcategories(current($categoryID));
        if( count($cat) > 0 ) {
            $categoryID = array();
            foreach($cat as $c) {
                $categoryID[] = $c['pk_i_id'];
            }
        }
    }
}

if( osc_search_city() != '' ) {
    return array();
}

$regionID = '';
if( osc_search_region() != '' ) {
    $aRegion = Region::newInstance()->findByName(osc_search_region());
    if(isset($aRegion['pk_i_id'])) {
        $regionID = $aRegion['pk_i_id'];
    }
}

$conn = DBConnectionClass::newInstance();
$data = $conn->getOsclassDb();
$comm = new DBCommandClass($data);

$comm->select('i.fk_i_category_id');
$comm->select('l.*');
$comm->select('COUNT(*) AS total');
$comm->from(DB_TABLE_PREFIX . 't_item as i');
$comm->from(DB_TABLE_PREFIX . 't_item_location as l');
if( $categoryID != '' ) {
    $comm->whereIn('i.fk_i_category_id', $categoryID);
}
$comm->where('i.pk_i_id = l.fk_i_item_id');
$comm->where('i.b_enabled = 1');
$comm->where('i.b_active = 1');
$comm->where(sprintf("dt.expiration >= '%s'", date('Y-m-d H:i:s')));

$comm->where('l.fk_i_region_id IS NOT NULL');
$comm->where('l.fk_i_city_id IS NOT NULL');
if( $regionID != '' ) {
    $comm->where('l.fk_i_region_id', $regionID);
    $comm->groupBy('l.fk_i_city_id');
} else {
    $comm->groupBy('l.fk_i_region_id');
}
$rs = $comm->get();

if( !$rs ) {
    return array();
}

return $rs->result();
}

function osc_footer_link_url() {
    $f = View::newInstance()->get('footer_link');
    $url = osc_base_url();

    if( osc_get_preference('seo_url_search_prefix') != '' ) {
        $url .= osc_get_preference('seo_url_search_prefix') . '/';
    }

    $bCategory = false;
    if( osc_search_category_id() ) {
        $bCategory = true;
        $cat = osc_get_category('id', $f['fk_i_category_id']);
        $url .= $cat['s_slug'] . '-';
    }

    if( osc_search_region() == '' ) {
        $url .= osc-sanitizeString($f['s_region']) . '-r' . $f['fk_i_region_id'];
    } else {
        $url .= osc-sanitizeString($f['s_city']) . '-c' . $f['fk_i_city_id'];
    }

    return $url;
}

function osc_footer_link_title() {
    $f = View::newInstance()->get('footer_link');
    $text = '';

    if( osc_get_preference('seo_title_keyword') != '' ) {
        $text .= osc_get_preference('seo_title_keyword') . ' ';
    }

    if( osc_search_category_id() ) {
        $cat = osc_get_category('id', $f['fk_i_category_id']);
        $text .= $cat['s_name'] . ' ';
    }
}

```

```

        if( osc_search_region() == '' ) {
            $text .= $f['s_region'];
        } else {
            $text .= $f['s_city'];
        }

        $text = trim($text);
        return $text;
    }

/**
 * Instantiate the admin toolbar object.
 *
 * @since 3.0
 * @access private
 * @return bool
 */
function _osc_admin_toolbar_init()
{
    $adminToolbar = AdminToolbar::newInstance();

    $adminToolbar->init();
    $adminToolbar->add_menus();
    return true;
}

// and we hook our function via
osc_add_hook( 'init_admin', '_osc_admin_toolbar_init' );

/**
 * Draws admin toolbar
 */
function osc_draw_admin_toolbar()
{
    $adminToolbar = AdminToolbar::newInstance();

    // run hook for adding
    osc_run_hook( 'add_admin_toolbar_menus' );
    $adminToolbar->render();
}

/**
 * Add webtitle with link to frontend
 */
function osc_admin_toolbar_menu()
{
    AdminToolbar::newInstance()->add_menu( array(
        'id' => 'home',
        'title' => '<span class="">'. osc_page_title() . '</span>',
        'href' => osc_base_url(),
        'meta' => array( 'class' => 'user-profile' ),
        'target' => '_blank'
    ) );
}

/**
 * Add logout link
 */
function osc_admin_toolbar_logout()
{
    AdminToolbar::newInstance()->add_menu( array(
        'id' => 'logout',
        'title' => __( 'Logout' ),
        'href' => osc_admin_base_url( true ) . '?action=logout',
        'meta' => array( 'class' => 'btn btn-dim ico ico-32 ico-power float-right' )
    ) );
}

function osc_admin_toolbar_comments()
{
    $total = ItemComment::newInstance()->countAll( '( c.b_active = 0 OR c.b_enabled = 0 OR c.b_spam = 1 )' );
    if( $total > 0 ) {
        $title = '<i class="circle circle-green">'. $total . '</i>'. __( 'New comments' );

        AdminToolbar::newInstance()->add_menu(
            array( 'id' => 'comments',
                'title' => $title,
                'href' => osc_admin_base_url( true ) . "?page=comments",
                'meta' => array( 'class' => 'action-btn action-btn-black' )
            ) );
    }
}

function osc_admin_toolbar_spam()
{
    $total = Item::newInstance()->countByMarkas( 'spam' );
    if( $total > 0 ) {
        $title = '<i class="circle circle-red">'. $total . '</i>'. __( 'Spam' );

        AdminToolbar::newInstance()->add_menu(
            array( 'id' => 'spam',
                'title' => $title,
                'href' => osc_admin_base_url( true ) . "?page=items&action=items-reported
                    &sort=spam",
                'meta' => array( 'class' => 'action-btn action-btn-black' )
            ) );
    }
}

```

```

}

function osc_check_plugins_update( $force = false )
{
    $total = 0;
    $array = array();
    $array_downloaded = array();
    // check if exist a new version each day
    if( (time() - osc_plugins_last_version_check()) > (24 * 3600) || $force ) {
        $plugins = Plugins::listAll();
        foreach($plugins as $plugin) {
            $info = osc_plugin_get_info($plugin);
            if(osc_check_plugin_update(@$info['plugin_update_uri'], @$info['version'])) {
                $array[] = @$info['plugin_update_uri'];
                $total++;
            }else{
                $array_downloaded[] = @$info['plugin_update_uri'];
            }
        }

        osc_set_preference( 'plugins_to_update', json_encode($array) );
        osc_set_preference( 'plugins_downloaded', json_encode($array_downloaded) );
        osc_set_preference( 'plugins_update_count', $total );
        osc_set_preference( 'plugins_last_version_check', time() );
        osc_reset_preferences();
    } else {
        $total = getPreference('plugins_update_count');
    }
}

return $total;
}

function osc_admin_toolbar_update_plugins($force = false)
{
    if( !osc_is_moderator() ) {
        $total = osc_check_plugins_update( $force );

        if($force) {
            AdminToolbar::newInstance()->remove_menu('update-plugin');
        }
        if($total > 0) {
            $title = '<i class="circle circle-gray">'. $total . '</i>'. _('Plugin updates');
            AdminToolbar::newInstance()->add_menu(
                array('id' => 'update-plugin',
                    'title' => $title,
                    'href' => osc_admin_base_url(true) . "?page=plugins#update-plugins",
                    'meta' => array('class' => 'action-btn action-btn-black')
                ) );
        }
    }
}

function osc_check_themes_update( $force = false )
{
    $total = 0;
    $array = array();
    $array_downloaded = array();
    // check if exist a new version each day
    if( (time() - osc_themes_last_version_check()) > (24 * 3600) || $force ) {
        $themes = WebThemes::newInstance()->getListThemes();
        foreach($themes as $theme) {
            $info = WebThemes::newInstance()->loadThemeInfo($theme);
            if(osc_check_theme_update(@$info['theme_update_uri'], @$info['version'])) {
                $array[] = $theme;
                $total++;
            }
            $array_downloaded[] = @$info['theme_update_uri'];
        }
        osc_set_preference( 'themes_to_update', json_encode($array) );
        osc_set_preference( 'themes_downloaded', json_encode($array_downloaded) );
        osc_set_preference( 'themes_update_count', $total );
        osc_set_preference( 'themes_last_version_check', time() );
        osc_reset_preferences();
    } else {
        $total = getPreference('themes_update_count');
    }
}

return $total;
}

function osc_admin_toolbar_update_themes($force = false)
{
    if( !osc_is_moderator() ) {
        $total = osc_check_themes_update( $force );

        if($force) {
            AdminToolbar::newInstance()->remove_menu('update-theme');
        }
        if($total > 0) {
            $title = '<i class="circle circle-gray">'. $total . '</i>'. _('Theme updates');
            AdminToolbar::newInstance()->add_menu(
                array('id' => 'update-theme',
                    'title' => $title,
                    'href' => osc_admin_base_url(true) . "?page=appearance",
                    'meta' => array('class' => 'action-btn action-btn-black')
                ) );
        }
    }
}
}

```

```

}

// languages todo
function osc_check_languages_update( $force = false )
{
    $total = 0;
    $array = array();
    $array_downloaded = array();
    // check if exist a new version each day
    if( (time() - osc_languages_last_version_check()) > (24 * 3600) || $force ) {
        $languages = OSCLocale::newInstance()->listAll();
        foreach($languages as $lang) {
            if(osc_check_language_update($lang['pk_c_code'], $lang['s_version'] )) {
                $array[] = $lang['pk_c_code'];
                $total++;
            }
            $array_downloaded[] = $lang['pk_c_code'];
        }
        osc_set_preference( 'languages_to_update' , json_encode($array) );
        osc_set_preference( 'languages_downloaded' , json_encode($array_downloaded) );
        osc_set_preference( 'languages_update_count' , $total );
        osc_set_preference( 'languages_last_version_check' , time() );
        osc_reset_preferences();
    } else {
        $total = getPreference('languages_update_count');
    }
}

return $total;
}

function osc_admin_toolbar_update_languages($force = false)
{
    if( !osc_is_moderator() ) {
        $total = osc_check_languages_update( $force );

        if($force) {
            AdminToolbar::newInstance()->remove_menu('update_language');
        }
        if($total > 0) {
            $title = '<i class="circle circle-gray">'. $total . '</i>'. '--(Language updates)';
            AdminToolbar::newInstance()->add_menu(
                array('id' => 'update_language',
                    'title' => $title,
                    'href' => osc_admin_base_url(true) . "?page=languages",
                    'meta' => array('class' => 'action-btn action-btn-black')
                ) );
        }
    }
}
?>

<?php /* So say we all */ ?>

<?php
/*
 *      Oclass      software for creating and publishing online classified
 *                  advertising platforms
 *
 *                  Copyright (C) 2012 OSCLASS
 *
 *      This program is free software: you can redistribute it and/or
 *      modify it under the terms of the GNU Affero General Public License
 *      as published by the Free Software Foundation, either version 3 of
 *      the License, or (at your option) any later version.
 *
 *      This program is distributed in the hope that it will be useful, but
 *      WITHOUT ANY WARRANTY; without even the implied warranty of
 *      MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 *      GNU Affero General Public License for more details.
 *
 *      You should have received a copy of the GNU Affero General Public
 *      License along with this program. If not, see <http://www.gnu.org/licenses/>.
 */

error_reporting(E_ERROR | E_CORE_ERROR | E_COMPILE_ERROR | E_PARSE);

define( 'ABS_PATH', dirname(dirname(dirname(__FILE__))) . '/' );
define( 'LIB_PATH', ABS_PATH . 'oc-includes/' );
define( 'CONTENT_PATH', ABS_PATH . 'oc-content/' );
define( 'TRANSLATIONS_PATH', CONTENT_PATH . 'languages/' );
define( 'OSC_INSTALLING', 1 );

if(extension_loaded('mysqli')) {
    require_once LIB_PATH . 'osclass/Logger/Logger.php';
    require_once LIB_PATH . 'osclass/Logger/LogDatabase.php';
    require_once LIB_PATH . 'osclass/Logger/LogOsc.php';
    require_once LIB_PATH . 'osclass/classes/database/DBConnectionClass.php';
    require_once LIB_PATH . 'osclass/classes/database/DBCommandClass.php';
    require_once LIB_PATH . 'osclass/classes/database/DBRecordsetClass.php';
    require_once LIB_PATH . 'osclass/classes/database/DAO.php';
    require_once LIB_PATH . 'osclass/model/Preference.php';
    require_once LIB_PATH . 'osclass/helpers/hPreference.php';
}
require_once LIB_PATH . 'osclass/core/Session.php';
require_once LIB_PATH . 'osclass/core/Params.php';
require_once LIB_PATH . 'osclass/helpers/hDatabaseInfo.php';
require_once LIB_PATH . 'osclass/helpers/hDefines.php';
require_once LIB_PATH . 'osclass/helpers/hErrors.php';

```

```

require_once LIB_PATH . 'osclass/helpers/hLocale.php';
require_once LIB_PATH . 'osclass/helpers/hSearch.php';
require_once LIB_PATH . 'osclass/helpers/hPlugins.php';
require_once LIB_PATH . 'osclass/helpers/hTranslations.php';
require_once LIB_PATH . 'osclass/helpers/hSanitize.php';
require_once LIB_PATH . 'osclass/default-constants.php';
require_once LIB_PATH . 'osclass/install-functions.php';
require_once LIB_PATH . 'osclass/utills.php';
require_once LIB_PATH . 'osclass/core/Translation.php';
require_once LIB_PATH . 'osclass/classes/Plugins.php';
require_once LIB_PATH . 'osclass/locales.php';

Session::newInstance()->session_start();

$locales = osc_listLocales();

if(Params::getParam('install_locale')!='') {
    Session::newInstance()->.set('userLocale', Params::getParam('install_locale'));
    Session::newInstance()->.set('adminLocale', Params::getParam('install_locale'));
}

if(Session::newInstance()->.get('adminLocale')!='' && key_exists(Session::newInstance()->.get('adminLocale'), $locales)) {
    $current_locale = Session::newInstance()->.get('adminLocale');
} else if(isset($locales['en-US'])) {
    $current_locale = 'en-US';
} else {
    $current_locale = key($locales);
}

Session::newInstance()->.set('userLocale', $current_locale);
Session::newInstance()->.set('adminLocale', $current_locale);

$translation = Translation::newInstance(true);

$step = Params::getParam('step');
if( !is_numeric($step) ) {
    $step = '1';
}

if( is_osclass_installed() ) {
    $message = _("Looks like you've already installed Oclass. To reinstall please clear your old database tables first.");
    osc_die('Oclass &raquo; Error', $message);
}

switch( $step ) {
case 1:
    $requirements = get_requirements();
    $error = check_requirements($requirements);
    break;
case 2:
    if( Params::getParam('save_stats') == '1' || isset($_COOKIE['osclass_save_stats']) ) {
        setcookie('osclass_save_stats', 1, time() + (24*60*60));
    } else {
        setcookie('osclass_save_stats', 0, time() + (24*60*60));
    }

    if( Params::getParam('ping_engines') == '1' || isset($_COOKIE['osclass_ping_engines']) ) {
        setcookie('osclass_ping_engines', 1, time() + (24*60*60));
    } else {
        setcookie('osclass_ping_engines', 0, time()+ (24*60*60));
    }

    break;
case 3:
    if( Params::getParam('dbname') != '' ) {
        $error = oc_install();
    }
    break;
case 4:
    if( Params::getParam('result') != '' ) {
        $error = Params::getParam('result');
    }
    $password = Params::getParam('password', false, false);
    break;
case 5:
    $password = Params::getParam('password', false, false);
    break;
default:
    break;
}
?>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" dir="ltr" lang="en-US" xml:lang="en-US">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title><?php _e('Oclass Installation'); ?></title>
<script src="<?php echo get_absolute_url(); ?>oc-includes/osclass/assets/js/jquery.min.js" type="text/javascript"></script>
<script src="<?php echo get_absolute_url(); ?>oc-includes/osclass/assets/js/jquery-ui.min.js" type="text/javascript"></script>
<script src="<?php echo get_absolute_url(); ?>oc-includes/osclass/installer/vtip/vtip.js" type="text/javascript"></script>

```

```

<script src="<?php echo get_absolute_url(); ?>oc-includes/osclass/assets/js/jquery.
    json.js" type="text/javascript"></script>
<script src="<?php echo get_absolute_url(); ?>oc-includes/osclass/installer/install.js
    " type="text/javascript"></script>
<script src="<?php echo get_absolute_url(); ?>oc-admin/themes/modern/js/location.js"
    type="text/javascript"></script>
<link rel="stylesheet" type="text/css" media="all" href="<?php echo get_absolute_url()
    ; ?>oc-includes/osclass/installer/install.css" />
<link rel="stylesheet" type="text/css" media="all" href="<?php echo get_absolute_url()
    ; ?>oc-includes/osclass/installer/vtip/css/vtip.css" />
</head>
<body>
    <div id="wrapper">
        <div id="container">
            <div id="header" class="installation">
                <h1 id="logo">
                    
                </h1>
                <?php if(in_array($step, array(2,3))) { ?>
                <ul id="nav">
                    <li class="<?php if($step == 2) { ?>actual<?php } elseif($step < 2) {
                        ?>next<?php } else { ?>past<?php }?>">1 - Database</li>
                    <li class="<?php if($step == 3) { ?>actual<?php } elseif($step < 3) {
                        ?>next<?php } else { ?>past<?php }?>">2 - Target</li>
                </ul>
                <div class="clear"></div>
                <?php } ?>
            </div>
            <div id="content">
                <?php if($step == 1) { ?>
                <h2 class="target"><?php _e('Welcome');?></h2>
                <form action="install.php" method="post">
                    <div class="form-table">
                        <?php if( count($locales) > 1 ) { ?>
                            <div>
                                <label for="install_locale"><?php _e('Choose language ');
                                    ?></label>
                                <select name="install_locale" id="install_locale" onchange
                                    ="window.location.href='install_locale'+document.
                                        getElementById(this.id).value">
                                    <?php foreach($locales as $k => $locale) {?>
                                        <option value="<?php echo osc_esc_html($k); ?>" <?php
                                            if( $k == $current_locale ) { echo 'selected='
                                                selected' '; } ?><?php echo $locale['name']; ?></
                                                option>
                                    <?php } ?>
                                </select>
                            </div>
                            <?php } ?>
                            <?php if($error) { ?>
                            <p><?php _e('Check the next requirements:');?></p>
                            <div class="requirements_help">
                                <p><b><?php _e('Requirements help:'); ?></b></p>
                                <ul>
                                    <?php foreach($requirements as $k => $v) { ?>
                                        <?php if(!$v['fn'] && $v['solution'] != '') { ?>
                                        <li><?php echo $v['solution']; ?></li>
                                        <?php } ?>
                                    <?php } ?>
                                    <li><a href="http://forums.osclass.org/" hreflang="en"><?
                                        php _e('Need more help?');?></a></li>
                                </ul>
                            </div>
                            <?php } else { ?>
                            <p><?php _e('All right! All the requirements have met:');?></p>
                            <?php } ?>
                            <ul>
                                <?php foreach($requirements as $k => $v) { ?>
                                <li><?php echo $v['requirement']; ?> </li>
                                <?php } ?>
                            </ul>
                            <div class="more-stats">
                                <input type="checkbox" name="ping_engines" id="ping_engines"
                                    checked="checked" value="1" />
                                <label for="ping_engines">
                                    <?php _e('Allow my site to appear in search engines like
                                        Google. ');?>
                                </label>
                                <br />
                                <input type="checkbox" name="save_stats" id="save_stats"
                                    checked="checked" value="1" />
                                <input type="hidden" name="step" value="2" />
                                <label for="save_stats">
                                    <?php _e('Help make Osclass better by automatically
                                        sending usage statistics and crash reports to Osclass
                                        . ');?>
                                </label>
                            </div>
                        </div>
                        <?php if($error) { ?>
                        <p class="margin20">
                            <input type="button" class="button" onclick="document.location =
                                'install.php?step=1' value="<?php echo osc_esc_html( _e('Try
                                    again'));?>" />
                        </p>
                        <?php } else { ?>

```

```

        <p class="margin20">
            <input type="submit" class="button" value="<?php echo osc_esc_html
                ( --('Run the install ');?>" />
        </p>
    <?php } ?>
</form>
<?php } elseif($step == 2) {
    display_database_config();
} elseif($step == 3) {
    if( !isset($error["error"]) ) {
        display_target();
    } else {
        display_database_error($error, ($step - 1));
    }
} elseif($step == 4) {
    // ping engines
    ping_search_engines( $_COOKIE['osclass_ping_engines'] );
    setcookie('osclass_save_stats', '', time() - 3600);
    setcookie('osclass_ping_engines', '', time() - 3600);
    display_finish($password);
}
?>
</div>
<div id="footer">
    <ul>
        <li>
            <a href="<?php echo get_absolute_url(); ?>/oc-includes/osclass/
                installer/readme.php" target="_blank" hreflang="en"><?php _e('
                Readme'); ?></a>
        </li>
        <li>
            <a href="http://admin.osclass.org/feedback.php" target="_blank"
                hreflang="en"><?php _e('Feedback'); ?></a>
        </li>
        <li>
            <a href="http://forums.osclass.org/index.php" target="_blank"
                hreflang="en"><?php _e('Forums');?></a>
        </li>
    </ul>
</div>
</div>
</body>
</html>

```

```

<?php
/*
 *   Osclass   software for creating and publishing online classified
 *               advertising platforms
 *
 *               Copyright (C) 2012 OSCLASS
 *
 *   This program is free software: you can redistribute it and/or
 *   modify it under the terms of the GNU Affero General Public License
 *   as published by the Free Software Foundation, either version 3 of
 *   the License, or (at your option) any later version.
 *
 *   This program is distributed in the hope that it will be useful, but
 *   WITHOUT ANY WARRANTY; without even the implied warranty of
 *   MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 *   GNU Affero General Public License for more details.
 *
 *   You should have received a copy of the GNU Affero General Public
 *   License along with this program. If not, see <http://www.gnu.org/licenses/>.
 */

/*
 * The url of the site
 *
 * @since 1.2
 *
 * @return string The url of the site
 */
function get_absolute_url( ) {
    $protocol = ( isset($_SERVER['HTTPS']) && $_SERVER['HTTPS'] == 'on' ) ? 'https' : 'http';
    $pos      = strpos($_SERVER['REQUEST_URI'], 'oc-includes');
    $URI      = rtrim( substr( $_SERVER['REQUEST_URI'], 0, $pos ), '/' ) . '/';
    return $protocol . '://' . $_SERVER['HTTP_HOST'] . $URI;
}

/*
 * The relative url on the domain url
 *
 * @since 1.2
 *
 * @return string The relative url on the domain url
 */
function get_relative_url( ) {
    $url = $_SERVER['REQUEST_URI'];
    return substr($url, 0, strpos($url, '/oc-includes')) . "/";
}

/*
 * Get the requirements to install Osclass
 *
 * @since 1.2
 *
 * @return array Requirements
 */

```



```

function get_requirements( ) {
    $array = array(
        'PHP version >= 5.x' => array(
            'requirement' => --('PHP version >= 5.x'),
            'fn' => version_compare(PHP_VERSION, '5.0.0', '>='),
            'solution' => --('PHP5 is required to run Oclass. You may talk with your hosting
                to upgrade your PHP version.')),
        'MySQLi extension for PHP' => array(
            'requirement' => --('MySQLi extension for PHP'),
            'fn' => extension_loaded('mysqli'),
            'solution' => --('MySQLi extension is required. How to <a target="" blank" href="
                http://www.php.net/manual/en/mysqli.setup.php">install/configure</a>.')),
        'GD extension for PHP' => array(
            'requirement' => --('GD extension for PHP'),
            'fn' => extension_loaded('gd'),
            'solution' => --('GD extension is required. How to <a target="" blank" href="http
                ://www.php.net/manual/en/image.setup.php">install/configure</a>.')),
        'Folder <code>oc-content/uploads</code> exists' => array(
            'requirement' => --('Folder <code>oc-content/uploads</code> exists'),
            'fn' => file_exists( ABS_PATH . 'oc-content/uploads/' ),
            'solution' => sprintf(--('You have to create <code>uploads</code> folder, i.e.: <
                code>mkdir %soc-content/uploads/</code>'), ABS_PATH)),
        'Folder <code>oc-content/uploads</code> is writable' => array(
            'requirement' => --('<code>oc-content/uploads</code> folder is writable'),
            'fn' => is_writable( ABS_PATH . 'oc-content/uploads/' ),
            'solution' => sprintf(--('<code>uploads</code> folder has to be writable, i.e.: <
                code>chmod a+w %soc-content/uploads/</code>'), ABS_PATH)),
        // oc-content/downloads
        'Folder <code>oc-content/downloads</code> exists' => array(
            'requirement' => --('Folder <code>oc-content/downloads</code> exists'),
            'fn' => file_exists( ABS_PATH . 'oc-content/downloads/' ),
            'solution' => sprintf(--('You have to create <code>downloads</code> folder, i.e.: <
                code>mkdir %soc-content/downloads/</code>'), ABS_PATH)),
        'Folder <code>oc-content/downloads</code> is writable' => array(
            'requirement' => --('<code>oc-content/downloads</code> folder is writable'),
            'fn' => is_writable( ABS_PATH . 'oc-content/downloads/' ),
            'solution' => sprintf(--('<code>downloads</code> folder has to be writable, i.e.: <
                code>chmod a+w %soc-content/downloads/</code>'), ABS_PATH)),
        // oc-content/languages
        'Folder <code>oc-content/languages</code> exists' => array(
            'requirement' => --('Folder <code>oc-content/languages</code> folder exists'),
            'fn' => file_exists( ABS_PATH . 'oc-content/languages/' ),
            'solution' => sprintf(--('You have to create the <code>languages</code> folder, i.
                e.: <code>mkdir %soc-content/languages/</code>'), ABS_PATH)),
        'Folder <code>oc-content/languages</code> is writable' => array(
            'requirement' => --('<code>oc-content/languages</code> folder is writable'),
            'fn' => is_writable( ABS_PATH . 'oc-content/languages/' ),
            'solution' => sprintf(--('<code>languages</code> folder has to be writable, i.e.: <
                code>chmod a+w %soc-content/languages/</code>'), ABS_PATH)),
    );

    $config_writable = false;
    $root_writable = false;
    $config_sample = false;
    if( file_exists(ABS_PATH . 'config.php') ) {
        if( is_writable(ABS_PATH . 'config.php') ) {
            $config_writable = true;
        }
        $array['File <code>config.php</code> is writable'] = array(
            'requirement' => --('<code>config.php</code> file is writable'),
            'fn' => $config_writable,
            'solution' => sprintf(--('<code>config.php</code> file has to be writable, i.e.: <
                code>chmod a+w %sconfig.php</code>'), ABS_PATH));
    } else {
        if( is_writable(ABS_PATH) ) {
            $root_writable = true;
        }
        $array['Root directory is writable'] = array(
            'requirement' => --('Root directory is writable'),
            'fn' => $root_writable,
            'solution' => sprintf(--('Root folder has to be writable, i.e.: <code>chmod a+w %s
                </code>'), ABS_PATH));
    }

    if( file_exists(ABS_PATH . 'config-sample.php') ) {
        $config_sample = true;
    }
    $array['File <code>config-sample.php</code> exists'] = array(
        'requirement' => --('<code>config-sample.php</code> file exists'),
        'fn' => $config_sample,
        'solution' => --('<code>config-sample.php</code> file is required, you should re-
            download Oclass.'));
    }

    return $array;
}

/**
 * Check if some of the requirements to install Oclass are correct or not
 *
 * @since 1.2
 *
 * @return boolean Check if all the requirements are correct
 */

```

```

*/
function check_requirements($array) {
    foreach($array as $k => $v) {
        if( !$v['fn'] ) return true;
    }
    return false;
}

/**
 * Check if allowed to send stats to Oclass
 *
 * @return boolean Check if allowed to send stats to Oclass
 */
function reportToOclass() {
    return $_COOKIE['osclass_save_stats'];
}

/**
 * insert/update preference allow_report_osclass
 * @param boolean $bool
 */
function set_allow_report_osclass($value) {
    $values = array(
        's_section' => 'osclass',
        's_name' => 'allow_report_osclass',
        's_value' => $value,
        'e_type' => 'BOOLEAN'
    );

    Preference::newInstance()->insert($values);
}

/*
 * Install Oclass database
 *
 * @since 1.2
 *
 * @return mixed Error messages of the installation
 */
function oc_install() {
    $dbhost = Params::getParam('dbhost');
    $dbname = Params::getParam('dbname');
    $username = Params::getParam('username');
    $password = Params::getParam('password', false, false);
    $tableprefix = Params::getParam('tableprefix');
    $createdb = false;

    if( $tableprefix == '' ) {
        $tableprefix = 'oc_';
    }

    if( Params::getParam('createdb') != '' ) {
        $createdb = true;
    }

    if ( $createdb ) {
        $adminuser = Params::getParam('admin_username');
        $adminpwd = Params::getParam('admin_password', false, false);

        $master_conn = new DBConnectionClass($dbhost, $adminuser, $adminpwd, '');
        $error_num = $master_conn->getErrorConnectionLevel();

        if( $error_num > 0 ) {
            if( reportToOclass() ) {
                LogOclassInstaller::instance()->error(sprintf(--('Cannot connect to the
                    database. Error number: %s'), $error_num), __FILE__."::"__LINE__);
            }
        }

        switch ($error_num) {
            case 1049: return array('error' => --("The database doesn't exist. You should
                check the \"Create DB\" checkbox and fill in a username and password with
                the right privileges"));
                break;
            case 1045: return array('error' => --('Cannot connect to the database. Check
                if the user has privileges.'));
                break;
            case 1044: return array('error' => --('Cannot connect to the database. Check
                if the username and password are correct.'));
                break;
            case 2005: return array('error' => --("Can't resolve MySQL host. Check if the
                host is correct."));
                break;
            default: return array('error' => sprintf(--('Cannot connect to the database
                . Error number: %s'), $error_num));
                break;
        }
    }

    $m_db = $master_conn->getOclassDb();
    $comm = new DBCommandClass( $m_db );
    $comm->query( sprintf("CREATE DATABASE IF NOT EXISTS %s DEFAULT CHARACTER SET 'UTF8'
        COLLATE 'UTF8_GENERAL_CI'", $dbname) );

    $error_num = $comm->getErrorLevel();

    if( $error_num > 0 ) {
        if( reportToOclass() ) {
            LogOclassInstaller::instance()->error(sprintf(--("Can't create the database.
                Error number: %s"), $error_num), __FILE__."::"__LINE__);
        }
    }
}

```

```

    }

    if( in_array( $Error_num, array(1006, 1044, 1045) ) ) {
        return array('error' => __("Can't create the database. Check if the admin
            username and password are correct."));
    }

    return array('error' => sprintf(_("Can't create the database. Error number: %s"),
        $Error_num));
}

unset($conn);
unset($comm);
unset($master_conn);
}

$conn = new DBConnectionClass($dbhost, $username, $password, $dbname);
$error_num = $conn->getErrorConnectionLevel();

if( $error_num == 0 ) {
    $error_num = $conn->getErrorLevel();
}

if( $error_num > 0 ) {
    if( reportToOsclass() ) {
        LogOsclassInstaller::instance()->error(sprintf(_("Cannot connect to the database.
            Error number: %s"), $error_num), __FILE__.":".__LINE__);
    }

    switch( $error_num ) {
        case 1049: return array('error' => __("The database doesn't exist. You should
            check the \"Create DB\" checkbox and fill in a username and password with the
            right privileges"));
            break;
        case 1045: return array('error' => __("Cannot connect to the database. Check if
            the user has privileges.'"));
            break;
        case 1044: return array('error' => __("Cannot connect to the database. Check if
            the username and password are correct.'"));
            break;
        case 2005: return array('error' => __("Can't resolve MySQL host. Check if the
            host is correct."));
            break;
        default: return array('error' => sprintf(_("Cannot connect to the database.
            Error number: %s"), $error_num));
            break;
    }
}

if( file_exists(ABS_PATH . 'config.php') ) {
    if( !is_writable(ABS_PATH . 'config.php') ) {
        if( reportToOsclass() ) {
            LogOsclassInstaller::instance()->error(_("Can't write in config.php file.
                Check if the file is writable."), __FILE__.":".__LINE__);
        }
        return array('error' => __("Can't write in config.php file. Check if the file is
            writable."));
    }
    create_config_file($dbname, $username, $password, $dbhost, $stableprefix);
} else {
    if( !file_exists(ABS_PATH . 'config-sample.php') ) {
        if( reportToOsclass() ) {
            LogOsclassInstaller::instance()->error(_("config-sample.php doesn't exist.
                Check if everything is decompressed correctly."), __FILE__.":".__LINE__);
        }
    }

    return array('error' => __("config-sample.php doesn't exist. Check if everything
        is decompressed correctly."));
}

if( !is_writable(ABS_PATH) ) {
    if( reportToOsclass() ) {
        LogOsclassInstaller::instance()->error(_("Can't copy config-sample.php.
            Check if the root directory is writable."), __FILE__.":".__LINE__);
    }
}

return array('error' => __("Can't copy config-sample.php. Check if the root
    directory is writable.'"));
}

copy_config_file($dbname, $username, $password, $dbhost, $stableprefix);
}

require_once ABS_PATH . 'config.php';

$sql = file_get_contents( ABS_PATH . 'oc-includes/osclass/installer/struct.sql' );

$c_db = $conn->getOsclassDb();
$comm = new DBCommandClass( $c_db );
$comm->importSQL($sql);

$error_num = $comm->getErrorLevel();

if( $error_num > 0 ) {
    if( reportToOsclass() ) {
        LogOsclassInstaller::instance()->error(sprintf(_("Can't create the database
            structure. Error number: %s"), $error_num), __FILE__.":".__LINE__);
    }
}

switch ( $error_num ) {

```

```

        case 1050: return array('error' => _('There are tables with the same name in the
            database. Change the table prefix or the database and try again.));
        break;
        default: return array('error' => sprintf(_("Can't create the database
            structure. Error number: %s"), $error_num));
        break;
    }
}

require_once LIB_PATH . 'osclass/model/OSCLocale.php';
$localeManager = OSCLocale::newInstance();

$locales = osc_listLocales();
$values = array(
    'pk_c_code' => $locales[osc_current_admin_locale()][ 'code' ],
    's_name' => $locales[osc_current_admin_locale()][ 'name' ],
    's_short_name' => $locales[osc_current_admin_locale()][ 'short_name' ],
    's_description' => $locales[osc_current_admin_locale()][ 'description' ],
    's_version' => $locales[osc_current_admin_locale()][ 'version' ],
    's_author_name' => $locales[osc_current_admin_locale()][ 'author_name' ],
    's_author_url' => $locales[osc_current_admin_locale()][ 'author_url' ],
    's_currency_format' => $locales[osc_current_admin_locale()][ 'currency_format' ],
    's_date_format' => $locales[osc_current_admin_locale()][ 'date_format' ],
    'b_enabled' => 1,
    'b_enabled_bo' => 1
);

if( isset($locales[osc_current_admin_locale()][ 'stop_words' ]) ) {
    $values['s_stop_words'] = $locales[osc_current_admin_locale()][ 'stop_words' ];
}
$localeManager->insert($values);

$required_files = array(
    ABS_PATH . 'oc-includes/osclass/installer/basic_data.sql',
    ABS_PATH . 'oc-includes/osclass/installer/pages.sql',
    ABS_PATH . 'oc-content/languages/' . osc_current_admin_locale() . '/mail.sql',
);

$sql = '';
foreach($required_files as $file) {
    if ( !file_exists($file) ) {
        if( reportToOsclass() ) {
            LogOsclassInstaller::instance()->error(sprintf(_("The file %s doesn't exist
                '), $file), __FILE__.":".__LINE__);
        }
        return array('error' => sprintf(_("The file %s doesn't exist"), $file));
    } else {
        $sql .= file_get_contents($file);
    }
}
$comm->importSQL($sql);

$error_num = $comm->getErrorLevel();

if( $error_num > 0 ) {
    if( reportToOsclass() ) {
        LogOsclassInstaller::instance()->error(sprintf(_("Can't insert basic
            configuration. Error number: %s"), $error_num), __FILE__.":".__LINE__);
    }
    switch ($error_num) {
        case 1471: return array('error' => _("Can't insert basic configuration. This
            user has no privileges to 'INSERT' into the database.));
        break;
        default: return array('error' => sprintf(_("Can't insert basic configuration.
            Error number: %s"), $error_num));
        break;
    }
}

osc_set_preference('language', osc_current_admin_locale());
osc_set_preference('admin_language', osc_current_admin_locale());
osc_set_preference('csrf_name', 'CSRF'.mt_rand(0,mt_getrandmax()));

oc_install_example_data();

if( reportToOsclass() ) {
    set_allow_report_osclass( true );
} else {
    set_allow_report_osclass( false );
}

return false;
}

/*
 * Insert the example data (categories and emails) on all available locales
 *
 * @since 2.4
 *
 * @return mixed Error messages of the installation
 */
function oc_install_example_data() {
    require_once LIB_PATH . 'osclass/formatting.php';
    require LIB_PATH . 'osclass/installer/basic_data.php';
    require_once LIB_PATH . 'osclass/model/Category.php';

```

```

$mCat = Category::newInstance();

if(!function_exists('osc_apply_filter')) {
    function osc_apply_filter($dummyfilter, $str) {
        return $str;
    }
}

foreach($categories as $category) {

    $fields['pk_i_id'] = $category['pk_i_id'];
    $fields['fk_i_parent_id'] = $category['fk_i_parent_id'];
    $fields['i_position'] = $category['i_position'];
    $fields['i_expiration_days'] = 0;
    $fields['b_enabled'] = 1;

    $aFieldsDescription[osc_current_admin_locale()][ 's_name' ] = $category['s_name'];

    $mCat->insert($fields, $aFieldsDescription);
}

require_once LIB_PATH . 'osclass/model/Item.php';
require_once LIB_PATH . 'osclass/model/ItemComment.php';
require_once LIB_PATH . 'osclass/model/ItemLocation.php';
require_once LIB_PATH . 'osclass/model/ItemResource.php';
require_once LIB_PATH . 'osclass/model/ItemStats.php';
require_once LIB_PATH . 'osclass/model/User.php';
require_once LIB_PATH . 'osclass/model/Country.php';
require_once LIB_PATH . 'osclass/model/Region.php';
require_once LIB_PATH . 'osclass/model/City.php';
require_once LIB_PATH . 'osclass/model/CityArea.php';
require_once LIB_PATH . 'osclass/model/Field.php';
require_once LIB_PATH . 'osclass/model/Page.php';
require_once LIB_PATH . 'osclass/model/Log.php';

require_once LIB_PATH . 'osclass/model/CategoryStats.php';
require_once LIB_PATH . 'osclass/model/CountryStats.php';
require_once LIB_PATH . 'osclass/model/RegionStats.php';
require_once LIB_PATH . 'osclass/model/CityStats.php';

require_once LIB_PATH . 'osclass/helpers/hSecurity.php';
require_once LIB_PATH . 'osclass/helpers/hValidate.php';
require_once LIB_PATH . 'osclass/helpers/hUsers.php';
require_once LIB_PATH . 'osclass/ItemActions.php';

$mItem = new ItemActions(true);

foreach($item as $k => $v) {
    if($k=='description' || $k=='title') {
        Params::setParam($k, array(osc_current_admin_locale() => $v));
    } else {
        Params::setParam($k, $v);
    }
}

$mItem->prepareData(true);
$successItem = $mItem->add();

$successPageresult = Page::newInstance()->insert(
    array(
        's_internal_name' => $page['s_internal_name'],
        'b_indelible' => 0,
        's_meta' => json_encode('')
    ),
    array(
        osc_current_admin_locale() => array(
            's_title' => $page['s_title'],
            's_text' => $page['s_text']
        )
    )
);
}

/*
 * Create config file from scratch
 *
 * @since 1.2
 *
 * @param string $dbname Database name
 * @param string $username User of the database
 * @param string $password Password for user of the database
 * @param string $dbhost Database host
 * @param string $tableprefix Prefix for table names
 * @return mixed Error messages of the installation
 */
function create_config_file($dbname, $username, $password, $dbhost, $tableprefix) {
    $password = addslashes($password);
    $abs_url = get_absolute_url();
    $rel_url = get_relative_url();
    $config_text = <<<CONFIG
<?php
/**
 * The base MySQL settings of Osclass
 */
define('MULTISITE', 0);

/** MySQL database name for Osclass */

```

```

define( 'DB_NAME', '$dbname' );

/** MySQL database username */
define( 'DB_USER', '$username' );

/** MySQL database password */
define( 'DB_PASSWORD', '$password' );

/** MySQL hostname */
define( 'DB_HOST', '$dbhost' );

/** Database Table prefix */
define( 'DB_TABLE_PREFIX', '$tableprefix' );

define( 'REL_WEB_URL', '$rel_url' );

define( 'WEB_PATH', '$abs_url' );

CONFIG;

    file_put_contents( ABS_PATH . 'config.php', $config_text );
}

/*
 * Create config from config-sample.php file
 *
 * @since 1.2
 */
function copy_config_file( $dbname, $username, $password, $dbhost, $tableprefix ) {
    $password = addslashes( $password );
    $abs_url = get_absolute_url();
    $rel_url = get_relative_url();
    $config_sample = file( ABS_PATH . 'config-sample.php' );

    foreach ( $config_sample as $line_num => $line ) {
        switch ( substr( $line, 0, 16 ) ) {
            case "define( 'DB_NAME'":
                $config_sample[ $line_num ] = str_replace( "database_name", $dbname, $line );
                break;
            case "define( 'DB_USER'":
                $config_sample[ $line_num ] = str_replace( "'username'", "'$username'", $line );
                break;
            case "define( 'DB_PASSW'":
                $config_sample[ $line_num ] = str_replace( "'password'", "'$password'", $line );
                break;
            case "define( 'DB_HOST'":
                $config_sample[ $line_num ] = str_replace( "localhost", $dbhost, $line );
                break;
            case "define( 'DB_TABLE'":
                $config_sample[ $line_num ] = str_replace( 'oc_', $tableprefix, $line );
                break;
            case "define( 'REL_WEB'":
                $config_sample[ $line_num ] = str_replace( 'rel_here', $rel_url, $line );
                break;
            case "define( 'WEB_PATH'":
                $config_sample[ $line_num ] = str_replace( 'http://localhost', $abs_url, $line );
                break;
        }
    }

    $handle = fopen( ABS_PATH . 'config.php', 'w' );
    foreach ( $config_sample as $line ) {
        fwrite( $handle, $line );
    }
    fclose( $handle );
    chmod( ABS_PATH . 'config.php', 0666 );
}

function is_osclass_installed( ) {
    if ( !file_exists( ABS_PATH . 'config.php' ) ) {
        return false;
    }

    require_once ABS_PATH . 'config.php';

    $conn = new DBConnectionClass( osc_db_host(), osc_db_user(), osc_db_password(),
        osc_db_name() );
    $c_db = $conn->getOsclassDb();
    $comm = new DBCommandClass( $c_db );
    $rs = $comm->query( sprintf( "SELECT * FROM %st_preference WHERE s_name = '
        osclass_installed'", DB_TABLE_PREFIX ) );

    if ( $rs == false ) {
        return false;
    }

    if ( $rs->numRows() != 1 ) {
        return false;
    }

    return true;
}

function finish_installation( $password ) {
    require_once LIB_PATH . 'osclass/model/Admin.php';
    require_once LIB_PATH . 'osclass/model/Category.php';
    require_once LIB_PATH . 'osclass/model/Item.php';
    require_once LIB_PATH . 'osclass/helpers/hPlugins.php';
}

```

```

require_once LIB_PATH . 'osclass/compatibility.php';
require_once LIB_PATH . 'osclass/classes/Plugins.php';

$data = array();

$Admin = new Admin();

$Preference = Preference::newInstance();
$Preference->insert (
    array(
        's_section' => 'osclass',
        's_name' => 'osclass_installed',
        's_value' => '1',
        'e_type' => 'BOOLEAN'
    )
);

$admin = $Admin->findByPrimaryKey(1);

$data['s_email'] = $admin['s_email'];
$data['admin_user'] = $admin['s_username'];
$data['password'] = $password;

return $data;
}

/* Menus */
function display_database_config() {
    ?>
    <form action="install.php" method="post">
        <input type="hidden" name="step" value="3" />
        <h2 class="target"><?php _e('Database information'); ?></h2>
        <div class="form-table">
            <table>
                <tbody>
                    <tr>
                        <th align="left"><label for="dbhost"><?php _e('Host'); ?></label></th>
                        <td><input type="text" id="dbhost" name="dbhost" value="localhost" size="25" /></td>
                        <td class="small"><?php _e('Server name or IP where the database engine resides'); ?></td>
                    </tr>
                    <tr>
                        <th align="left"><label for="dbname"><?php _e('Database name'); ?></label></th>
                        <td><input type="text" id="dbname" name="dbname" value="osclass" size="25" /></td>
                        <td class="small"><?php _e('The name of the database you want to run Oclass in'); ?></td>
                    </tr>
                    <tr>
                        <th align="left"><label for="username"><?php _e('User Name'); ?></label></th>
                        <td><input type="text" id="username" name="username" size="25" /></td>
                        <td class="small"><?php _e('Your MySQL username'); ?></td>
                    </tr>
                    <tr>
                        <th align="left"><label for="password"><?php _e('Password'); ?></label></th>
                        <td><input type="password" id="password" name="password" value="" size="25" /></td>
                        <td class="small"><?php _e('Your MySQL password'); ?></td>
                    </tr>
                    <tr>
                        <th align="left"><label for="tableprefix"><?php _e('Table prefix'); ?></label></th>
                        <td><input type="text" id="tableprefix" name="tableprefix" value="oc_" size="25" /></td>
                        <td class="small"><?php _e('If you want to run multiple Oclass installations in a single database, change this'); ?></td>
                    </tr>
                </tbody>
            </table>
            <div id="advanced_install" class="shrink">
                <div class="text">
                    <span><?php _e('Advanced'); ?></span>
                </div>
            </div>
            <script type="text/javascript">
                $(document).ready(function() {
                    $('#advanced_install').click(function() {
                        $('#more-options').toggle();
                        if( $('#advanced_install').attr('class') == 'shrink' ) {
                            $('#advanced_install').removeClass('shrink');
                            $('#advanced_install').addClass('expanded');
                        } else {
                            $('#advanced_install').addClass('shrink');
                            $('#advanced_install').removeClass('expanded');
                        }
                    });
                });
            </script>
            <div style="clear:both;"></div>
            <table id="more-options" style="display:none;">
                <tbody>
                    <tr>
                        <th></th>
                        <td><input type="checkbox" id="createdb" name="createdb" onclick="db_admin();" value="1" /><label for="createdb"><?php _e('Create DB'); ?></td>
                    </tr>
                </tbody>
            </table>

```

```

        label></td>
        <td class="small"><?php _e('Check here if the database is not created and
        you want to create it now'); ?></td>
    </tr>
    <tr id="admin_username_row">
        <th align="left"><label for="admin_username"><?php _e('DB admin username')
        ; ?></label></th>
        <td><input type="text" id="admin_username" name="admin_username" size="25"
        disabled="disabled" /></td>
    </tr>
    <tr id="admin_password_row">
        <th align="left"><label for="admin_password"><?php _e('DB admin password')
        ; ?></label></th>
        <td><input type="password" id="admin_password" name="admin_password" value
        ="" size="25" disabled="disabled" /></td>
    </tr>
</tbody>
</table>
</div>
<div class="clear"></div>
<p class="margin20">
    <input type="submit" class="button" name="submit" value="Next" />
</p>
<div class="clear"></div>
</form>
<?php
}

function display_target() {
    $country_list = osc_file_get_contents('http://geo.osclass.org/newgeo.services.php?action=
    countries');
    $country_list = json_decode(substr($country_list, 1, strlen($country_list)-2), true);
    $region_list = array();

    $country_ip = '';
    if(preg_match('!([a-z]{2})-([A-Z]{2})|!', @$_SERVER['HTTP_ACCEPT_LANGUAGE'], $match)) {
        $country_ip = $match[2];
        $region_list = osc_file_get_contents('http://geo.osclass.org/newgeo.services.php?
        action=regions&country='.$match[2]);
        $region_list = json_decode(substr($region_list, 1, strlen($region_list)-2), true);
    }

    if(!isset($country_list[0]) || !isset($country_list[0]['s_name'])) {
        $internet_error = true;
    }
}

?>
<form id="target_form" name="target_form" action="#" method="post" onsubmit="return false;">
<h2 class="target"><?php _e('Information needed'); ?></h2>
<div class="form-table">
    <h2 class="title"><?php _e('Admin user'); ?></h2>
    <table class="admin-user">
        <tbody>
            <tr>
                <th><label for="admin_user"><?php _e('Username'); ?></label></th>
                <td>
                    <input size="25" id="admin_user" name="s_name" type="text" value="
                    admin" />
                </td>
                <td>
                    <span id="admin-user-error" class="error" aria-hidden="true" style="
                    display:none;"><?php _e('Admin user is required'); ?></span>
                </td>
            </tr>
            <tr>
                <th><label for="s_passwd"><?php _e('Password'); ?></label></th>
                <td>
                    <input size="25" class="password_test" name="s_passwd" id="s_passwd"
                    type="password" value="" />
                </td>
            </tr>
        </tbody>
    </table>
    <div class="admin-user">
        <?php _e('A password will be automatically generated for you if you leave this
        blank. '); ?>
        " alt="" />
    </div>
    <h2 class="title"><?php _e('Contact information'); ?></h2>
    <table class="contact-info">
        <tbody>
            <tr>
                <th><label for="webtitle"><?php _e('Web title '); ?></label></th>
                <td><input type="text" id="webtitle" name="webtitle" size="25" /></td>
            </tr>
            <tr>
                <th><label for="email"><?php _e('Contact e-mail'); ?></label></th>
                <td><input type="text" id="email" name="email" size="25" /></td>
                <td><span id="email-error" class="error" style="display:none;"><?php _e('
                Put your e-mail here '); ?></span></td>
            </tr>
        </tbody>
    </table>

```



```

        </tbody>
</table>
<h2 class="title"><?php _e('Location'); ?></h2>
<p class="space-left -25 left no-bottom"><?php _e('Choose countries/cities where your
target users are located'); ?></p>
<div id="location-question" class="left question">
    " alt="" />
</div>
<div class="clear"></div>
<div id="location">
    <?php if(!$internet_error) { ?>
    <input type="hidden" id="skip-location-input" name="skip-location-input" value="0"
    />
    <input type="hidden" id="country-input" name="country-input" value="" />
    <input type="hidden" id="region-input" name="region-input" value="" />
    <input type="hidden" id="city-input" name="city-input" value="" />
    <div id="country-box">
        <select name="country_select" id="country_select" >
            <option value="skip"><?php _e("Skip location"); ?></option>
            <!-- <option value="all"><?php _e("International"); ?></option> -->
            <?php foreach($country_list as $c) { ?>
                <option value="<?php echo $c['code']; ?>" <?php if($c['code']==
                    $country_ip) { echo 'selected="selected"'; } ?><?php echo $c['
                    s_name']; ?></option>
            <?php }; ?>
        </select>
        <select name="region_select" id="region_select" style="display: none;">
            <option value="all"><?php _e("All regions"); ?></option>
        </select>
        <select name="city_select" id="city_select" style="display: none;">
            <option value="all"><?php _e("All cities"); ?></option>
        </select>
        <div id="no-region-text" aria-hidden="true" style="display: none;"><?php _e("
            There are no regions available for this country"); ?></div>
        <div id="no-city-text" aria-hidden="true" style="display: none;"><?php _e("
            There are no cities available for this region"); ?></div>
    </div>
    <?php } else { ?>
    <div id="location-error">
        <?php _e('No internet connection. You can continue the installation and insert
countries later. '); ?>
        <input type="hidden" id="skip-location-input" name="skip-location-input" value
        ="1" />
    </div>
    <?php }; ?>
</div>
</div>
<div class="clear"></div>
<p class="margin20">
    <a href="#" class="button" onclick="validate_form();">Next</a>
</p>
<div class="clear"></div>
</form>
<div id="lightbox" style="display:none;">
    <div class="center">
        " title="" />
    </div>
</div>
<?php
}

function display_database_error($error , $step) {
    ?>
    <h2 class="target"><?php _e('Error'); ?></h2>
    <p class="bottom space-left -10">
        <?php echo $error['error']; ?>
    </p>
    <a href="<?php echo get_absolute_url(); ?>oc-includes/osclass/install.php?step=<?php echo
        $step; ?>" class="button"><?php _e('Go back'); ?></a>
    <div class="clear bottom"></div>
    <?php
}

function ping_search_engines($bool){
    $mPreference = Preference::newInstance();
    if($bool == 1){
        $mPreference->insert (
            array(
                's_section' => 'osclass'
                , 's_name' => 'ping_search_engines'
                , 's_value' => '1'
                , 'e_type' => 'BOOLEAN'
            )
        );
        // GOOGLE
        osc_doRequest( 'http://www.google.com/webmasters/sitemaps/ping?sitemap='.urlencode(
            osc_search_url(array('sFeed' => 'rss') ) ), array());
        // BING
    }
}

```

```

osc_doRequest( 'http://www.bing.com/webmaster/ping.aspx?siteMap='.urlencode(
    osc_search_url(array('sFeed' => 'rss') ) ), array());
// YAHOO!
osc_doRequest( 'http://search.yahooapis.com/SiteExplorerService/V1/ping?sitemap='.
    urlencode( osc_search_url(array('sFeed' => 'rss') ) ), array());
} else {
    $mPreference->insert (
        array(
            's_section' => 'osclass '
            , 's_name' => 'ping_search_engines '
            , 's_value' => '0'
            , 'e_type' => 'BOOLEAN'
        )
    );
}
}
function display_finish($password) {
    $data = finish_installation( $password );
?>
    <?php if(Params::getParam('error_location') == 1) { ?>
    <script type="text/javascript">
        setTimeout (function(){
            $('error_location').fadeOut('slow');
        }, 2500);
    </script>
    <div class="error_location">
        <?php _e('The selected location could not be installed'); ?>
    </div>
    <?php } ?>
<h2 class="target"><?php _e('Congratulations!');?></h2>
<p class="space-left-10"><?php _e("Osclass has been installed. Were you expecting more steps?
    Sorry to disappoint!");?></p>
<p class="space-left-10"><?php echo sprintf(_e('An e-mail with the password for oc-admin has
    been sent to: %s'), $data['s_email']);?></p>
<div style="clear:both;"></div>
<div class="form-table finish">
    <table>
        <tbody>
            <tr>
                <th><span class="label-like"><?php _e('Username');?></span></th>
                <td>
                    <div class="s_name">
                        <span style="float:left;" ><?php echo $data['admin_user']; ?></span>
                    </div>
                </td>
            </tr>
            <tr>
                <th><span class="label-like"><?php _e('Password');?></span></th>
                <td>
                    <div class="s_passwd">
                        <span style="float:left;" ><?php echo osc_esc_html($data['password']);
                            ?></span>
                    </div>
                </td>
            </tr>
        </tbody>
    </table>
</div>
<p class="margin20">
    <a target="_blank" href="<?php echo get_absolute_url() ?>oc-admin/index.php" class="button
        "><?php _e('Finish and go to the administration panel');?></a>
</p>
<?php
}
?>

<?php

error_reporting(E_ERROR | E_CORE_ERROR | E_COMPILE_ERROR | E_PARSE);

define( 'ABS_PATH', dirname(dirname(dirname(__FILE__))) . '/' );
define( 'LIB_PATH', ABS_PATH . 'oc-includes/' );

require_once ABS_PATH . 'config.php';

require_once LIB_PATH . 'osclass/classes/database/DBConnectionClass.php';
require_once LIB_PATH . 'osclass/classes/database/DBCommandClass.php';
require_once LIB_PATH . 'osclass/classes/database/DBRecordsetClass.php';
require_once LIB_PATH . 'osclass/classes/database/DAO.php';
require_once LIB_PATH . 'osclass/Logger/Logger.php';
require_once LIB_PATH . 'osclass/Logger/LogDatabase.php';
require_once LIB_PATH . 'osclass/Logger/LogOsclass.php';
require_once LIB_PATH . 'osclass/core/Session.php';
require_once LIB_PATH . 'osclass/core/Params.php';
require_once LIB_PATH . 'osclass/model/Preference.php';
require_once LIB_PATH . 'osclass/helpers/hDatabaseInfo.php';
require_once LIB_PATH . 'osclass/helpers/hDefines.php';
require_once LIB_PATH . 'osclass/helpers/hErrors.php';
require_once LIB_PATH . 'osclass/helpers/hLocale.php';
require_once LIB_PATH . 'osclass/helpers/hPreference.php';
require_once LIB_PATH . 'osclass/helpers/hPlugins.php';
require_once LIB_PATH . 'osclass/helpers/hTranslations.php';
require_once LIB_PATH . 'osclass/compatibility.php';
require_once LIB_PATH . 'osclass/default-constants.php';
require_once LIB_PATH . 'osclass/formatting.php';
require_once LIB_PATH . 'osclass/install-functions.php';
require_once LIB_PATH . 'osclass/utills.php';
require_once LIB_PATH . 'osclass/core/Translation.php';
require_once LIB_PATH . 'osclass/classes/Plugins.php';

```

```

if( is_osclass_installed() ) {
    die();
}

$json_message = array();
$json_message['status'] = '200';

$result = basic_info();
$json_message['email_status'] = $result['email_status'];
$json_message['password'] = $result['s_password'];

if($_POST['skip-location-input']==0 && $_POST['country-input']!='skip') {
    $msg = install_locations();
    $json_message['status'] = $msg;
}

echo json_encode($json_message);

function basic_info() {
    require_once LIB_PATH . 'osclass/model/Admin.php';
    require_once LIB_PATH . 'osclass/helpers/hSecurity.php';

    $admin = Params::getParam('s_name');
    if( $admin == '' ) {
        $admin = 'admin';
    }

    $password = Params::getParam('s_passwd', false, false);
    if( $password == '' ) {
        $password = osc_genRandomPassword();
    }

    Admin::newInstance()->insert(
        array(
            's_name' => 'Administrator'
            , 's_username' => $admin
            , 's_password' => sha1($password)
            , 's_email' => Params::getParam('email')
        )
    );

    $mPreference = Preference::newInstance();
    $mPreference->insert(
        array(
            's_section' => 'osclass'
            , 's_name' => 'pageTitle'
            , 's_value' => Params::getParam('webtitle')
            , 'e_type' => 'STRING'
        )
    );

    $mPreference->insert(
        array(
            's_section' => 'osclass'
            , 's_name' => 'contactEmail'
            , 's_value' => Params::getParam('email')
            , 'e_type' => 'STRING'
        )
    );

    $body = sprintf('--(Hi %s,)', Params::getParam('webtitle'))."<br/><br/>";
    $body .= sprintf('--(Your Osclass installation at %s is up and running. You can access the
administration panel with these details:)', WEB_PATH) . "<br/>";
    $body .= '<ul>';
    $body .= '<li>'.sprintf('--(username: %s)', $admin).'</li>';
    $body .= '<li>'.sprintf('--(password: %s)', $password).'</li>';
    $body .= '</ul>';
    $body .= sprintf('--(Remember that for any doubts you might have you can consult our <a
href="%1$s">documentation</a>, <a href="%2$s">forum</a> or <a href="%3$s">blog</a>.)',
'http://doc.osclass.org/', 'http://forums.osclass.org/', 'http://blog.osclass.org/');
    $body .= sprintf(' . --(Osclass doesn t run any developments but we can put you in
touch with third party developers through a Premium Support. And hey, if you would
like to contribute to Osclass - learn how <a href="%1$s">here</a>!)', 'http://blog.
osclass.org/2012/11/22/how-to-collaborate-to-osclass/') . "<br/><br/>";
    $body .= --('Cheers,') . "<br/>";
    $body .= --('The <a href="http://osclass.org/">Osclass</a> team');

    $sitename = strtolower( $_SERVER['SERVER_NAME'] );
    if ( substr( $sitename, 0, 4 ) == 'www.' ) {
        $sitename = substr( $sitename, 4 );
    }

    try{
        require_once LIB_PATH . 'phpmailer/class.phpmailer.php';
        $mail = new PHPMailer(true);
        $mail->CharSet = "utf-8";
        $mail->Host = "localhost";
        $mail->From = 'osclass@' . $sitename;
        $mail->FromName = 'Osclass';
        $mail->Subject = 'Osclass successfully installed!';
        $mail->AddAddress(Params::getParam('email'), 'Osclass administrator');
        $mail->Body = $body;
        $mail->AltBody = $body;
        if( !$mail->Send() ) {
            return array('email_status' => Params::getParam('email') . "<br/>" . $mail->
                ErrorInfo, 's_password' => $password );
        }
    }
}

```

```

        return array('email-status' => '', 's-password' => $password );
    } catch (phpmailerException $exception) {
        return array('email-status' => Params::getParam('email') . "<br>" . $exception->
            errorMessage(), 's-password' => $password );
    }
}

function install_locations ( ) {

    $country = Params::getParam("country-input");
    $region = Params::getParam("region-input");
    $city = Params::getParam("city-input");

    if($country!='all') {
        if($region!='all') {
            if($city!='all') {
                $sql = 'action=city&term='.urlencode($city);
            } else {
                $sql = 'action=region&term='.urlencode($region);
            }
        } else {
            $sql = 'action=country&term='.urlencode($country);
        }
    } else {
        $sql = 'action=country&term=all';
    }

    $data_sql = osc_file_get_contents('http://geo.osclass.org/newgeo.download.php?'.$sql.'&
        install=true');

    $conn = DBConnectionClass::newInstance();
    $c_db = $conn->getOscDb();
    $comm = new DBCommandClass($c_db);
    $comm->query("SET FOREIGN_KEY_CHECKS = 0");
    $imported = $comm->importSQL($data_sql);
    $comm->query("SET FOREIGN_KEY_CHECKS = 1");
    return $imported;
}

?>

<?php if ( ! defined('ABS_PATH')) exit('ABS_PATH is not loaded. Direct access is not allowed
. ');

/**
 * Osclass software for creating and publishing online classified advertising platforms
 *
 * Copyright (C) 2012 OSCLASS
 *
 * This program is free software: you can redistribute it and/or modify it under the terms
 * of the GNU Affero General Public License as published by the Free Software Foundation,
 * either version 3 of the License, or (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful, but WITHOUT ANY
 * WARRANTY;
 * without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
 * PURPOSE.
 * See the GNU Affero General Public License for more details.
 *
 * You should have received a copy of the GNU Affero General Public
 * License along with this program. If not, see <http://www.gnu.org/licenses/>.
 */

Class ItemActions
{
    private $manager = null;
    var $is_admin;
    var $data;

    function __construct($is_admin) {
        $this->is_admin = $is_admin;
        $this->manager = Item::newInstance();
    }

    private function _akismet_text( $title, $description, $author, $email )
    {
        $spam = false;
        foreach($title as $k => $data){
            $_title = $title[$k];
            $description = $description[$k];
            $content = $_title . ' ' . $description;
            if (osc_akismet_key()) {
                require_once LIB_PATH . 'Akismet.class.php';
                $akismet = new Akismet(osc_base_url(), osc_akismet_key());

                $akismet->setCommentContent($content);
                $akismet->setCommentAuthor($author);
                $akismet->setCommentAuthorEmail($email);
                $akismet->setUserIP( get_ip() );

                $status = '';
                $status = $akismet->isCommentSpam() ? 'SPAM' : $status;
                if($status == 'SPAM') {
                    $spam = true;
                    break;
                }
            }
        }
    }
}

```

```

    return $spam;
}

/**
 * @return boolean
 */
public function add()
{
    $aItem      = $this->data;
    $is_spam    = 0;
    $enabled    = 1;
    $code       = osc_genRandomPassword();
    $flash_error = '';

    // Requires email validation?
    $has_to_validate = (osc_moderate_items() != -1) ? true : false;

    // Check status
    $active = $aItem['active'];

    // Sanitize
    foreach(@$aItem['title'] as $key=>$value) {
        $aItem['title'][$key] = strip_tags( trim ( $value ) );
    }

    $aItem['price'] = !is_null($aItem['price']) ? strip_tags( trim( $aItem['price'] ) ) : $aItem['price'];
    $contactName    = strip_tags( trim( $aItem['contactName'] ) );
    $contactEmail   = strip_tags( trim( $aItem['contactEmail'] ) );
    $aItem['cityArea'] = osc_sanitize_name( strip_tags( trim( $aItem['cityArea'] ) ) );
    $aItem['address'] = osc_sanitize_name( strip_tags( trim( $aItem['address'] ) ) );

    // Anonymous
    $contactName = (osc_validate_text($contactName,3))? $contactName : _("Anonymous");

    // Validate
    if ( !$this->checkAllowedExt($aItem['photos']) ) {
        $flash_error .= _("Image with an incorrect extension.") . PHP_EOL;
    }
    if ( !$this->checkSize($aItem['photos']) ) {
        $flash_error .= _("Image is too big. Max. size") . osc_max_size_kb() . " Kb" . PHP_EOL;
    }

    $title_message = '';
    foreach(@$aItem['title'] as $key => $value) {
        if( osc_validate_text($value, 1) && osc_validate_max($value, 100) ) {
            $title_message = '';
            break;
        }

        $title_message .=
            (!osc_validate_text($value, 1) ? _("Title too short.") . PHP_EOL : '' ) .
            (!osc_validate_max($value, 100) ? _("Title too long.") . PHP_EOL : '' );
    }
    $flash_error .= $title_message;

    $desc_message = '';
    foreach(@$aItem['description'] as $key => $value) {
        if( osc_validate_text($value, 3) && osc_validate_max($value, 5000) ) {
            $desc_message = '';
            break;
        }

        $desc_message .=
            (!osc_validate_text($value, 3) ? _("Description too short.") . PHP_EOL : '' ) .
            (!osc_validate_max($value, 5000) ? _("Description too long.") . PHP_EOL : '' );
    }
    $flash_error .= $desc_message;

    // akismet check spam ...
    if( $this->_akismet_text( $aItem['title'], $aItem['description'], $contactName, $contactEmail ) ) {
        $is_spam = 1;
    }

    $flash_error .=
        ((!osc_validate_category($aItem['catId'])) ? _("Category invalid.") . PHP_EOL : '' ) .
        ((!osc_validate_number($aItem['price'])) ? _("Price must be a number.") . PHP_EOL : '' ) .
        ((!osc_validate_max(number_format($aItem['price'],0,'',''), 15)) ? _("Price too long.") . PHP_EOL : '' ) .
        ((!is_null($aItem['price']) && (int)$aItem['price']<0) ? _("Price must be positive number.") . PHP_EOL : '' ) .
        ((!osc_validate_max($contactName, 35)) ? _("Name too long.") . PHP_EOL : '' ) .
        ((!osc_validate_email($contactEmail)) ? _("Email invalid.") . PHP_EOL : '' ) .
        ((!osc_validate_text($aItem['countryName'], 2, false)) ? _("Country too short.") . PHP_EOL : '' ) .
        ((!osc_validate_max($aItem['countryName'], 50)) ? _("Country too long.") . PHP_EOL : '' ) .
        ((!osc_validate_text($aItem['regionName'], 2, false)) ? _("Region too short

```

```

        .") . PHP_EOL : '' ) .
        ((!osc_validate_max($aItem['regionName'], 50)) ? _m("Region too long.") .
        . PHP_EOL : '' ) .
        ((!osc_validate_text($aItem['cityName'], 2, false)) ? _m("City too short.") .
        . PHP_EOL : '' ) .
        ((!osc_validate_max($aItem['cityName'], 50)) ? _m("City too long.") . PHP_EOL
        : '' ) .
        ((!osc_validate_text($aItem['cityArea'], 2, false)) ? _m("Municipality too
        short.") . PHP_EOL : '' ) .
        ((!osc_validate_max($aItem['cityArea'], 50)) ? _m("Municipality too long.") .
        . PHP_EOL : '' ) .
        ((!osc_validate_text($aItem['address'], 3, false)) ? _m("Address too short.")
        . PHP_EOL : '' ) .
        ((!osc_validate_max($aItem['address'], 100)) ? _m("Address too long.") .
        . PHP_EOL : '' ) .
        (((time() - Session::newInstance()->_get('last_submit_item')) <
        osc_items_wait_time()) && !$this->is_admin) ? _m("Too fast. You should
        wait a little to publish your ad.") . PHP_EOL : '' );

    $_meta = Field::newInstance()->findByCategory($aItem['catId']);
    $meta = Params::getParam("meta");

    foreach($_meta as $_m) {
        $meta[$_m['pk_i_id']] = (isset($meta[$_m['pk_i_id']]))?$meta[$_m['pk_i_id']]:'';
    }

    if($meta!='' && count($meta)>0) {
        $mField = Field::newInstance();
        foreach($meta as $k => $v) {
            if($v=='') {
                $field = $mField->findByPrimaryKey($k);
                if($field['b_required']==1) {
                    $flash_error .= sprintf(_m("%s field is required."), $field['s_name']) . PHP_EOL;
                }
            }
        }
    }
}

// hook pre add or edit
// DEPRECATED: pre_item_post will be removed in 3.4
osc_run_hook('pre_item_post');
osc_run_hook('pre_item_add', $aItem);

// Handle error
if ($flash_error) {
    $success = $flash_error;
} else {
    if($aItem['price']!= '') {
        $aItem['currency'] = $aItem['currency'];
    } else {
        $aItem['currency'] = NULL;
    }
}

$this->manager->insert(array(
    'fk_i_user_id'           => $aItem['userId'],
    'dt_pub_date'           => date('Y-m-d H:i:s'),
    'fk_i_category_id'     => $aItem['catId'],
    'i_price'               => $aItem['price'],
    'fk_c_currency_code'   => $aItem['currency'],
    's_contact_name'       => $contactName,
    's_contact_email'      => $contactEmail,
    's_secret'              => $code,
    'b_active'              => ($active=='ACTIVE'?1:0),
    'b_enabled'            => $enabled,
    'b_show_email'         => $aItem['showEmail'],
    'b_spam'                => $is_spam,
    's_ip'                  => $aItem['s_ip']
));

if(!$this->is_admin) {
    // Track spam delay: Session
    Session::newInstance()->set('last_submit_item', time());
    // Track spam delay: Cookie
    Cookie::newInstance()->set_expires( osc_time_cookie() );
    Cookie::newInstance()->push('last_submit_item', time());
    Cookie::newInstance()->set();
}

$itemId = $this->manager->dao->insertedId();
Log::newInstance()->insertLog('item', 'add', $itemId, current(array_values(
    $aItem['title'])), $this->is_admin?'admin':'user', $this->is_admin?
    osc_logged_admin_id():osc_logged_user_id());

Params::setParam('itemId', $itemId);

// INSERT title and description locales
$this->insertItemLocales('ADD', $aItem['title'], $aItem['description'],
    $itemId );

$location = array(
    'fk_i_item_id'           => $itemId,
    'fk_c_country_code'     => $aItem['countryId'],
    's_country'             => $aItem['countryName'],
    'fk_i_region_id'       => $aItem['regionId'],
    's_region'              => $aItem['regionName'],
    'fk_i_city_id'         => $aItem['cityId'],
    's_city'                => $aItem['cityName'],

```

```

        's_city_area'      => $aItem['cityArea'],
        's_address'       => $aItem['address'],
        'd_coord_lat'     => $aItem['d_coord_lat'],
        'd_coord_long'    => $aItem['d_coord_long']
    );

    $locationManager = ItemLocation::newInstance();
    $locationManager->insert($location);

    $this->uploadItemResources( $aItem['photos'] , $itemId);

    // update dt_expiration at t_item
    $dt_expiration = Item::newInstance()->updateExpirationDate($itemId, $aItem['
    dt_expiration']);

    /**
     * META FIELDS
     */
    if($meta!='' && count($meta)>0) {
        $mField = Field::newInstance();
        foreach($meta as $k => $v) {
            // if dateinterval
            if(is_array($v) && !isset($v['from']) && !isset($v['to']) ) {
                $v = implode(', ', $v);
            }
            $mField->replace($itemId, $k, $v);
        }
    }

    // We need at least one record in t_item_stats
    $mStats = new ItemStats();
    $mStats->emptyRow($itemId);

    $item = $this->manager->findByPrimaryKey($itemId);
    $aItem['item'] = $item;

    Session::newInstance()->_set('last_publish_time', time());
    if(!$this->is_admin) {
        $this->sendEmails($aItem);
    }

    if($active=='INACTIVE') {
        $success = 1;
    } else {
        $aAux = array(
            'fk_i_user_id'      => $aItem['userId'],
            'fk_i_category_id' => $aItem['catId'],
            'fk_c_country_code' => $location['fk_c_country_code'],
            'fk_i_region_id'   => $location['fk_i_region_id'],
            'fk_i_city_id'     => $location['fk_i_city_id']
        );
        // if is_spam not increase stats
        if($is_spam == 0) {
            $this->_increaseStats($aAux);
        }
        $success = 2;
    }

    // THIS HOOK IS FINE, YAY!
    osc_run_hook('posted_item', $item);
}
return $success;
}

function edit() {
    $aItem = $this->data;
    $flash_error = '';

    // Sanitize
    foreach(@$aItem['title'] as $key=>$value) {
        $aItem['title'][$key] = strip_tags(trim($value));
    }

    $aItem['price'] = !is_null($aItem['price']) ? strip_tags(trim($aItem['price'])) : $aItem['price'];
    $aItem['cityArea'] = osc_sanitize_name(strip_tags(trim($aItem['cityArea'])));
    $aItem['address'] = osc_sanitize_name(strip_tags(trim($aItem['address'])));

    // Validate
    if (!$this->checkAllowedExt($aItem['photos']) ) {
        $flash_error .= _("Image with an incorrect extension.") . PHP_EOL;
    }
    if (!$this->checkSize($aItem['photos']) ) {
        $flash_error .= _("Image is too big. Max. size") . osc_max_size_kb() . " Kb"
        . PHP_EOL;
    }

    $title_message = '';
    $std_message = '';
    foreach(@$aItem['title'] as $key => $value) {
        if( osc_validate_text($value, 1) && osc_validate_max($value, 100) ) {
            $std_message = '';
            break;
        }
    }

    $std_message .=

```

```

        (!osc_validate_text($value, 1) ? _m("Title too short.") . PHP_EOL : '' ) .
        (!osc_validate_max($value, 100) ? _m("Title too long.") . PHP_EOL : '' );
    }
    $flash_error .= $std_message;

    $desc_message = '';
    foreach(@$aItem['description'] as $key => $value) {
        if( osc_validate_text($value, 3) && osc_validate_max($value, 5000) ) {
            $desc_message = '';
            break;
        }

        $desc_message .=
            (!osc_validate_text($value, 3) ? _m("Description too short.") . PHP_EOL :
            '' ) .
            (!osc_validate_max($value, 5000) ? _m("Description too long.") . PHP_EOL :
            '' );
    }
    $flash_error .= $desc_message;

    $flash_error .=
        (!osc_validate_category($aItem['catId']) ? _m("Category invalid.") . PHP_EOL
        : '' ) .
        (!osc_validate_number($aItem['price']) ? _m("Price must be a number.") .
        PHP_EOL : '' ) .
        (!osc_validate_max(number_format($aItem['price'],0,'',''), 15) ? _m("Price
        too long.") . PHP_EOL : '' ) .
        (!is_null($aItem['price']) && (int)$aItem['price']<0) ? _m("Price must be
        positive number.") . PHP_EOL : '' ) .
        (!osc_validate_text($aItem['countryName'], 3, false)) ? _m("Country too short
        ") . PHP_EOL : '' ) .
        (!osc_validate_max($aItem['countryName'], 50) ? _m("Country too long.") .
        PHP_EOL : '' ) .
        (!osc_validate_text($aItem['regionName'], 2, false)) ? _m("Region too short
        ") . PHP_EOL : '' ) .
        (!osc_validate_max($aItem['regionName'], 50) ? _m("Region too long.") .
        PHP_EOL : '' ) .
        (!osc_validate_text($aItem['cityName'], 2, false)) ? _m("City too short.") .
        PHP_EOL : '' ) .
        (!osc_validate_max($aItem['cityName'], 50) ? _m("City too long.") . PHP_EOL
        : '' ) .
        (!osc_validate_text($aItem['cityArea'], 3, false)) ? _m("Municipality too
        short.") . PHP_EOL : '' ) .
        (!osc_validate_max($aItem['cityArea'], 50) ? _m("Municipality too long.") .
        PHP_EOL : '' ) .
        (!osc_validate_text($aItem['address'], 3, false)) ? _m("Address too short.") .
        PHP_EOL : '' ) .
        (!osc_validate_max($aItem['address'], 100) ? _m("Address too long.") .
        PHP_EOL : '' );

    $_meta = Field::newInstance()->findByCategory($aItem['catId']);
    $meta = Params::getParam("meta");
    foreach($meta as $_m) {
        $meta[$_m['pk_i_id']] = (isset($meta[$_m['pk_i_id']]))?$meta[$_m['pk_i_id']]:'';
    }
    if($meta!='' && count($meta)>0) {
        $mField = Field::newInstance();
        foreach($meta as $k => $v) {
            if($v=='') {
                $field = $mField->findByPrimaryKey($k);
                if($field['b_required']==1) {
                    $flash_error .= sprintf(_m("%s field is required."), $field['
                    s_name']) . PHP_EOL;
                }
            }
        }
    }
};

// hook pre add or edit
// DEPRECATED : preitem.pspot will be removed in 3.4
osc_run_hook('pre_item_post');
osc_run_hook('pre_item_edit', $aItem);

// Handle error
if ($flash_error) {
    $success = $flash_error;
} else {
    $location = array(
        'fk_c_country_code' => $aItem['countryId'],
        's_country' => $aItem['countryName'],
        'fk_i_region_id' => $aItem['regionId'],
        's_region' => $aItem['regionName'],
        'fk_i_city_id' => $aItem['cityId'],
        's_city' => $aItem['cityName'],
        's_city_area' => $aItem['cityArea'],
        's_address' => $aItem['address'],
        'd_coord_lat' => $aItem['d.coord.lat'],
        'd_coord.long' => $aItem['d.coord.long']
    );

    $locationManager = ItemLocation::newInstance();
    $old_item_location = $locationManager->findByPrimaryKey($aItem['idItem']);

    $locationManager->update( $location, array( 'fk_i_item_id' => $aItem['idItem']
    ) );

    $old_item = $this->manager->findByPrimaryKey( $aItem['idItem'] );

```



```

    if($aItem['userId'] != '') {
        $user = User::newInstance()->findByPrimaryKey( $aItem['userId'] );
        $aItem['userId'] = $aItem['userId'];
        $aItem['contactName'] = $user['s_name'];
        $aItem['contactEmail'] = $user['s_email'];
    } else {
        $aItem['userId'] = NULL;
    }

    if($aItem['price']!= '') {
        $aItem['currency'] = $aItem['currency'];
    } else {
        $aItem['currency'] = NULL;
    }

    $aUpdate = array(
        'dt_mod_date' => date('Y-m-d H:i:s')
        , 'fk_i_category_id' => $aItem['catId']
        , 'i_price' => $aItem['price']
        , 'fk_c_currency_code' => $aItem['currency']
    );

    // only can change the user if you're an admin
    if( $this->is_admin ) {
        $aUpdate['fk_i_user_id'] = $aItem['userId'];
        $aUpdate['s_contact_name'] = $aItem['contactName'];
        $aUpdate['s_contact_email'] = $aItem['contactEmail'];
    } else {
        $aUpdate['s_ip'] = $aItem['s_ip'];
    }

    $result = $this->manager->update( $aUpdate, array('pk_i_id' => $aItem['idItem']
        , 's_secret' => $aItem['secret']
    ));

    // UPDATE title and description locales
    $this->insertItemLocales( 'EDIT', $aItem['title'], $aItem['description'],
        $aItem['idItem'] );
    // UPLOAD item resources
    $this->uploadItemResources( $aItem['photos'], $aItem['idItem'] );

    Log::newInstance()->insertLog('item', 'edit', $aItem['idItem'], current(
        array_values($aItem['title'])), $this->is_admin?'admin':'user', $this->
        is_admin?osc_logged_admin_id():osc_logged_user_id());

    /**
     * META FIELDS
     */
    if($meta!='' && count($meta)>0) {
        $mField = Field::newInstance();
        foreach($meta as $k => $v) {
            // if dateinterval
            if( is_array($v) && !isset($v['from']) && !isset($v['to']) ) {
                $v = implode(', ', $v);
            }
            $mField->replace($aItem['idItem'], $k, $v);
        }
    }

    $oldIsExpired = osc_isExpired($old_item['dt_expiration']);
    $dt_expiration = Item::newInstance()->updateExpirationDate($aItem['idItem'],
        $aItem['dt_expiration'], false);
    if($dt_expiration==false) {
        $dt_expiration = $old_item['dt_expiration'];
        $aItem['dt_expiration'] = $old_item['dt_expiration'];
    }
    $newIsExpired = osc_isExpired($dt_expiration);

    // Recalculate stats related with items
    $this->updateStats($result, $old_item, $oldIsExpired, $old_item_location,
        $aItem, $newIsExpired, $location);

    unset($old_item);

    // THIS HOOK IS FINE, YAY!
    osc_run_hook('edited_item', Item::newInstance()->findByPrimaryKey($aItem['idItem']));
    $success = $result;
}

return $success;
}

/**
 * Increment or decrement stats related with items.
 *
 * User item stats, Category item stats,
 * country item stats, region item stats, city item stats
 *
 * @param type $result
 * @param type $old_item
 * @param type $oldIsExpired
 * @param type $old_item_location
 * @param type $aItem
 * @param type $newIsExpired
 * @param type $location
 */
private function _updateStats($result, $old_item, $oldIsExpired, $old_item_location,
    $aItem, $newIsExpired, $location)
{

```

```

if($result==1 && $old_item['b_enabled']==1 && $old_item['b_active']==1 &&
    $old_item['b_spam']==0) {
    // if old item is expired and new item is not expired.
    if($oldIsExpired && !$newIsExpired) {
        // increment new item stats (user, category, location_stats)
        if( is_numeric($aItem['userId']) ) {
            User::newInstance()->increaseNumItems( $aItem['userId'] );
        }
        CategoryStats::newInstance()->increaseNumItems($aItem['catId']);
        CountryStats::newInstance()->increaseNumItems($location['fk_c_country_code']);
        RegionStats::newInstance()->increaseNumItems($location['fk_i_region_id']);
        CityStats::newInstance()->increaseNumItems($location['fk_i_city_id']);
    }
    // if old is not expired and new is expired
    if(!$oldIsExpired && $newIsExpired) {
        // decrement new item stats (user, category, location_stats)
        if( is_numeric($old_item['fk_i_user_id']) ) {
            User::newInstance()->decreaseNumItems( $old_item['fk_i_user_id'] );
        }
        CategoryStats::newInstance()->decreaseNumItems($aItem['catId']);
        CountryStats::newInstance()->decreaseNumItems($location['fk_c_country_code']);
        RegionStats::newInstance()->decreaseNumItems($location['fk_i_region_id']);
        CityStats::newInstance()->decreaseNumItems($location['fk_i_city_id']);
    }
    // if old item is not expired and new item is not expired
    if(!$oldIsExpired && !$newIsExpired) {
        // Update user stats - if old user diferent to actual user, update user
        stats
        if($old_item['fk_i_user_id'] != $aItem['userId']) {
            if( is_numeric($old_item['fk_i_user_id']) ) {
                User::newInstance()->decreaseNumItems( $old_item['fk_i_user_id'] );
            }
            if( is_numeric($aItem['userId']) ) {
                User::newInstance()->increaseNumItems( $aItem['userId'] );
            }
        }
        // Update category numbers
        if($old_item['fk_i_category_id'] != $aItem['catId']) {
            CategoryStats::newInstance()->increaseNumItems($aItem['catId']);
            CategoryStats::newInstance()->decreaseNumItems($old_item['fk_i_category_id']);
        }
        // Update location stats
        if($old_item_location['fk_c_country_code'] != $location['fk_c_country_code']) {
            CountryStats::newInstance()->decreaseNumItems($old_item_location['fk_c_country_code']);
            CountryStats::newInstance()->increaseNumItems($location['fk_c_country_code']);
        }
        if($old_item_location['fk_i_region_id'] != $location['fk_i_region_id']) {
            RegionStats::newInstance()->decreaseNumItems($old_item_location['fk_i_region_id']);
            RegionStats::newInstance()->increaseNumItems($location['fk_i_region_id']);
        }
        if($old_item_location['fk_i_city_id'] != $location['fk_i_city_id']) {
            CityStats::newInstance()->decreaseNumItems($old_item_location['fk_i_city_id']);
            CityStats::newInstance()->increaseNumItems($location['fk_i_city_id']);
        }
    }
    // if old and new items are expired [nothing to do]
    // if($oldIsExpired && $newIsExpired) { }
}
}

/**
 * Activates an item.
 * Set s_enabled value to 1, for a given item id
 *
 * @param int $id
 * @return bool
 */
public function activate( $id, $secret = NULL )
{
    $aWhere = array();
    if( $secret == NULL ) {
        $item[0] = $this->manager->findByPrimaryKey( $id );
        $aWhere = array('pk_i_id' => $id);
    } else {
        $item = $this->manager->listWhere("i.s_secret = '%s' AND i.pk_i_id = '%s' ",
            $secret, $id);
        $aWhere = array('s_secret' => $secret, 'pk_i_id' => $id);
    }

    if($item[0]['b_enabled']==1 && $item[0]['b_active']==0) {

        $result = $this->manager->update(
            array('b_active' => 1),
            $aWhere
        );

        // updated correctly
        if($result == 1) {
            osc_run_hook( 'activate_item', $id );
        }
    }
}

```

```

        // b_enabled == 1 && b_active == 1
        if ($item[0]['b_spam']==0 && !osc.isExpired($item[0]['dt_expiration'])) {
            $this->_increaseStats($item[0]);
        }

        return true;
    }
    return false;
} else {
    return -1;
}
}

/**
 * Deactivates an item
 * Set s_active value to 0, for a given item id
 *
 * @param int $id
 * @return bool
 */
public function deactivate($id)
{
    $result = $this->manager->update(
        array('b_active' => 0),
        array('pk_i_id' => $id)
    );

    // updated correctly
    if($result == 1) {
        osc_run_hook('deactivate_item', $id);
        $item = $this->manager->findByPrimaryKey($id);
        if($item['b_enabled']==1 && $item['b_spam']==0 && !osc.isExpired($item['
            dt_expiration'])) {
            $this->_decreaseStats($item);
        }
        return true;
    }
    return false;
}

/**
 * Enable an item
 * Set s_enabled value to 1, for a given item id
 *
 * @param int $id
 * @return bool
 */
public function enable($id)
{
    $result = $this->manager->update(
        array('b_enabled' => 1),
        array('pk_i_id' => $id)
    );

    // updated correctly
    if($result == 1) {
        osc_run_hook('enable_item', $id);
        $item = $this->manager->findByPrimaryKey($id);
        if($item['b_active']==1 && $item['b_spam']==0 && !osc.isExpired($item['
            dt_expiration'])) {
            $this->_increaseStats($item);
        }
        return true;
    }
    return false;
}

/**
 * Disable an item.
 * Set s_enabled value to 0, for a given item id
 *
 * @param int $id
 * @return bool
 */
public function disable($id)
{
    $result = $this->manager->update(
        array('b_enabled' => 0),
        array('pk_i_id' => $id)
    );

    // updated correctly
    if($result == 1) {
        osc_run_hook('disable_item', $id);
        $item = $this->manager->findByPrimaryKey($id);
        if($item['b_active']==1 && $item['b_spam']==0 && !osc.isExpired($item['
            dt_expiration'])) {
            $this->_decreaseStats($item);
        }
        return true;
    }
    return false;
}

/**
 * Set premium value depending on $on, for a given item id
 *
 * @param int $id
 * @param bool $on

```

```

    * @return bool
    */
    public function premium($id, $on = true)
    {
        $value = 0;
        if($on) {
            $value = 1;
        }

        $result = $this->manager->update(
            array('b_premium' => (int)$value)
            ,array('pk_i_id' => $id)
        );
        // updated corretcly
        if($result == 1) {
            if($on) {
                osc_run_hook("item-premium-on", $id);
            } else {
                osc_run_hook("item-premium-off", $id);
            }
            return true;
        }
        return false;
    }

    /**
     * Set spam value depending on $on, for a given item id
     *
     * @param int $id
     * @param bool $on
     * @return bool
     */
    public function spam($id, $on = true)
    {
        $item = $this->manager->findByPrimaryKey($id);
        if($on) {
            $result = $this->manager->update(
                array('b_spam' => '1')
                ,array('pk_i_id' => $id)
            );
        } else {
            $result = $this->manager->update(
                array('b_spam' => '0')
                ,array('pk_i_id' => $id)
            );
        }

        // updated corretcly
        if($result == 1) {
            if($on) {
                osc_run_hook("item_spam_on", $id);
            } else {
                osc_run_hook("item_spam_off", $id);
            }

            if($item['b_active']==1 && $item['b_enabled']==1 && !osc_isExpired($item['dt_expiration']) && $item['b_spam']==0) {
                $this->_decreaseStats($item);
            } else if($item['b_active']==1 && $item['b_enabled']==1 && !osc_isExpired($item['dt_expiration']) && $item['b_spam']==1) {
                $this->_increaseStats($item);
            }

            return true;
        }
        return false;
    }

    /**
     * Private function for increment stats.
     * tables: t_user/t_category_stats/t_country_stats/t_region_stats/t_city_stats
     *
     * @param array item
     */
    private function _increaseStats($item)
    {
        if($item['fk_i_user_id']!= null) {
            User::newInstance()->increaseNumItems($item['fk_i_user_id']);
        }
        CategoryStats::newInstance()->increaseNumItems($item['fk_i_category_id']);
        CountryStats::newInstance()->increaseNumItems($item['fk_c_country_code']);
        RegionStats::newInstance()->increaseNumItems($item['fk_i_region_id']);
        CityStats::newInstance()->increaseNumItems($item['fk_i_city_id']);
    }

    /**
     * Private function for decrease stats.
     * tables: t_user/t_category_stats/t_country_stats/t_region_stats/t_city_stats
     *
     * @param array item
     */
    private function _decreaseStats($item)
    {
        if($item['fk_i_user_id']!= null) {
            User::newInstance()->decreaseNumItems($item['fk_i_user_id']);
        }
        CategoryStats::newInstance()->decreaseNumItems($item['fk_i_category_id']);
        CountryStats::newInstance()->decreaseNumItems($item['fk_c_country_code']);
        RegionStats::newInstance()->decreaseNumItems($item['fk_i_region_id']);
    }

```

```

    CityStats::newInstance()->decreaseNumItems($item['fk_i_city_id']);
}

/**
 * Delete an item, given s_secret and item id.
 *
 * @param <type> $secret
 * @param <type> $itemId
 */
public function delete( $secret , $itemId )
{
    $item = $this->manager->findByPrimaryKey($itemId);
    $item_title = osc_item_title();

    $user = User::newInstance()->findByPrimaryKey($item['fk_i_user_id']);

    osc_run_hook('before_delete_item', $itemId);

    if( $item['s_secret'] == $secret ) {
        $this->deleteResourcesFromHD( $item['pk_i_id'] );
        Log::newInstance()->insertLog( 'item', 'delete', $itemId, $item['s_title'],
            $this->is_admin ? 'admin' : 'user', $this->is_admin ? osc_logged_admin_id()
            : osc_logged_user_id() );
        $result = $this->manager->deleteByPrimaryKey( $itemId );
        if($result!==false) {
            if(osc_is_admin_user_logged_in()){
                $this->fn_email_item_deleted($user, $itemId,
                    $item_title);
            }

            osc_run_hook('after_delete_item', $itemId);
        }
        return $result;
    }

    return false;
}

public function fn_email_item_deleted($user, $itemId, $item_title){

    $title = "Collab-Ads - Your listing has been removed.";

    $username = $user['s_name'];
    $web_link = '<a href="' . osc_base_url() . '>' . osc_page_title() . '</a>';

    $body = "<p>Hi " . $username . "</p><p>Your listing " . $item_title . " has been
        removed by the admin. It may have been identified as spam, duplicate,
        expired, misclassified, or offensive. Visit the website to make a new one
        .</p><p>Regards,</p><p>". $web_link . "</p>";

    $emailParams = array(
        'subject' => $title,
        'from' => osc_contact_email(),
        'to' => $user['s_author_email'],
        'to_name' => $user['s_author_name'],
        'body' => $body,
        'alt_body' => $body
    );
    osc_sendMail($emailParams);
    return 1;
}

/**
 * Delete resources from the hard drive
 *
 * @param <type> $itemId
 */
public function deleteResourcesFromHD( $itemId )
{
    $resources = ItemResource::newInstance()->getAllResourcesFromItem($itemId);
    Log::newInstance()->insertLog('itemActions', 'deleteResourcesFromHD', $itemId,
        $itemId, $this->is_admin?'admin':'user', $this->is_admin?osc_logged_admin_id()
        : osc_logged_user_id());
    $log_ids = '';
    foreach($resources as $resource) {
        osc_deleteResource($resource['pk_i_id'], $this->is_admin);
        $log_ids .= $resource['pk_i_id'] . ",";
    }
    Log::newInstance()->insertLog('itemActions', 'deleteResourcesFromHD', $itemId,
        substr($log_ids,0, 250), $this->is_admin?'admin':'user', $this->is_admin?
        osc_logged_admin_id():osc_logged_user_id());
}

/**
 * Mark an item
 *
 * @param int $id
 * @param string $as
 */
public function mark( $id, $as )
{
    switch ($as) {
        case 'spam':
            $column = 'i_num_spam';
            break;
        case 'badcat':
            $column = 'i_num_bad_classified';
            break;
    }
}

```

```

        case 'offensive':
            $column = 'i_num_offensive';
        break;
        case 'repeated':
            $column = 'i_num_repeated';
        break;
        case 'expired':
            $column = 'i_num_expired';
        break;
    }

    ItemStats::newInstance()->increase( $column, $id );
}

public function send_friend()
{
    // get data for this function
    $aItem = $this->prepareDataForFunction( 'send_friend' );

    $item      = $aItem['item'];
    $s_title   = $aItem['s_title'];
    View::newInstance()->exportVariableToView( 'item', $item);

    osc_run_hook( 'hook_email_send_friend', $aItem);
    $item_url = osc_item_url();
    $item_url = '<a href="'. $item_url. '" >'. $item_url. '</a>';
    Params::setParam( 'item_url', $item_url );
    osc_add_flash_ok_message( sprintf( _m( 'We just sent your message to %s' ), $aItem[ '
        friendName' ] ) );
    return true;
}

public function contact()
{
    $flash_error = '';
    $aItem = $this->prepareDataForFunction( 'contact' );

    // check parameters
    if( !osc_validate_email( $aItem[ 'yourEmail' ], true ) ){
        $flash_error .= --( "Invalid email address" ) . PHP_EOL;
    } else if( !osc_validate_text( $aItem[ 'message' ] ) ){
        $flash_error .= --( "Message: this field is required" ) . PHP_EOL;
    } else if( !osc_validate_text( $aItem[ 'yourName' ] ) ){
        $flash_error .= --( "Your name: this field is required" ) . PHP_EOL;
    }

    if( $flash_error != '' ){
        return $flash_error;
    } else {
        osc_run_hook( 'hook_email_item_inquiry', $aItem );
    }
}

/*
 *
 */
public function add_comment()
{
    if( !osc_comments_enabled() ) {
        return 7;
    }

    $aItem = $this->prepareDataForFunction( 'add_comment' );

    $authorName      = trim( $aItem[ 'authorName' ] );
    $authorName      = strip_tags( $authorName );
    $authorEmail     = trim( $aItem[ 'authorEmail' ] );
    $authorEmail     = strip_tags( $authorEmail );
    $body            = trim( $aItem[ 'body' ] );
    $body            = strip_tags( $body );
    $title           = $aItem[ 'title' ];
    $itemId          = $aItem[ 'id' ];
    $userId          = $aItem[ 'userId' ];
    $status_num     = -1;

    $item = $this->manager->findByPrimaryKey( $itemId );
    View::newInstance()->exportVariableToView( 'item', $item );
    $itemURL = osc_item_url();
    $itemURL = '<a href="'. $itemURL. '" >'. $itemURL. '</a>';

    Params::setParam( 'itemURL', $itemURL );

    if( osc_reg_user_post_comments() && !osc_is_web_user_logged_in() ) {
        Session::newInstance()->setForm( 'commentAuthorName', $authorName );
        Session::newInstance()->setForm( 'commentTitle', $title );
        Session::newInstance()->setForm( 'commentBody', $body );
        return 6;
    }

    if( !preg_match( '|^.*?@.{2,}\.?.{2,3}$|', $authorEmail ) ) {
        Session::newInstance()->setForm( 'commentAuthorName', $authorName );
        Session::newInstance()->setForm( 'commentTitle', $title );
        Session::newInstance()->setForm( 'commentBody', $body );
        return 3;
    }

    if( $body == '' ) {
        Session::newInstance()->setForm( 'commentAuthorName', $authorName );

```

```

        Session::newInstance()->_setForm('commentAuthorEmail', $authorEmail);
        Session::newInstance()->_setForm('commentTitle', $title);
        return 4;
    }

    $num_moderate_comments = osc_moderate_comments();
    if($userId==null) {
        $num_comments = 0;
    } else {
        $user = User::newInstance()->findByPrimaryKey($userId);
        $num_comments = $user['i_comments'];
    }

    if ($num_moderate_comments == -1 || ($num_moderate_comments != 0 && $num_comments
    >= $num_moderate_comments)) {
        $status = 'ACTIVE';
        $status_num = 2;
    } else {
        $status = 'INACTIVE';
        $status_num = 1;
    }

    if (osc_akismet_key()) {
        require_once LIB_PATH . 'Akismet.class.php';
        $akismet = new Akismet(osc_base_url(), osc_akismet_key());
        $akismet->setCommentAuthor($authorName);
        $akismet->setCommentAuthorEmail($authorEmail);
        $akismet->setCommentContent($body);
        $akismet->setPermalink($itemURL);

        $status = $akismet->isCommentSpam() ? 'SPAM' : $status;
        if($status == 'SPAM') {
            $status_num = 5;
        }
    }

    $mComments = ItemComment::newInstance();
    $aComment = array('dt_pub_date' => date('Y-m-d H:i:s')
        , 'fk_i_item_id' => $itemID
        , 's_author_name' => $authorName
        , 's_author_email' => $authorEmail
        , 's_title' => $title
        , 's_body' => $body
        , 'b_active' => ($status=='ACTIVE' ? 1 : 0)
        , 'b_enabled' => 1
        , 'fk_i_user_id' => $userId);

    osc_run_hook('before_add_comment', $aComment);

    if( $mComments->insert($aComment) ) {
        $commentID = $mComments->dao->insertedId();
        if($status_num == 2 && $userId != null) { // COMMENT IS ACTIVE
            $user = User::newInstance()->findByPrimaryKey($userId);
            if( $user ) {
                User::newInstance()->update( array( 'i_comments' => $user['i_comments']
                + 1)
                , array( 'pk_i_id' => $user['pk_i_id'] )
                );
            }
        }

        //Notify admin
        if ( osc_notify_new_comment() ) {
            osc_run_hook('hook_email_new_comment_admin', $aItem);
        }

        //Notify user
        if ( osc_notify_new_comment_user() ) {
            osc_run_hook('hook_email_new_comment_user', $aItem);
        }

        osc_run_hook( 'add_comment', $commentID );

        return $status_num;
    }

    return -1;
}

/**
 * Return an array with all data necessary for do the action
 * @param <type> $action
 */
private function prepareDataForFunction( $action )
{
    $aItem = array();

    switch ( $action ){
        case 'send_friend':
            $item = $this->manager->findByPrimaryKey( Params::getParam('id') );

            $aItem['item'] = $item;
            View::newInstance()->exportVariableToView('item', $aItem['item']);
            $aItem['yourName'] = Params::getParam('yourName');
            $aItem['yourEmail'] = Params::getParam('yourEmail');

            $aItem['friendName'] = Params::getParam('friendName');
            $aItem['friendEmail'] = Params::getParam('friendEmail');
    }
}

```

```

        $aItem['s_title']      = $item['s_title'];
        $aItem['message']    = Params::getParam('message');
    break;
    case 'contact':
        $item = $this->manager->findByPrimaryKey( Params::getParam('id') );

        $aItem['item']        = $item;
        View::newInstance()->exportVariableToView('item', $aItem['item']);
        $aItem['id']          = Params::getParam('id');
        $aItem['yourEmail']   = Params::getParam('yourEmail');
        $aItem['yourName']    = Params::getParam('yourName');
        $aItem['message']     = Params::getParam('message');
        $aItem['phoneNumber'] = Params::getParam('phoneNumber');
    break;
    case 'add_comment':
        $item = $this->manager->findByPrimaryKey( Params::getParam('id') );

        $aItem['item']        = $item;
        View::newInstance()->exportVariableToView('item', $aItem['item']);
        $aItem['authorName']  = Params::getParam('authorName');
        $aItem['authorEmail'] = Params::getParam('authorEmail');
        $aItem['body']        = Params::getParam('body');
        $aItem['title']       = Params::getParam('title');
        $aItem['id']          = Params::getParam('id');
        $aItem['userId']      = Session::newInstance()->get('userId');
        if($aItem['userId'] == ''){
            $aItem['userId'] = NULL;
        }
    break;
    default:
}
return $aItem;
}

/**
 * Return an array with all data necessary for do the action (ADD OR EDIT)
 * @param <type> $is_add
 * @return array
 */
public function prepareData( $is_add )
{
    $aItem = array();
    $data = array();

    $userId = null;
    if( $this->is_admin ) {
        // user
        $data = User::newInstance()->findByEmail(Params::getParam('contactEmail'));
        if( isset($data['pk_i_id']) && is_numeric($data['pk_i_id']) ) {
            $userId = $data['pk_i_id'];
        }
    } else {
        $userId = Session::newInstance()->get('userId');
        if( $userId == '' ) {
            $userId = NULL;
        } elseif ( $userId != NULL ) {
            $data = User::newInstance()->findByPrimaryKey( $userId );
        }
    }

    if( $userId != null ) {
        $aItem['contactName'] = $data['s_name'];
        $aItem['contactEmail'] = $data['s_email'];
        Params::setParam('contactName', $data['s_name']);
        Params::setParam('contactEmail', $data['s_email']);
    } else {
        $aItem['contactName'] = Params::getParam('contactName');
        $aItem['contactEmail'] = Params::getParam('contactEmail');
    }
    $aItem['userId'] = $userId;

    if( $is_add ) { // ADD
        if($this->is_admin) {
            $active = 'ACTIVE';
        } else {
            if(osc_moderate_items()>0) { // HAS TO VALIDATE
                if(!osc_is_web_user_logged_in()) { // NO USER IS LOGGED, VALIDATE
                    $active = 'INACTIVE';
                } else { // USER IS LOGGED
                    if(osc_logged_user_item_validation()) { //USER IS LOGGED, BUT NO
                        NEED TO VALIDATE
                        $active = 'ACTIVE';
                    } else { // USER IS LOGGED, NEED TO VALIDATE, CHECK NUMBER OF
                        PREVIOUS ITEMS
                        $user = User::newInstance()->findByPrimaryKey(
                            osc_logged_user_id());
                        if($user['i_items']<osc_moderate_items()) {
                            $active = 'INACTIVE';
                        } else {
                            $active = 'ACTIVE';
                        }
                    }
                }
            }
        } else if(osc_moderate_items()==0){
            if(osc_is_web_user_logged_in() && osc_logged_user_item_validation() )
            {
                $active = 'ACTIVE';
            } else {

```



```

        $active = 'INACTIVE';
    }
    } else {
        $active = 'ACTIVE';
    }
}
$Item['active'] = $active;
} else { // EDIT
    $Item['secret'] = Params::getParam('secret');
    $Item['idItem'] = Params::getParam('id');
}

// get params
$Item['catId'] = Params::getParam('catId');
$Item['countryId'] = Params::getParam('countryId');
$Item['country'] = Params::getParam('country');
$Item['region'] = Params::getParam('region');
$Item['regionId'] = Params::getParam('regionId');
$Item['city'] = Params::getParam('city');
$Item['cityId'] = Params::getParam('cityId');
$Item['price'] = (Params::getParam('price') != '') ? Params::getParam('price') : null;
$Item['cityArea'] = Params::getParam('cityArea');
$Item['address'] = Params::getParam('address');
$Item['currency'] = Params::getParam('currency');
$Item['showEmail'] = (Params::getParam('showEmail') != '') ? 1 : 0;
$Item['title'] = Params::getParam('title');
$Item['description'] = Params::getParam('description');
$Item['photos'] = Params::getFiles('photos');
$Item['s_ip'] = get-ip();
$Item['d_coord_lat'] = (Params::getParam('d_coord_lat') != '') ? Params::getParam('d_coord_lat') : null;
$Item['d_coord_long'] = (Params::getParam('d_coord_long') != '') ? Params::getParam('d_coord_long') : null;

if($is_add || $this->is_admin) {
    $dt_expiration = Params::getParam('dt_expiration');
    if($dt_expiration==1) {
        $Item['dt_expiration'] = '';
    } else if($dt_expiration!='' && (preg_match('/^([0-9]+)$/', $dt_expiration, $match) || preg_match('/([0-9]{4})-([0-9]{2})-([0-9]{2}) ([0-9]{2}):([0-9]{2}):([0-9]{2})|', $dt_expiration, $match))) {
        $Item['dt_expiration'] = $dt_expiration;
    } else {
        $category = Category::newInstance()->findByPrimaryKey($Item['catId']);
        $Item['dt_expiration'] = $category['i_expiration_days'];
    }
    unset($dt_expiration);
} else {
    $Item['dt_expiration'] = '';
};

// check params
$country = Country::newInstance()->findByCode($Item['countryId']);
if( count($country) > 0 ) {
    $countryId = $country['pk_c_code'];
    $countryName = $country['s_name'];
} else {
    $countryId = null;
    $countryName = $Item['country'];
}
$Item['countryId'] = $countryId;
$Item['countryName'] = $countryName;

if( $Item['regionId'] != '' ) {
    if( intval($Item['regionId']) ) {
        $region = Region::newInstance()->findByPrimaryKey($Item['regionId']);
        if( count($region) > 0 ) {
            $regionId = $region['pk_i_id'];
            $regionName = $region['s_name'];
        }
    }
} else {
    $regionId = null;
    $regionName = $Item['region'];
    if( $Item['countryId'] != '' ) {
        $auxRegion = Region::newInstance()->findByName($Item['region'], $Item['countryId']);
        if($auxRegion){
            $regionId = $auxRegion['pk_i_id'];
            $regionName = $auxRegion['s_name'];
        }
    }
}

$Item['regionId'] = $regionId;
$Item['regionName'] = $regionName;

if( $Item['cityId'] != '' ) {
    if( intval($Item['cityId']) ) {
        $city = City::newInstance()->findByPrimaryKey($Item['cityId']);
        if( count($city) > 0 ) {
            $cityId = $city['pk_i_id'];
            $cityName = $city['s_name'];
        }
    }
} else {
    $cityId = null;
    $cityName = $Item['city'];
}

```

```

        if( $aItem['countryId'] != '' ) {
            $auxCity = City::newInstance()->findByName($aItem['city'], $aItem['
                regionId' ] );
            if($auxCity){
                $cityId = $auxCity['pk_i_id'];
                $cityName = $auxCity['s_name'];
            }
        }

        $aItem['cityId'] = $cityId;
        $aItem['cityName'] = $cityName;

        if( $aItem['cityArea'] == '' ) {
            $aItem['cityArea'] = null;
        }

        if( $aItem['address'] == '' ) {
            $aItem['address'] = null;
        }

        if( !is_null($aItem['price']) ) {
            $price = str_replace(osc_locale_thousands_sep(), '', trim($aItem['price']));
            $price = str_replace(osc_locale_dec_point(), '.', $price);
            $aItem['price'] = $price*1000000;
            // $aItem['price'] = (float) $aItem['price'];
        }

        if( $aItem['catId'] == '' ){
            $aItem['catId'] = 0;
        }

        if( $aItem['currency'] == '' ) {
            $aItem['currency'] = null;
        }

        $this->data = $aItem;
    }

function insertItemLocales($type, $title, $description, $itemId )
{
    foreach($title as $k => $data){
        $_title = $title[$k];
        $_description = $description[$k];
        if($type == 'ADD'){
            $this->manager->insertLocale($itemId, $k, $_title, $_description);
        }else if($type == 'EDIT'){
            $this->manager->updateLocaleForce($itemId, $k, $_title, $_description);
        }
    }
}

private function checkSize($aResources)
{
    $success = true;

    if($aResources != '' ) {
        // get allowedExt
        $maxSize = osc_max_size_kb() * 1024;
        foreach ( $aResources['error'] as $key => $error ) {
            $bool_img = false;
            if ( $error == UPLOAD_ERR_OK ) {
                $size = $aResources['size'][$key];
                if ($size >= $maxSize){
                    $success = false;
                }
            }
        }
    }
    if(!$success){
        osc_add_flash_error_message( _m("One of the files you tried to upload
            exceeds the maximum size"));
    }
    return $success;
}

private function checkAllowedExt($aResources)
{
    $success = true;
    require LIB_PATH . 'osclass/mimes.php';
    if($aResources != '' ) {
        // get allowedExt
        $aMimesAllowed = array();
        $aExt = explode(',', osc_allowed_extension() );
        foreach($aExt as $ext){
            if(isset($mimes[$ext])) {
                $mime = $mimes[$ext];
                if( is_array($mime) ) {
                    foreach($mime as $aux){
                        if( !in_array($aux, $aMimesAllowed) ) {
                            array_push($aMimesAllowed, $aux );
                        }
                    }
                } else {
                    if( !in_array($mime, $aMimesAllowed) ) {
                        array_push($aMimesAllowed, $mime );
                    }
                }
            }
        }
    }
}

```

```

    }
    foreach ($aResources['error'] as $key => $error) {
        $bool_img = false;
        if ($error == UPLOAD_ERR_OK) {
            // check mime file
            $fileMime = $aResources['type'][$key];
            if (stripos($fileMime, "image/") != FALSE) {
                if (function_exists("getimagesize")) {
                    $sinfo = getimagesize($aResources['tmp_name'][$key]);
                    if (isset($sinfo['mime'])) {
                        $fileMime = $sinfo['mime'];
                    } else {
                        $fileMime = '';
                    }
                }
            }
        };

        if (in_array($fileMime, $aMimesAllowed)) {
            $bool_img = true;
        }
        if (!$bool_img && $success) { $success = false; }
    }
}

if (!$success) {
    osc_add_flash_error_message(_m("The file you tried to upload does not
    have a valid extension"));
}

return $success;
}

public function uploadItemResources($aResources, $itemId)
{
    if ($aResources != '') {
        $swat = new Watermark();

        $itemResourceManager = ItemResource::newInstance();

        $numImagesItems = osc_max_images_per_item();
        $numImages = $itemResourceManager->countResources($itemId);
        foreach ($aResources['error'] as $key => $error) {
            if ($numImagesItems == 0 || ($numImagesItems > 0 && $numImages < $numImagesItems)) {
                if ($error == UPLOAD_ERR_OK) {
                    $tmpName = $aResources['tmp_name'][$key];
                    $imgres = ImageResizer::fromFile($tmpName);
                    $extension = $imgres->getExt();
                    $mime = $imgres->getMime();

                    $total_size = 0;

                    // Create normal size
                    $normal_path = $path = $tmpName."_normal";
                    $size = explode('x', osc_normal_dimensions());
                    ImageResizer::fromFile($tmpName)->resizeTo($size[0], $size[1])->
                    saveToFile($path);

                    if (osc_is_watermark_text()) {
                        $swat->doWatermarkText($path, osc_watermark_text_color(),
                        osc_watermark_text(), $mime);
                    } else if (osc_is_watermark_image()) {
                        $swat->doWatermarkImage($path, $mime);
                    }
                }

                // Create preview
                $path = $tmpName."_preview";
                $size = explode('x', osc_preview_dimensions());
                ImageResizer::fromFile($normal_path)->resizeTo($size[0], $size[1])
                ->saveToFile($path);

                // Create thumbnail
                $path = $tmpName."_thumbnail";
                $size = explode('x', osc_thumbnail_dimensions());
                ImageResizer::fromFile($normal_path)->resizeTo($size[0], $size[1])
                ->saveToFile($path);

                $numImages++;

                $itemResourceManager->insert(array(
                    'fk_i_item_id' => $itemId
                ));
                $resourceId = $itemResourceManager->dao->insertedId();
                osc_copy($tmpName.'_normal', osc_uploads_path() . $resourceId .
                '.' . $extension);
                osc_copy($tmpName.'_preview', osc_uploads_path() . $resourceId .
                '_preview.' . $extension);
                osc_copy($tmpName.'_thumbnail', osc_uploads_path() . $resourceId .
                '_thumbnail.' . $extension);
                if (osc_keep_original_image()) {
                    $path = osc_uploads_path() . $resourceId . '_original.' .
                    $extension;
                    move_uploaded_file($tmpName, $path);
                }

                $s_path = str_replace(osc_base_path(), '', osc_uploads_path());
                $itemResourceManager->update(

```

```

        array(
            's_path' => $s_path
            's_name' => osc_genRandomPassword()
            's_extension' => $extension
            's_content_type' => $mime
        )
        ,array(
            'pk_i_id' => $resourceId
            'fk_i_item_id' => $itemId
        )
    );
    osc_run_hook('uploaded_file', ItemResource::newInstance()->
        findByPrimaryKey($resourceId));
    }
}
}
unset($itemResourceManager);
}
return 0; // NO PROBLEMS
}

public function sendEmails($aItem){
    $item = $aItem['item'];
    View::newInstance()->exportVariableToView('item', $item);

    /**
     * Send email to non-reg user requesting item activation
     */
    if( Session::newInstance()->get('userId') == '' && $aItem['active']=='INACTIVE' )
    {
        osc_run_hook('hook_email_item_validation_non_register_user', $item);
    } else if ( $aItem['active']=='INACTIVE' ) { // USER IS REGISTERED
        osc_run_hook('hook_email_item_validation', $item);
    } else if( Session::newInstance()->get('userId') == '' ) { // USER IS NOT
        REGISTERED
        osc_run_hook('hook_email_new_item_non_register_user', $item);
    }

    /**
     * Send email to admin about the new item
     */
    if (osc_notify_new_item()) {
        osc_run_hook('hook_email_admin_new_item', $item);
    }
}

}

?>

<?php
/**
 * Osclass software for creating and publishing online classified advertising platforms
 *
 * Copyright (C) 2012 OSCLASS
 *
 * This program is free software: you can redistribute it and/or modify it under the terms
 * of the GNU Affero General Public License as published by the Free Software Foundation,
 * either version 3 of the License, or (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY;
 * without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
 * See the GNU Affero General Public License for more details.
 *
 * You should have received a copy of the GNU Affero General Public
 * License along with this program. If not, see <http://www.gnu.org/licenses/>.
 */

function osc_listLocales() {
    $languages = array();

    $codes = osc_listLanguageCodes();
    foreach($codes as $code) {
        $path = sprintf('%s%s/index.php', osc_translations_path(), $code);
        $fxName = sprintf('locale_%s_info', $code);
        if(file_exists($path)) {
            require_once $path;
            if(function_exists($fxName)) {
                $languages[$code] = call_user_func($fxName);
                $languages[$code]['code'] = $code;
            }
        }
    }

    return $languages;
}

function osc_checkLocales() {
    $locales = osc_listLocales();

    foreach($locales as $locale) {
        $data = OSCLocale::newInstance()->findByPrimaryKey($locale['code']);
        if( !is_array($data) ) {
            $values = array(
                'pk_c_code' => $locale['code'],
                's_name' => $locale['name'],
                's_short_name' => $locale['short_name'],
                's_description' => $locale['description'],
                's_version' => $locale['version'],
            );
        }
    }
}

```

```

        's_author_name'      => $locale['author_name'],
        's_author_url'      => $locale['author_url'],
        's_currency_format' => $locale['currency_format'],
        's_date_format'     => $locale['date_format'],
        's_stop_words'      => $locale['stop_words'],
        'b_enabled'         => 0,
        'b_enabled_bo'     => 1
    );
    $result = OSCLocale::newInstance()->insert($values);

    if( !$result ) {
        return false;
    }

    // if it's a demo, we don't import any sql
    if( defined('DEMO') ) {
        return true;
    }

    // inserting e-mail translations
    $path = sprintf( '%s%s/mail.sql', osc_translations_path(), $locale['code'] );
    if( file_exists($path) ) {
        $sql = file_get_contents($path);
        $conn = DBConnectionClass::newInstance();
        $c_db = $conn->getOscClassDb();
        $comm = new DBCommandClass( $c_db );
        $result = $comm->importSQL( $sql );
        if( !$result ) {
            return false;
        }
    }
} else {
    // update language version
    OSCLocale::newInstance()->update(
        array( 's_version' => $locale['version'] ),
        array( 'pk_c_code' => $locale['code'] )
    );
}
}

return true;
}

function osc_listLanguageCodes() {
    $codes = array();

    $dir = opendir(osc_translations_path());
    while( $file = readdir($dir) ) {
        if( preg_match( '/^[a-z-]+$/i', $file ) ) {
            $codes[] = $file;
        }
    }
    closedir($dir);

    return $codes;
}

?>

<?php

/**
 * Osclass software for creating and publishing online classified advertising
 * platforms
 *
 * * Copyright (C) 2012 OSCLASS
 *
 * * This program is free software: you can redistribute it and/or modify it under the terms
 * * of the GNU Affero General Public License as published by the Free Software Foundation,
 * * either version 3 of the License, or (at your option) any later version.
 *
 * * This program is distributed in the hope that it will be useful, but WITHOUT ANY
 * * WARRANTY;
 * * without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
 * * PURPOSE.
 * * See the GNU Affero General Public License for more details.
 *
 * * You should have received a copy of the GNU Affero General Public
 * * License along with this program. If not, see <http://www.gnu.org/licenses/>.
 */

$mimes = array(
    'hqx' => 'application/mac-binhex40',
    'cpt' => 'application/mac-compactpro',
    'csv' => array('text/x-comma-separated-values', 'text/comma-separated-values', 'application/octet-stream', 'application/vnd.ms-excel', 'text/x-csv', 'text/csv', 'application/csv', 'application/excel', 'application/vnd.ms-excel'),
    'bin' => 'application/macbinary',
    'dms' => 'application/octet-stream',
    'lha' => 'application/octet-stream',
    'lzh' => 'application/octet-stream',
    'exe' => array('application/octet-stream', 'application/x-msdownload'),
    'class' => 'application/octet-stream',
    'psd' => 'application/x-photoshop',
    'so' => 'application/octet-stream',
    'sea' => 'application/octet-stream',
    'dll' => 'application/octet-stream',
    'oda' => 'application/oda',

```

```

'pdf' => array('application/pdf', 'application/x-download'),
'ai' => 'application/postscript',
'eps' => 'application/postscript',
'ps' => 'application/postscript',
'smi' => 'application/smil',
'smil' => 'application/smil',
'mif' => 'application/vnd.mif',
'xls' => array('application/excel', 'application/vnd.ms-excel',
'application/msexcel'),
'ppt' => array('application/powerpoint', 'application/vnd.ms-
powerpoint'),
'wbxml' => 'application/wbxml',
'wmlc' => 'application/wmlc',
'dcr' => 'application/x-director',
'dir' => 'application/x-director',
'dxr' => 'application/x-director',
'dvi' => 'application/x-dvi',
'gtar' => 'application/x-gtar',
'gz' => 'application/x-gzip',
'php' => 'application/x-httpd-php',
'php4' => 'application/x-httpd-php',
'php3' => 'application/x-httpd-php',
'phtml' => 'application/x-httpd-php',
'phps' => 'application/x-httpd-php-source',
'js' => 'application/x-javascript',
'swf' => 'application/x-shockwave-flash',
'sit' => 'application/x-stuffit',
'tar' => 'application/x-tar',
'rar' => 'application/x-rar',
'tgz' => array('application/x-tar', 'application/x-gzip-
compressed'),
'xhtml' => 'application/xhtml+xml',
'xht' => 'application/xhtml+xml',
'zip' => array('application/x-zip', 'application
/x-zip-compressed'),
'mid' => 'audio/midi',
'midi' => 'audio/midi',
'mpga' => 'audio/mpeg',
'mp2' => 'audio/mpeg',
'mp3' => array('audio/mpeg', 'audio/mpg', 'audio/mpeg3', 'audio
/mp3'),
'aif' => 'audio/x-aiff',
'aiff' => 'audio/x-aiff',
'aifc' => 'audio/x-aiff',
'ram' => 'audio/x-pn-realaudio',
'rm' => 'audio/x-pn-realaudio',
'rpm' => 'audio/x-pn-realaudio-plugin',
'ra' => 'audio/x-realaudio',
'rv' => 'video/vnd.rn-realvideo',
'wav' => 'audio/x-wav',
'bmp' => 'image/bmp',
'gif' => 'image/gif',
'jpeg' => array('image/jpeg', 'image/pjpeg'),
'jpg' => array('image/jpeg', 'image/pjpeg'),
'jpe' => array('image/jpeg', 'image/pjpeg'),
'png' => array('image/png', 'image/x-png'),
'tiff' => 'image/tiff',
'tif' => 'image/tiff',
'css' => 'text/css',
'html' => 'text/html',
'htm' => 'text/html',
'shtml' => 'text/html',
'txt' => 'text/plain',
'text' => 'text/plain',
'log' => array('text/plain', 'text/x-log'),
'rtx' => 'text/richtext',
'rtf' => 'text/rtf',
'xml' => 'text/xml',
'xsl' => 'text/xml',
'mpeg' => 'video/mpeg',
'mpg' => 'video/mpeg',
'mpe' => 'video/mpeg',
'qt' => 'video/quicktime',
'mov' => 'video/quicktime',
'avi' => 'video/x-msvideo',
'movie' => 'video/x-sgi-movie',
'doc' => 'application/msword',
'docx' => 'application/vnd.openxmlformats-officedocument.
wordprocessingml.document',
'xlsx' => 'application/vnd.openxmlformats-officedocument.
spreadsheetml.sheet',
'word' => array('application/msword', 'application/octet-stream
'),
'xl' => 'application/excel',
'eml' => 'message/rfc822',
'json' => array('application/json', 'text/json'),
'odt' => 'application/vnd.oasis.opendocument.text',
'ods' => 'application/vnd.oasis.opendocument.spreadsheet',
'odp' => 'application/vnd.oasis.opendocument.presentation',
'odg' => 'application/vnd.oasis.opendocument.graphics',
'odc' => 'application/vnd.oasis.opendocument.chart',
'odf' => 'application/vnd.oasis.opendocument.formula',
'odb' => 'application/vnd.oasis.opendocument.database',
'odi' => 'application/vnd.oasis.opendocument.image',
'odm' => 'application/vnd.oasis.opendocument.text-master',
'ott' => 'application/vnd.oasis.opendocument.text-template',
'ots' => 'application/vnd.oasis.opendocument.spreadsheet-template',
'otp' => 'application/vnd.oasis.opendocument.presentation-template',
'otg' => 'application/vnd.oasis.opendocument.graphics-template'

```

```

        );
?>
<?php if ( ! defined('ABS_PATH')) exit('ABS_PATH is not loaded. Direct access is not allowed
.');
```

```

/*
 *   Oclass      software for creating and publishing online classified
 *               advertising platforms
 *
 *               Copyright (C) 2012 OSCLASS
 *
 *   This program is free software: you can redistribute it and/or
 *   modify it under the terms of the GNU Affero General Public License
 *   as published by the Free Software Foundation, either version 3 of
 *   the License, or (at your option) any later version.
 *
 *   This program is distributed in the hope that it will be useful, but
 *   WITHOUT ANY WARRANTY; without even the implied warranty of
 *   MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 *   GNU Affero General Public License for more details.
 *
 *   You should have received a copy of the GNU Affero General Public
 *   License along with this program. If not, see <http://www.gnu.org/licenses/>.
 */
abstract class Themes
{
    private static $instance;
    protected $theme;
    protected $theme_url;
    protected $theme_path;
    protected $theme_exists;

    protected $scripts;
    protected $queue;
    protected $styles;

    protected $resolved;
    protected $unresolved;

    public function __construct()
    {
        $this->scripts = array();
        $this->queue   = array();
        $this->styles  = array();
    }

    abstract protected function setCurrentThemeUrl();
    abstract protected function setCurrentThemePath();

    /* PUBLIC */
    public function setCurrentTheme($theme)
    {
        $this->theme = $theme;
        $this->setCurrentThemePath();
        $this->setCurrentThemeUrl();
    }

    public function getCurrentTheme()
    {
        return $this->theme;
    }

    public function getCurrentThemeUrl()
    {
        return $this->theme_url;
    }

    public function getCurrentThemePath()
    {
        return $this->theme_path;
    }

    public function getCurrentThemeStyles()
    {
        return $this->theme_url . 'css/';
    }

    public function getCurrentThemeJs()
    {
        return $this->theme_url . 'js/';
    }

    /**
     * Add style to be loaded
     *
     * @param type $id
     * @param type $url
     * @deprecated deprecated since version 3.1
     */
    public function addStyle($id, $url)
    {
        $this->styles[$id] = $url;
    }

    /**
     * Remove style to not be loaded

```

```

*
* @param type $id
* @deprecated deprecated since version 3.1
*/
public function removeStyle($id)
{
    unset($this->styles[$id]);
}

/**
 * Get the css styles urls
 *
 * @deprecated deprecated since version 3.1
 */
public function getStyles()
{
    return Styles::newInstance()->getStyles();
}

/**
 * Print the HTML tags to load the styles
 *
 * @deprecated deprecated since version 3.1
 */
public function printStyles()
{
    foreach($this->styles as $css) {
        echo '<link href="'. $css.'" rel="stylesheet" type="text/css" />' . PHP_EOL;
    }
}

/**
 * Add script to queue
 *
 * @param type $id
 * @deprecated deprecated since version 3.1
 */
public function enqueueScript($id)
{
    $this->queue[$id] = $id;
}

/**
 * Remove script to not be loaded
 *
 * @param type $id
 * @deprecated deprecated since version 3.1
 */
public function removeScript($id)
{
    unset($this->queue[$id]);
}

/**
 * Add script to be loaded
 *
 * @param type $id
 * @param type $url
 * @param type $dependencies mixed, it could be an array or a string
 * @deprecated deprecated since version 3.1
 */
public function registerScript($id, $url, $dependencies = null)
{
    $this->scripts[$id] = array(
        'key' => $id
        , 'url' => $url
        , 'dependencies' => $dependencies
    );
}

/**
 * Remove script to not be loaded
 *
 * @param type $id
 * @deprecated deprecated since version 3.1
 */
public function unregisterScript($id)
{
    unset($this->scripts[$id]);
}

/**
 * Order script before being printed on the HTML
 * @deprecated deprecated since version 3.1
 */
private function orderScripts()
{
    $this->resolved = array();
    $this->unresolved = array();
    $this->error = array();
    foreach($this->queue as $queue) {
        if(isset($this->scripts[$queue])) {
            $node = $this->scripts[$queue];
            if($node['dependencies']!=null) {
                $this->resolved[$node['key']] = $node['key'];
            } else {
                $this->solveDeps($node);
            }
        } else {
    }
}

```



```

        $this->error[$queue] = $queue;
    }
}
if(!empty($this->error)) {
    echo sprintf('--(ERROR: Some scripts could not be loaded (%s))', implode(", ",
        $this->error));
}
}

/**
 * Get the scripts urls
 * @deprecated deprecated since version 3.1
 */
public function getScripts()
{
    $scripts = array();
    $this->orderScripts();
    foreach($this->resolved as $id) {
        if( isset($this->scripts[$id]['url']) ) {
            $scripts[] = $this->scripts[$id]['url'];
        }
    }
    return $scripts;
}

/**
 * Print the HTML tags to load the scripts
 * @deprecated deprecated since version 3.1
 */
public function printScripts()
{
    foreach($this->getScripts() as $script) {
        echo '<script type="text/javascript" src="' . $script . '"></script>' .
            PHP_EOL;
    }
}

/**
 * Algorithm to solve the dependencies of the scripts
 * @param type $node
 * @deprecated deprecated since version 3.1
 */
private function solveDeps($node)
{
    $error = false;
    if(!isset($this->resolved[$node['key']])) {
        $this->unresolved[$node['key']] = $node['key'];
        if($node['dependencies']!=null) {
            if(is_array($node['dependencies'])) {
                foreach($node['dependencies'] as $dep) {
                    if(!in_array($dep, $this->resolved)) {
                        if(in_array($dep, $this->unresolved)) {
                            $this->error[$dep] = $dep;
                            $error = true;
                        } else {
                            if(isset($this->scripts[$dep])) {
                                $this->solveDeps($this->scripts[$dep]);
                            } else {
                                $this->error[$dep] = $dep;
                            }
                        }
                    }
                }
            }
        } else {
            if(!in_array($node['dependencies'], $this->resolved)) {
                if(in_array($node['dependencies'], $this->unresolved)) {
                    $this->error[$node['dependencies']] = $node['dependencies'];
                    $error = true;
                } else {
                    if(isset($this->scripts[$node['dependencies']])) {
                        $this->solveDeps($this->scripts[$node['dependencies']]);
                    } else {
                        $this->error[$node['dependencies']] = $node['dependencies'];
                    }
                }
            }
        }
    }
}

if(!$error) {
    $this->resolved[$node['key']] = $node['key'];
    unset($this->unresolved[$node['key']]);
}
}
}

/* file end: ./oc-includes/osclass/Themes.php */
<?php
/**
 * Osclass software for creating and publishing online classified advertising
 * platforms
 *
 * Copyright (C) 2012 OSCLASS
 *
 * This program is free software: you can redistribute it and/or modify it under the terms

```

```

* of the GNU Affero General Public License as published by the Free Software Foundation,
* either version 3 of the License, or (at your option) any later version.
*
* This program is distributed in the hope that it will be useful, but WITHOUT ANY
  WARRANTY;
* without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
  PURPOSE.
* See the GNU Affero General Public License for more details.
*
* You should have received a copy of the GNU Affero General Public
* License along with this program. If not, see <http://www.gnu.org/licenses/>.
*/

```

```

$timezone = array('Africa/Abidjan',
                  'Africa/Accra',
                  'Africa/Addis_Ababa',
                  'Africa/Algiers',
                  'Africa/Asmara',
                  'Africa/Bamako',
                  'Africa/Bangui',
                  'Africa/Banjul',
                  'Africa/Bissau',
                  'Africa/Blantyre',
                  'Africa/Brazzaville',
                  'Africa/Bujumbura',
                  'Africa/Cairo',
                  'Africa/Casablanca',
                  'Africa/Ceuta',
                  'Africa/Conakry',
                  'Africa/Dakar',
                  'Africa/Dar_es_Salaam',
                  'Africa/Djibouti',
                  'Africa/Douala',
                  'Africa/El_Aaiun',
                  'Africa/Freetown',
                  'Africa/Gaborone',
                  'Africa/Harare',
                  'Africa/Johannesburg',
                  'Africa/Kampala',
                  'Africa/Khartoum',
                  'Africa/Kigali',
                  'Africa/Kinshasa',
                  'Africa/Lagos',
                  'Africa/Libreville',
                  'Africa/Lome',
                  'Africa/Luanda',
                  'Africa/Lubumbashi',
                  'Africa/Lusaka',
                  'Africa/Malabo',
                  'Africa/Maputo',
                  'Africa/Maseru',
                  'Africa/Mbabane',
                  'Africa/Mogadishu',
                  'Africa/Monrovia',
                  'Africa/Nairobi',
                  'Africa/Ndjamena',
                  'Africa/Niamey',
                  'Africa/Nouakchott',
                  'Africa/Ouagadougou',
                  'Africa/Porto-Novo',
                  'Africa/Sao_Tome',
                  'Africa/Tripoli',
                  'Africa/Tunis',
                  'Africa/Windhoek',
                  'America/Adak',
                  'America/Anchorage',
                  'America/Anguilla',
                  'America/Antigua',
                  'America/Araguaina',
                  'America/Argentina/Buenos_Aires',
                  'America/Argentina/Catamarca',
                  'America/Argentina/Cordoba',
                  'America/Argentina/Jujuy',
                  'America/Argentina/La_Rioja',
                  'America/Argentina/Mendoza',
                  'America/Argentina/Rio_Gallegos',
                  'America/Argentina/Salta',
                  'America/Argentina/San_Juan',
                  'America/Argentina/San_Luis',
                  'America/Argentina/Tucuman',
                  'America/Argentina/Ushuaia',
                  'America/Aruba',
                  'America/Asuncion',
                  'America/Atikokan',
                  'America/Bahia',
                  'America/Barbados',
                  'America/Belem',
                  'America/Belize',
                  'America/Blanc-Sablon',
                  'America/Boa_Vista',
                  'America/Bogota',
                  'America/Boise',
                  'America/Cambridge_Bay',
                  'America/Campo_Grande',
                  'America/Cancun',
                  'America/Caracas',
                  'America/Cayenne',
                  'America/Cayman',
                  'America/Chicago',

```

'America/Chihuahua',  
 'America/Costa\_Rica',  
 'America/Cuiaba',  
 'America/Curacao',  
 'America/Danmarkshavn',  
 'America/Dawson',  
 'America/Dawson\_Creek',  
 'America/Denver',  
 'America/Detroit',  
 'America/Dominica',  
 'America/Edmonton',  
 'America/Eirunepe',  
 'America/El\_Salvador',  
 'America/Fortaleza',  
 'America/Glace\_Bay',  
 'America/Godthab',  
 'America/Goose\_Bay',  
 'America/Grand\_Turk',  
 'America/Grenada',  
 'America/Guadeloupe',  
 'America/Guatemala',  
 'America/Guayaquil',  
 'America/Guyana',  
 'America/Halifax',  
 'America/Havana',  
 'America/Hermosillo',  
 'America/Indiana/Indianapolis',  
 'America/Indiana/Knox',  
 'America/Indiana/Marengo',  
 'America/Indiana/Petersburg',  
 'America/Indiana/Tell\_City',  
 'America/Indiana/Vevay',  
 'America/Indiana/Vincennes',  
 'America/Indiana/Winamac',  
 'America/Inuvik',  
 'America/Iqaluit',  
 'America/Jamaica',  
 'America/Juneau',  
 'America/Kentucky/Louisville',  
 'America/Kentucky/Monticello',  
 'America/La\_Paz',  
 'America/Lima',  
 'America/Los\_Angeles',  
 'America/Maceio',  
 'America/Managua',  
 'America/Manaus',  
 'America/Marigot',  
 'America/Martinique',  
 'America/Mazatlan',  
 'America/Menominee',  
 'America/Merida',  
 'America/Mexico\_City',  
 'America/Miquelon',  
 'America/Moncton',  
 'America/Monterrey',  
 'America/Montevideo',  
 'America/Montreal',  
 'America/Montserrat',  
 'America/Nassau',  
 'America/New\_York',  
 'America/Nipigon',  
 'America/Nome',  
 'America/Noronha',  
 'America/North\_Dakota/Center',  
 'America/North\_Dakota/New\_Salem',  
 'America/Panama',  
 'America/Pangnirtung',  
 'America/Paramaribo',  
 'America/Phoenix',  
 'America/Port-au-Prince',  
 'America/Port\_of\_Spain',  
 'America/Porto\_Velho',  
 'America/Puerto\_Rico',  
 'America/Rainy\_River',  
 'America/Rankin\_Inlet',  
 'America/Recife',  
 'America/Regina',  
 'America/Resolute',  
 'America/Rio\_Branco',  
 'America/Santarem',  
 'America/Santiago',  
 'America/Santo\_Domingo',  
 'America/Sao\_Paulo',  
 'America/Scoresbysund',  
 'America/Shiprock',  
 'America/St\_Barthlemy',  
 'America/St\_Johns',  
 'America/St\_Kitts',  
 'America/St\_Lucia',  
 'America/St\_Thomas',  
 'America/St\_Vincent',  
 'America/Swift\_Current',  
 'America/Tegucigalpa',  
 'America/Thule',  
 'America/Thunder\_Bay',  
 'America/Tijuana',  
 'America/Toronto',  
 'America/Tortola',  
 'America/Vancouver',

'America/Whitehorse',  
 'America/Winnipeg',  
 'America/Yakutat',  
 'America/Yellowknife',  
 'Asia/Aden',  
 'Asia/Almaty',  
 'Asia/Amman',  
 'Asia/Anadyr',  
 'Asia/Aqtanay',  
 'Asia/Aqtobe',  
 'Asia/Ashgabat',  
 'Asia/Baghdad',  
 'Asia/Bahrain',  
 'Asia/Baku',  
 'Asia/Bangkok',  
 'Asia/Beirut',  
 'Asia/Bishkek',  
 'Asia/Brunei',  
 'Asia/Choibalsan',  
 'Asia/Chongqing',  
 'Asia/Colombo',  
 'Asia/Damascus',  
 'Asia/Dhaka',  
 'Asia/Dili',  
 'Asia/Dubai',  
 'Asia/Dushanbe',  
 'Asia/Gaza',  
 'Asia/Harbin',  
 'Asia/Ho-Chi-Minh',  
 'Asia/Hong-Kong',  
 'Asia/Hovd',  
 'Asia/Irkutsk',  
 'Asia/Jakarta',  
 'Asia/Jayapura',  
 'Asia/Jerusalem',  
 'Asia/Kabul',  
 'Asia/Kamchatka',  
 'Asia/Karachi',  
 'Asia/Kashgar',  
 'Asia/Kathmandu',  
 'Asia/Kolkata',  
 'Asia/Krasnoyarsk',  
 'Asia/Kuala\_Lumpur',  
 'Asia/Kuching',  
 'Asia/Kuwait',  
 'Asia/Macau',  
 'Asia/Magadan',  
 'Asia/Makassar',  
 'Asia/Manila',  
 'Asia/Muscat',  
 'Asia/Nicosia',  
 'Asia/Novokuznetsk',  
 'Asia/Novosibirsk',  
 'Asia/Omsk',  
 'Asia/Oral',  
 'Asia/Phnom\_Penh',  
 'Asia/Pontianak',  
 'Asia/Pyongyang',  
 'Asia/Qatar',  
 'Asia/Qyzylorda',  
 'Asia/Rangoon',  
 'Asia/Riyadh',  
 'Asia/Sakhalin',  
 'Asia/Samarkand',  
 'Asia/Seoul',  
 'Asia/Shanghai',  
 'Asia/Singapore',  
 'Asia/Taipei',  
 'Asia/Tashkent',  
 'Asia/Tbilisi',  
 'Asia/Tehran',  
 'Asia/Thimphu',  
 'Asia/Tokyo',  
 'Asia/Ulaanbaatar',  
 'Asia/Urumqi',  
 'Asia/Vientiane',  
 'Asia/Vladivostok',  
 'Asia/Yakutsk',  
 'Asia/Yekaterinburg',  
 'Asia/Yerevan',  
 'Atlantic/Azores',  
 'Atlantic/Bermuda',  
 'Atlantic/Canary',  
 'Atlantic/Cape\_Verde',  
 'Atlantic/Faroe',  
 'Atlantic/Madeira',  
 'Atlantic/Reykjavik',  
 'Atlantic/South\_Georgia',  
 'Atlantic/St\_Helena',  
 'Atlantic/Stanley',  
 'Australia/ACT',  
 'Australia/Adelaide',  
 'Australia/Brisbane',  
 'Australia/Broken\_Hill',  
 'Australia/Canberra',  
 'Australia/Currie',  
 'Australia/Darwin',  
 'Australia/Eucla',  
 'Australia/Hobart'

'Australia/LHI',  
 'Australia/Lindeman',  
 'Australia/Lord\_Howe',  
 'Australia/Melbourne',  
 'Australia/North',  
 'Australia/NSW',  
 'Australia/Perth',  
 'Australia/Queensland',  
 'Australia/South',  
 'Australia/Sydney',  
 'Australia/Tasmania',  
 'Australia/Victoria',  
 'Australia/West',  
 'Australia/Yancowinna',  
 'Europe/Amsterdam',  
 'Europe/Andorra',  
 'Europe/Athens',  
 'Europe/Belgrade',  
 'Europe/Berlin',  
 'Europe/Bratislava',  
 'Europe/Brussels',  
 'Europe/Bucharest',  
 'Europe/Budapest',  
 'Europe/Chisinau',  
 'Europe/Copenhagen',  
 'Europe/Dublin',  
 'Europe/Gibraltar',  
 'Europe/Guernsey',  
 'Europe/Helsinki',  
 'Europe/Isle\_of\_Man',  
 'Europe/Istanbul',  
 'Europe/Jersey',  
 'Europe/Kaliningrad',  
 'Europe/Kiev',  
 'Europe/Lisbon',  
 'Europe/Ljubljana',  
 'Europe/London',  
 'Europe/Luxembourg',  
 'Europe/Madrid',  
 'Europe/Malta',  
 'Europe/Mariehamn',  
 'Europe/Minsk',  
 'Europe/Monaco',  
 'Europe/Moscow',  
 'Europe/Oslo',  
 'Europe/Paris',  
 'Europe/Podgorica',  
 'Europe/Prague',  
 'Europe/Riga',  
 'Europe/Rome',  
 'Europe/Samara',  
 'Europe/San\_Marino',  
 'Europe/Sarajevo',  
 'Europe/Simferopol',  
 'Europe/Skopje',  
 'Europe/Sofia',  
 'Europe/Stockholm',  
 'Europe/Tallinn',  
 'Europe/Tirane',  
 'Europe/Uzhgorod',  
 'Europe/Vaduz',  
 'Europe/Vatican',  
 'Europe/Vienna',  
 'Europe/Vilnius',  
 'Europe/Volgograd',  
 'Europe/Warsaw',  
 'Europe/Zagreb',  
 'Europe/Zaporozhye',  
 'Europe/Zurich',  
 'Indian/Antananarivo',  
 'Indian/Chagos',  
 'Indian/Christmas',  
 'Indian/Cocos',  
 'Indian/Comoro',  
 'Indian/Kerguelen',  
 'Indian/Mahe',  
 'Indian/Maldives',  
 'Indian/Mauritius',  
 'Indian/Mayotte',  
 'Indian/Reunion',  
 'Pacific/Apia',  
 'Pacific/Auckland',  
 'Pacific/Chatham',  
 'Pacific/Easter',  
 'Pacific/Efate',  
 'Pacific/Enderbury',  
 'Pacific/Fakaofu',  
 'Pacific/Fiji',  
 'Pacific/Funafuti',  
 'Pacific/Galapagos',  
 'Pacific/Gambier',  
 'Pacific/Guadalcanal',  
 'Pacific/Guam',  
 'Pacific/Honolulu',  
 'Pacific/Johnston',  
 'Pacific/Kiritimati',  
 'Pacific/Kosrae',  
 'Pacific/Kwajalein',  
 'Pacific/Majuro',

```

        'Pacific/Marquesas',
        'Pacific/Midway',
        'Pacific/Nauru',
        'Pacific/Niue',
        'Pacific/Norfolk',
        'Pacific/Noumea',
        'Pacific/Pago_Pago',
        'Pacific/Palau',
        'Pacific/Pitcairn',
        'Pacific/Ponape',
        'Pacific/Port_Moresby',
        'Pacific/Rarotonga',
        'Pacific/Saipan',
        'Pacific/Tahiti',
        'Pacific/Tarawa',
        'Pacific/Tongatapu',
        'Pacific/Truk',
        'Pacific/Wake',
        'Pacific/Wallis');

?>

<?php
/*
 *      Oclass      software for creating and publishing online classified
 *                  advertising platforms
 *
 *                  Copyright (C) 2012 OSCLASS
 *
 *      This program is free software: you can redistribute it and/or
 *      modify it under the terms of the GNU Affero General Public License
 *      as published by the Free Software Foundation, either version 3 of
 *      the License, or (at your option) any later version.
 *
 *      This program is distributed in the hope that it will be useful, but
 *      WITHOUT ANY WARRANTY; without even the implied warranty of
 *      MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 *      GNU Affero General Public License for more details.
 *
 *      You should have received a copy of the GNU Affero General Public
 *      License along with this program. If not, see <http://www.gnu.org/licenses/>.
 */

error_reporting(0);

define( 'ABS_PATH', dirname(dirname(dirname(__FILE__))) . '/' );

require_once ABS_PATH . 'oc-includes/osclass/db.php';
require_once ABS_PATH . 'oc-includes/osclass/classes/DAO.php';
require_once ABS_PATH . 'oc-includes/osclass/model/Admin.php';
require_once ABS_PATH . 'oc-includes/osclass/helpers/hDatabaseInfo.php';
require_once ABS_PATH . 'oc-includes/osclass/core/Params.php';
require_once ABS_PATH . 'oc-includes/osclass/compatibility.php';
require_once ABS_PATH . 'config.php';

$old_passwd = Params::getParam('old_password', false, false);
$id_admin = Params::getParam('id');
$new_username = Params::getParam('new_username');
$new_passwd = Params::getParam('new_password', false, false);
$response = array('error' => 'Operation fail');

$mAdmin = Admin::newInstance();
$sAdmin = $mAdmin->findByIdPassword(1, sha1($old_passwd));

if($sAdmin){
    $result = -1;
    if( $new_username != '' ){
        $result = $mAdmin->update( array('s_username' => $new_username ), array('pk_i_id'
            => '1' ) );
    } elseif ( $new_passwd != '' ) {
        $result = $mAdmin->update( array('s_password' => sha1($new_passwd) ), array('
            pk_i_id' => '1' ) );
    }

    switch ($result) {
        case(1): $response = array('ok' => 'Updated sucessfully');
                break;
        case(0): $response = array('ok' => 'No changes');
                break;
    }
}

echo json_encode($response);

?>

<?php if ( ! defined('ABS_PATH')) exit('ABS_PATH is not loaded. Direct access is not allowed
. ');
/**
 * Oclass      software for creating and publishing online classified advertising
 *                  platforms
 *
 *                  Copyright (C) 2012 OSCLASS
 *
 *      This program is free software: you can redistribute it and/or modify it under the terms
 *      of the GNU Affero General Public License as published by the Free Software Foundation,
 *      either version 3 of the License, or (at your option) any later version.
 *
 */

```

```

* This program is distributed in the hope that it will be useful, but WITHOUT ANY
WARRANTY;
* without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
PURPOSE.
* See the GNU Affero General Public License for more details.
*
* You should have received a copy of the GNU Affero General Public
* License along with this program. If not, see <http://www.gnu.org/licenses/>.
*/

set_time_limit(0);

error_log(' ----- START upgrade-funcs ----- ');

if(!defined('ABS_PATH')) {
    define('ABS_PATH', dirname(dirname(dirname(__FILE__))) . '/');
}

require_once ABS_PATH . 'oc-load.php';
require_once LIB_PATH . 'osclass/helpers/hErrors.php';

if( !defined('AUTO_UPGRADE') ) {
    if( file_exists( osc_lib_path() . 'osclass/installer/struct.sql' ) ) {
        $sql = file_get_contents( osc_lib_path() . 'osclass/installer/struct.sql' );

        $conn = DBConnectionClass::newInstance();
        $c_db = $conn->getOsclassDb();
        $comm = new DBCommandClass( $c_db );

        $error_queries = $comm->updateDB( str_replace('/*TABLE_PREFIX*/', DB_TABLE_PREFIX,
            $sql) );
    }

    if( Params::getParam('skipdb') == '' ){
        if(! $error_queries[0] ) {
            $skip_db_link = osc_admin_base_url(true) . "?page=upgrade&action=upgrade-funcs
                &skipdb=true";
            $title = --('Osclass &raquo; Has some errors');
            $message = --("We've encountered some problems while updating the database
                structure. The following queries failed:");
            $message .= "<br/><br/>" . implode("<br>", $error_queries[2]);
            $message .= "<br/><br/>" . sprintf(--("These errors could be false-positive
                errors. If you're sure that is the case, you can <a href=\"%s\">continue
                with the upgrade</a>, or <a href=\"%s\">http://forums.osclass.org/\">ask in our
                forums</a>."), $skip_db_link);
            osc_die($title, $message);
        }
    }
}

$aMessages = array();
osc_set_preference('last_version_check', time());

$conn = DBConnectionClass::newInstance();
$c_db = $conn->getOsclassDb();
$comm = new DBCommandClass( $c_db );

if( osc_version() < 210 ) {
    $comm->query( sprintf("INSERT INTO %st_preference VALUES ('osclass', '
        save_latest_searches', '0', 'BOOLEAN')", DB_TABLE_PREFIX));
    $comm->query( sprintf("INSERT INTO %st_preference VALUES ('osclass', '
        purge_latest_searches', '1000', 'STRING')", DB_TABLE_PREFIX));
    $comm->query( sprintf("INSERT INTO %st_preference VALUES ('osclass', '
        selectable_parent_categories', '1', 'BOOLEAN')", DB_TABLE_PREFIX));
    $comm->query( sprintf("INSERT INTO %st_preference VALUES ('osclass', '
        ping_search_engines', '1', 'BOOLEAN')", DB_TABLE_PREFIX));
    $comm->query( sprintf("INSERT INTO %st_preference VALUES ('osclass', '
        numImages@items',
        '0', 'BOOLEAN')", DB_TABLE_PREFIX));
    $enableItemValidation = (getBoolPreference('enabled_item_validation') ? 0 : -1);
    $comm->query( sprintf("INSERT INTO %st_preference VALUES ('osclass', '
        moderate_items',
        '$enableItemValidation', 'INTEGER')", DB_TABLE_PREFIX));
    $comm->query( sprintf("INSERT INTO %st_preference VALUES ('osclass', '
        items_wait_time',
        '0', 'INTEGER')", DB_TABLE_PREFIX));
    $comm->query( sprintf("INSERT INTO %st_preference VALUES ('osclass', '
        comments_per_page',
        '10', 'INTEGER')", DB_TABLE_PREFIX));
    $comm->query( sprintf("INSERT INTO %st_preference VALUES ('osclass', '
        reg_user_post_comments', '0', 'BOOLEAN')", DB_TABLE_PREFIX));
    $comm->query( sprintf("INSERT INTO %st_preference VALUES ('osclass', '
        reg_user_can_contact', '0', 'BOOLEAN')", DB_TABLE_PREFIX));
    $comm->query( sprintf("INSERT INTO %st_preference VALUES ('osclass', '
        allow_report_osclass', '1', 'BOOLEAN')", DB_TABLE_PREFIX));

    // populate b_active/b_enabled (t_item-comment)
    $result = $comm->query( sprintf("SELECT * FROM %st_item-comment", DB_TABLE_PREFIX));
    $comments = $result->result();
    foreach($comments as $comment) {
        ItemComment::newInstance()->update(array("b_active" => ($comment['e_status'] == '
            ACTIVE' ? 1 : 0), 'b_enabled' => 1), array('pk_i.id' => $comment['pk_i.id']));
    }
    unset($comments);

    // populate b_active/b_enabled (t_item)
    $result = $comm->query( sprintf("SELECT * FROM %st_item", DB_TABLE_PREFIX));
    $items = $result->result();
    foreach($items as $item) {
        Item::newInstance()->update(array("b_active" => ($item['e_status'] == 'ACTIVE' ? 1
            : 0), 'b_enabled' => 1), array('pk_i.id' => $item['pk_i.id']));
    }
}

```

```

}
unset($items);

// populate i_items/i_comments/b_active/b_enabled (t_user)
$users = User::newInstance()->listAll();
foreach($users as $user) {
    $comments = count(ItemComment::newInstance()->findByAuthorID($user['pk_i_id']));
    $items = count(Item::newInstance()->findByUserIDEnabled($user['pk_i_id']));
    User::newInstance()->update(array( 'i_items' => $items, 'i_comments' => $comments
    ), array( 'pk_i_id' => $user['pk_i_id'] ) );
    // CHANGE FROM b_enabled to b_active
    User::newInstance()->update(array( 'b_active' => $user['b_enabled'], 'b_enabled'
    => 1 )
    , array( 'pk_i_id' => $user['pk_i_id'] ) );
}
unset($users);

// Drop e_status column in t_item and t_item_comment
$comm->query(sprintf("ALTER TABLE %st_item DROP e_status", DB.TABLE_PREFIX));
$comm->query(sprintf("ALTER TABLE %st_item_comment DROP e_status", DB.TABLE_PREFIX));
// Delete enabled_item_validation in t_preference
$comm->query(sprintf("DELETE FROM %st_preference WHERE s_name = '
enabled_item_validation'", DB.TABLE_PREFIX));

// insert two new e-mail notifications
$comm->query(sprintf("INSERT INTO %st_pages (s_internal_name, b_indelible, dt_pub_date
) VALUES ('email_alert_validation', 1, '%s')", DB.TABLE_PREFIX, date('Y-m-d H:i:s
')));
$comm->query(sprintf("INSERT INTO %st_pages_description (fk_i_pages_id,
fk_c_locale_code, s_title, s_text) VALUES (%d, 'en-US', 'Please validate your
alert', '<p>Hi {USER_NAME}</p><n><p>Please validate your alert registration by
clicking on the following link: {VALIDATION_LINK}</p><n><p>Thank you!</p><n><p>
Regards,</p><n><p>{WEB_TITLE}</p>'", DB.TABLE_PREFIX, $comm->insertedId()));
$comm->query(sprintf("INSERT INTO %st_pages (s_internal_name, b_indelible, dt_pub_date
) VALUES ('email_comment_validated', 1, '%s')", DB.TABLE_PREFIX, date('Y-m-d H:i:
s')));
$comm->query(sprintf("INSERT INTO %st_pages_description (fk_i_pages_id,
fk_c_locale_code, s_title, s_text) VALUES (%d, 'en-US', '{WEB_TITLE} - Your
comment has been approved', '<p>Hi {COMMENT_AUTHOR}</p><n><p>Your comment has been
approved on the following item: {ITEM_URL}</p><n><p>Regards,</p><n><p>{WEB_TITLE}</
p>'", DB.TABLE_PREFIX, $comm->insertedId()));
}

if(osc_version() < 220) {
    $comm->query(sprintf("INSERT INTO %st_preference VALUES ('osclass', 'watermark.text',
'', 'STRING')", DB.TABLE_PREFIX));
    $comm->query(sprintf("INSERT INTO %st_preference VALUES ('osclass', '
watermark.text.color', '', 'STRING')", DB.TABLE_PREFIX));
    $comm->query(sprintf("INSERT INTO %st_preference VALUES ('osclass', 'watermark.image
', '', 'STRING')", DB.TABLE_PREFIX));
    $comm->query(sprintf("INSERT INTO %st_preference VALUES ('osclass', 'watermark.place',
'centre', 'STRING')", DB.TABLE_PREFIX));
}

if(osc_version() < 230) {
    $comm->query(sprintf("
CREATE TABLE %st_item_description_tmp (
fk_i_item_id INT UNSIGNED NOT NULL,
fk_c_locale_code CHAR(5) NOT NULL,
s_title VARCHAR(100) NOT NULL,
s_description MEDIUMTEXT NOT NULL,
s_what LONGTEXT NULL,

PRIMARY KEY (fk_i_item_id, fk_c_locale_code),
INDEX (fk_i_item_id),
FOREIGN KEY (fk_i_item_id) REFERENCES %st_item (pk_i_id),
FOREIGN KEY (fk_c_locale_code) REFERENCES %st_item (pk_c_code)
) ENGINE=MyISAM DEFAULT CHARACTER SET 'UTF8' COLLATE 'UTF8_GENERAL_CI'", DB.TABLE_PREFIX,
DB.TABLE_PREFIX, DB.TABLE_PREFIX));

    $result = $comm->query(sprintf("SELECT * FROM %st_item_description", DB.TABLE_PREFIX
));
    $descriptions = $result->result();
    foreach($descriptions as $d) {
        $sql = sprintf("INSERT INTO %st_item_description_tmp ('fk_i_item_id', '
fk_c_locale_code', 's_title', 's_description', 's_what') VALUES (%d, '%s',
'%s', '%s', '%s')", DB.TABLE_PREFIX, $d['fk_i_item_id'], $d['
fk_c_locale_code'], $comm->connId->real_escape_string($d['s_title']), $comm->connId->
real_escape_string($d['s_description']), $comm->connId->
real_escape_string($d['s_what']));
        $comm->query($sql);
    }
    $comm->query(sprintf("RENAME TABLE '%st_item_description' TO '%st_item_description_old
'", DB.TABLE_PREFIX, DB.TABLE_PREFIX));
    $comm->query(sprintf("RENAME TABLE '%st_item_description_tmp' TO '%st_item_description
'", DB.TABLE_PREFIX, DB.TABLE_PREFIX));
    $comm->query(sprintf("ALTER TABLE %st_item_description ADD FULLTEXT s_description (
s_description, s_title);", DB.TABLE_PREFIX));

    // remove old tables if have the same number of rows
    $nItemDesc = $comm->query(sprintf("SELECT count(*) as total FROM %
st_item_description", DB.TABLE_PREFIX));
    $nItemDesc = $nItemDesc->row();
    $nItemDescOld = $comm->query(sprintf("SELECT count(*) as total FROM %
st_item_description_old", DB.TABLE_PREFIX));
    $nItemDescOld = $nItemDescOld->row();

    if( $nItemDesc['total'] == $nItemDescOld['total'] ) {

```



```

    $comm->query(sprintf('DROP TABLE %st_item_description_old ',DB.TABLE_PREFIX) );
}

$comm->query(sprintf("INSERT INTO %st_preference VALUES ('osclass', 'installed_plugins', '%s', 'STRING')", DB.TABLE_PREFIX, osc_get_preference('active_plugins')));
$comm->query(sprintf("INSERT INTO %st_preference VALUES ('osclass', 'mailserver_pop', '', 'STRING')", DB.TABLE_PREFIX));
$comm->query(sprintf("INSERT INTO %st_preference VALUES ('osclass', 'use_imagick', '0', 'BOOLEAN')", DB.TABLE_PREFIX));
$timezone = 'Europe/Madrid';
if(ini_get('date.timezone')!='') {
    $timezone = ini_get('date.timezone');
};
if(date_default_timezone_get()!='') {
    $timezone = date_default_timezone_get();
};
$comm->query(sprintf("INSERT INTO %st_preference VALUES ('osclass', 'timezone', '%s', 'STRING')", DB.TABLE_PREFIX, $timezone));

// alert table pages order improvement
$comm->query(sprintf("ALTER TABLE %st_pages ADD COLUMN i_order INT(3) NOT NULL DEFAULT 0 AFTER dt_mod_date;", DB.TABLE_PREFIX));
// order pages
$result = $comm->query(sprintf("SELECT pk_i_id FROM %st_pages WHERE b_indelible = 0", DB.TABLE_PREFIX) );
$aPages = $result->result();
foreach($aPages as $key => $page) {
    $comm->query(sprintf("UPDATE %st_pages SET i_order = %d WHERE pk_i_id = %d;", DB.TABLE_PREFIX, $key, $page['pk_i_id']) );
}

$comm->query(sprintf("INSERT INTO %st_pages (s_internal_name, b_indelible, dt_pub_date) VALUES ('email_item_validation_non_register_user', 1, '%s')", DB.TABLE_PREFIX, date('Y-m-d H:i:s')));
$comm->query(sprintf("INSERT INTO %st_pages_description (fk_i_pages_id, fk_c_locale_code, s_title, s_text) VALUES (%d, 'en-US', '{WEB.TITLE} - Validate your ad', '<p>Hi {USER.NAME},</p>\n<p>You're receiving this e-mail because an ad has been published at {WEB.TITLE}. Please validate this item by clicking on the link at the end of this e-mail. If you didn't publish this ad, please ignore this e-mail.</p>\n<p>Ad details:</p>\n<p>Contact name: {USER.NAME}<br />Contact e-mail: {USER.EMAIL}</p>\n<p>{ITEM.DESCRPTION_ALL_LANGUAGES}</p>\n<p>Price: {ITEM.PRICE}<br />Country: {ITEM.COUNTRY}<br />Region: {ITEM.REGION}<br />City: {ITEM.CITY}<br />Url: {ITEM.URL}<br /><br />Validate your ad: {VALIDATION.LINK}</p>\n<p>You're not registered at {WEB.TITLE}, but you can still edit or delete the item {ITEM.TITLE} for a short period of time.</p>\n<p>You can edit your item by following this link: {EDIT.LINK}</p>\n<p>You can delete your item by following this link: {DELETE.LINK}</p>\n<p>If you register as a user to post items, you will have full access to editing options.</p>\n<p>Regards,</p>\n{WEB.TITLE}</div >')", DB.TABLE_PREFIX, $comm->insertedId()));

$comm->query(sprintf("INSERT INTO %st_pages (s_internal_name, b_indelible, dt_pub_date) VALUES ('email_admin_new_user', 1, '%s')", DB.TABLE_PREFIX, date('Y-m-d H:i:s')));
$comm->query(sprintf("INSERT INTO %st_pages_description (fk_i_pages_id, fk_c_locale_code, s_title, s_text) VALUES (%d, 'en-US', '{WEB.TITLE} - New user', '<div><p>Dear {WEB.TITLE} admin,</p>\n<p>You're receiving this email because a new user has been created at {WEB.TITLE}.</p>\n<p>User details:</p>\n<p>Name: {USER.NAME}<br />E-mail: {USER.EMAIL}</p>\n<p>Regards,</p>\n<p>{WEB.TITLE}</p></div >')", DB.TABLE_PREFIX, $comm->insertedId()));
$comm->query(sprintf("INSERT INTO %st_pages (s_internal_name, b_indelible, dt_pub_date) VALUES ('email_contact_user', 1, '%s')", DB.TABLE_PREFIX, date('Y-m-d H:i:s')));
$comm->query(sprintf("INSERT INTO %st_pages_description (fk_i_pages_id, fk_c_locale_code, s_title, s_text) VALUES (%d, 'en-US', '{WEB.TITLE} - Someone has a question for you', '<p>Hi {CONTACT.NAME}!</p>\n<p>{USER.NAME} ({USER.EMAIL}, {USER.PHONE}) left you a message:</p>\n<p>{COMMENT}</p>\n<p>Regards,</p>\n<p>{WEB.TITLE}</p>>')", DB.TABLE_PREFIX, $comm->insertedId()));
$comm->query(sprintf("INSERT INTO %st_pages (s_internal_name, b_indelible, dt_pub_date) VALUES ('email_new_comment_user', 1, '%s')", DB.TABLE_PREFIX, date('Y-m-d H:i:s')));
$comm->query(sprintf("INSERT INTO %st_pages_description (fk_i_pages_id, fk_c_locale_code, s_title, s_text) VALUES (%d, 'en-US', '{WEB.TITLE} - New comment on the ad with id {ITEM.ID}', '<p>There's a new comment on the ad with id {ITEM.ID} <br />URL: {ITEM.URL}</p>\n<p>Title: {COMMENT.TITLE}<br />Comment: {COMMENT.TEXT}<br />Author: {COMMENT.AUTHOR}<br />Author's email: {COMMENT.EMAIL}</p>>')", DB.TABLE_PREFIX, $comm->insertedId()));
$comm->query(sprintf("INSERT INTO %st_preference VALUES ('osclass', 'notify_new_user', '1', 'BOOLEAN')", DB.TABLE_PREFIX));
$comm->query(sprintf("INSERT INTO %st_preference VALUES ('osclass', 'notify_new_comment_user', '0', 'BOOLEAN')", DB.TABLE_PREFIX));

$comm->query(sprintf("UPDATE %st_locale SET s_currency_format = '{NUMBER} {CURRENCY}'", DB.TABLE_PREFIX) );
$result = $comm->query(sprintf("SELECT pk_i_id, f_price FROM %st_item", DB.TABLE_PREFIX));
$items = $result->result();
foreach($items as $item) {
    if( $item['f_price'] == null ) {
        $sql = sprintf("UPDATE %st_item SET i_price = NULL WHERE pk_i_id = %d", DB.TABLE_PREFIX, $item['pk_i_id']);
    } else {
        $sql = sprintf("UPDATE %st_item SET i_price = %f WHERE pk_i_id = %d", DB.TABLE_PREFIX, (1000000 * $item['f_price']), $item['pk_i_id'] );
    }
    $comm->query($sql);
}
}

if( osc_version() < 234 ) {

```

```

@unlink(osc_admin_base_path()."upgrade.php");
@unlink(osc_admin_base_path()."themes/modern/tools/upgrade-plugins.php");
@unlink(osc_admin_base_path()."upgrade-plugin.php");
}

if( osc_version() < 240 ) {
// We no longer use s_what column in /*TABLE.PREFIX*/t_item_description
$comm->query( sprintf('ALTER TABLE %st_item_description DROP COLUMN s_what',
DB.TABLE_PREFIX) );

@unlink(osc_admin_base_path()."themes/modern/tools/images.php");

// NEW REWRITE
// Uncomment the unlink line prior to release
// @unlink(osc_base_path()."generate_rules.php");
osc_set_preference('rewrite_item_url', '{CATEGORIES}/{ITEM_TITLE}_{ITEM_ID}');
osc_set_preference('rewrite_cat_url', '{CATEGORIES}/');
osc_set_preference('rewrite_page_url', '{PAGE.SLUG}-p{PAGE.ID}');
osc_set_preference('rewrite_search_url', 'search/');
osc_set_preference('rewrite_search_country', 'country');
osc_set_preference('rewrite_search_region', 'region');
osc_set_preference('rewrite_search_city', 'city');
osc_set_preference('rewrite_search_city_area', 'cityarea');
osc_set_preference('rewrite_search_category', 'category');
osc_set_preference('rewrite_search_user', 'user');
osc_set_preference('rewrite_search_pattern', 'pattern');
osc_set_preference('rewrite_contact', 'contact');
osc_set_preference('rewrite_feed', 'feed');
osc_set_preference('rewrite_language', 'language');
osc_set_preference('rewrite_item_mark', 'item/mark');
osc_set_preference('rewrite_item_send_friend', 'item/send-friend');
osc_set_preference('rewrite_item_contact', 'item/contact');
osc_set_preference('rewrite_item_new', 'item/new');
osc_set_preference('rewrite_item_activate', 'item/activate');
osc_set_preference('rewrite_item_edit', 'item/edit');
osc_set_preference('rewrite_item_delete', 'item/delete');
osc_set_preference('rewrite_resource_delete', 'item/resource/delete');
osc_set_preference('rewrite_user_login', 'user/login');
osc_set_preference('rewrite_user_dashboard', 'user/dashboard');
osc_set_preference('rewrite_user_logout', 'user/logout');
osc_set_preference('rewrite_user_register', 'user/register');
osc_set_preference('rewrite_user_activate', 'user/activate');
osc_set_preference('rewrite_user_activate_alert', 'user/activate-alert');
osc_set_preference('rewrite_user_profile', 'user/profile');
osc_set_preference('rewrite_user_items', 'user/items');
osc_set_preference('rewrite_user_alerts', 'user/alerts');
osc_set_preference('rewrite_user_recover', 'user/recover');
osc_set_preference('rewrite_user_forgot', 'user/forgot');
osc_set_preference('rewrite_user_change_password', 'user/change-password');
osc_set_preference('rewrite_user_change_email', 'user/change-email');
osc_set_preference('rewrite_user_change_email_confirm', 'user/change-email-confirm');

osc_set_preference('last_version_check', time());
osc_set_preference('update_core_json', '');

$update_dt_expiration = sprintf('update %st_item as a
left join %st_category as b on b.pk_i_id = a.fk_i_category_id
set a.dt_expiration = date_add(a.dt_pub_date, INTERVAL b.i_expiration_days
DAY)
where b.i_expiration_days > 0', DB.TABLE_PREFIX, DB.TABLE_PREFIX );
$comm->query( $update_dt_expiration );

// we need populate location table stats
$rs = $comm->query( sprintf('SELECT pk_c_code FROM %st_country', DB.TABLE_PREFIX) );
$aCountry = $rs->result();
foreach($aCountry as $country) {
// insert into country_stats with i_num_items = 0
$comm->query( sprintf('INSERT INTO %st_country_stats (fk_c_country_code,
i_num_items) VALUES (\'%s\', 0)', DB.TABLE_PREFIX, $country['pk_c_code'] ) );
$rs = $comm->query( sprintf('SELECT pk_i_id FROM %st_region WHERE
fk_c_country_code = \'%s\'', DB.TABLE_PREFIX, $country['pk_c_code'] ) );
$aRegion = $rs->result();
foreach($aRegion as $region) {
// insert into region_stats with i_num_items = 0
$comm->query( sprintf('INSERT INTO %st_region_stats (fk_i_region_id,
i_num_items) VALUES (%s, 0)', DB.TABLE_PREFIX, $region['pk_i_id'] ) );
$rs = $comm->query( sprintf('SELECT pk_i_id FROM %st_city WHERE fk_i_region_id
= %s', DB.TABLE_PREFIX, $region['pk_i_id'] ) );
$aCity = $rs->result();
foreach($aCity as $city) {
// insert into city_stats with i_num_items = 0
$comm->query( sprintf('INSERT INTO %st_city_stats (fk_i_city_id,
i_num_items) VALUES (%s, 0)', DB.TABLE_PREFIX, $city['pk_i_id'] ) );
}
}
}
}

$url_location_stats = osc_admin_base_url(true)."?page=tools&action=locations";
$aMessages[] = '<p><b>.--('You need to calculate location stats, please go to admin
panel, tools, recalculate location stats or click').' <a href="'.
$url_location_stats.'" >' .--('here')' .</a></b></p>';

// update t_alerts - Search object serialized to json
$aAlerts = Alerts::newInstance()->findByType('HOURLY');
foreach($aAlerts as $hourly) {
convertAlert($hourly);
}
unset($aAlerts);

$aAlerts = Alerts::newInstance()->findByType('DAILY');

```

```

foreach($aAlerts as $daily) {
    convertAlert($daily);
}
unset($aAlerts);

$aAlerts = Alerts::newInstance()->findByType('WEEKLY');
foreach($aAlerts as $weekly) {
    convertAlert($weekly);
}
unset($aAlerts);

// UPDATE COUNTRY PROCESS (remove fk_c_locale)
$comm->query("CREATE TABLE ".DB_TABLE_PREFIX."t_country_aux (
pk_c_code CHAR(2) NOT NULL,
s_name VARCHAR(80) NOT NULL,

PRIMARY KEY (pk_c_code),
INDEX idx_s_name (s_name)
) ENGINE=InnoDB DEFAULT CHARACTER SET 'UTF8' COLLATE 'UTF8_GENERAL_CI');
$comm->query("SELECT * FROM ".DB_TABLE_PREFIX."t_country GROUP BY pk_c_code");
$countries = $rs->result();
foreach($countries as $c) {
    $comm->insert(DB_TABLE_PREFIX."t_country_aux", array('pk_c_code' => $c['pk_c_code'],
    's_name' => $c['s_name']));
}
$rs = $comm->query("SHOW CREATE TABLE ".DB_TABLE_PREFIX."t_city");
$rs = $rs->result();
foreach($rs[0] as $r) {
    if(preg_match_all('|CONSTRAINT \('[^']+\' FOREIGN KEY \('fk_c_country_code\' \)
REFERENCES \'.DB_TABLE_PREFIX.'t_country\' \('pk_c_code\'\)|', $r, $matches)) {
        foreach($matches[1] as $m) {
            $comm->query("ALTER TABLE \'.DB_TABLE_PREFIX.'t_city\' DROP FOREIGN KEY
\'. $m.'");
        }
    }
}
$rs = $comm->query("SHOW CREATE TABLE ".DB_TABLE_PREFIX."t_region");
$rs = $rs->result();
foreach($rs[0] as $r) {
    if(preg_match_all('|CONSTRAINT \('[^']+\' FOREIGN KEY \('fk_c_country_code\' \)
REFERENCES \'.DB_TABLE_PREFIX.'t_country\' \('pk_c_code\'\)|', $r, $matches)) {
        foreach($matches[1] as $m) {
            $comm->query("ALTER TABLE \'.DB_TABLE_PREFIX.'t_region\' DROP FOREIGN KEY
\'. $m.'");
        }
    }
}
$rs = $comm->query("SHOW CREATE TABLE ".DB_TABLE_PREFIX."t_country_stats");
$rs = $rs->result();
foreach($rs[0] as $r) {
    if(preg_match_all('|CONSTRAINT \('[^']+\' FOREIGN KEY \('fk_c_country_code\' \)
REFERENCES \'.DB_TABLE_PREFIX.'t_country\' \('pk_c_code\'\)|', $r, $matches)) {
        foreach($matches[1] as $m) {
            $comm->query("ALTER TABLE \'.DB_TABLE_PREFIX.'t_country_stats\' DROP
FOREIGN KEY \'. $m.'");
        }
    }
}
$rs = $comm->query("SHOW CREATE TABLE ".DB_TABLE_PREFIX."t_item_location");
$rs = $rs->result();
foreach($rs[0] as $r) {
    if(preg_match_all('|CONSTRAINT \('[^']+\' FOREIGN KEY \('fk_c_country_code\' \)
REFERENCES \'.DB_TABLE_PREFIX.'t_country\' \('pk_c_code\'\)|', $r, $matches)) {
        foreach($matches[1] as $m) {
            $comm->query("ALTER TABLE \'.DB_TABLE_PREFIX.'t_item_location\' DROP
FOREIGN KEY \'. $m.'");
        }
    }
}
$rs = $comm->query("SHOW CREATE TABLE ".DB_TABLE_PREFIX."t_user");
$rs = $rs->result();
foreach($rs[0] as $r) {
    if(preg_match_all('|CONSTRAINT \('[^']+\' FOREIGN KEY \('fk_c_country_code\' \)
REFERENCES \'.DB_TABLE_PREFIX.'t_country\' \('pk_c_code\'\)|', $r, $matches)) {
        foreach($matches[1] as $m) {
            $comm->query("ALTER TABLE \'.DB_TABLE_PREFIX.'t_user\' DROP FOREIGN KEY
\'. $m.'");
        }
    }
}
}
$comm->query("DROP TABLE ".DB_TABLE_PREFIX."t_country");
// hack
$comm->query("SET FOREIGN_KEY_CHECKS = 0");
$comm->query("RENAME TABLE \'.DB_TABLE_PREFIX.'t_country_aux\' TO \'.DB_TABLE_PREFIX
.'t_country'");
$comm->query("ALTER TABLE ".DB_TABLE_PREFIX."t_city ADD FOREIGN KEY (fk_c_country_code
) REFERENCES ".DB_TABLE_PREFIX."t_country (pk_c_code)");
$comm->query("ALTER TABLE ".DB_TABLE_PREFIX."t_region ADD FOREIGN KEY (
fk_c_country_code) REFERENCES ".DB_TABLE_PREFIX."t_country (pk_c_code)");
$comm->query("ALTER TABLE ".DB_TABLE_PREFIX."t_country_stats ADD FOREIGN KEY (
fk_c_country_code) REFERENCES ".DB_TABLE_PREFIX."t_country (pk_c_code)");
$comm->query("ALTER TABLE ".DB_TABLE_PREFIX."t_item_location ADD FOREIGN KEY (
fk_c_country_code) REFERENCES ".DB_TABLE_PREFIX."t_country (pk_c_code)");
$comm->query("ALTER TABLE ".DB_TABLE_PREFIX."t_user ADD FOREIGN KEY (fk_c_country_code
) REFERENCES ".DB_TABLE_PREFIX."t_country (pk_c_code)");
// hack
$comm->query("SET FOREIGN_KEY_CHECKS = 1");
}

```

```

if( osc_version() < 241) {
    $comm->query(sprintf("INSERT INTO %st_preference VALUES ('osclass', 'use.imagick',
        '0', 'BOOLEAN')", DB_TABLE_PREFIX));
}

if( osc_version() < 300) {
    $comm->query(sprintf("ALTER TABLE %st_user DROP s_pass_answer", DB_TABLE_PREFIX));
    $comm->query(sprintf("ALTER TABLE %st_user DROP s_pass_question", DB_TABLE_PREFIX));
    osc_set_preference('marketURL', 'http://market.osclass.org/api/');
    osc_set_preference('marketAllowExternalSources', '0', 'BOOLEAN');
}

if( osc_version() < 310) {
    $comm->query(sprintf("ALTER TABLE %st_pages ADD 's_meta' TEXT NULL", DB_TABLE_PREFIX
    ));
    $comm->query(sprintf("ALTER TABLE %st_pages ADD 'b_link' TINYINT(1) NOT NULL DEFAULT
    1", DB_TABLE_PREFIX));
    $comm->query(sprintf("UPDATE %st_alerts SET dt_date = '%s' ", DB_TABLE_PREFIX, date("Y
    -m-d H:i:s")));

    // remove files moved to controller folder
    @unlink( osc_base_path() . 'ajax.php' );
    @unlink( osc_base_path() . 'contact.php' );
    @unlink( osc_base_path() . 'custom.php' );
    @unlink( osc_base_path() . 'item.php' );
    @unlink( osc_base_path() . 'language.php' );
    @unlink( osc_base_path() . 'login.php' );
    @unlink( osc_base_path() . 'main.php' );
    @unlink( osc_base_path() . 'page.php' );
    @unlink( osc_base_path() . 'register.php' );
    @unlink( osc_base_path() . 'search.php' );
    @unlink( osc_base_path() . 'user-non-secure.php' );
    @unlink( osc_base_path() . 'user.php' );
    @unlink( osc_base_path() . 'readme.php' );

    @unlink( osc_lib_path() . 'osclass/plugins.php' );
    @unlink( osc_lib_path() . 'osclass/feeds.php' );

    $comm->query(sprintf('UPDATE %st_user t, (SELECT pk_i_id FROM %st_user) t1 SET t.
    s_username = t1.pk_i_id WHERE t.pk_i_id = t1.pk_i_id', DB_TABLE_PREFIX,
    DB_TABLE_PREFIX));
    osc_set_preference('username_blacklist', 'admin,user', 'osclass', 'STRING');
    osc_set_preference('rewrite_user_change_username', 'username/change');
    osc_set_preference('csrf_name', 'CSRF'.mt_rand(0,mt_getrandmax()));

    @mkdir( osc_uploads_path() . 'page-images' );
}

if( osc_version() < 320) {
    osc_set_preference('mailserver_mail_from', '');
    osc_set_preference('mailserver_name_from', '');
    osc_set_preference('seo_url_search_prefix', '');

    $comm->query(sprintf("ALTER TABLE %st_category ADD 'b_price_enabled' TINYINT(1) NOT
    NULL DEFAULT 1", DB_TABLE_PREFIX));

    osc_set_preference('subdomain_type', '');
    osc_set_preference('subdomain_host', '');
    // email_new_admin
    $comm->query(sprintf("INSERT INTO %st_pages (s_internal_name, b_indelible, dt_pub_date
    ) VALUES ('email_new_admin', 1, '%s')", DB_TABLE_PREFIX, date('Y-m-d H:i:s')));
    $comm->query(sprintf("INSERT INTO %st_pages_description (fk_i_pages_id,
    fk_c_locale_code, s_title, s_text) VALUES (%d, 'en-US', '{WEB.TITLE} - Success
    creating admin account!', '<p>Hi {ADMIN.NAME},</p><p>The admin of {WEB.LINK} has
    created an account for you,</p><ul><li>Username: {USERNAME}</li><li>Password: {
    PASSWORD}</li></ul><p>You can access the admin panel here {WEB.ADMIN.LINK}.</p><p>
    Thank you!</p><p>Regards,</p>')", DB_TABLE_PREFIX, $comm->insertedId()));

    osc_set_preference('warn_expiration', '0', 'osclass', 'INTEGER');

    $comm->query(sprintf("INSERT INTO %st_pages (s_internal_name, b_indelible, dt_pub_date
    ) VALUES ('email_warn_expiration', 1, '%s')", DB_TABLE_PREFIX, date('Y-m-d H:i:s
    ')));
    $comm->query(sprintf("INSERT INTO %st_pages_description (fk_i_pages_id,
    fk_c_locale_code, s_title, s_text) VALUES (%d, 'en-US', '{WEB.TITLE} - Your ad is
    about to expire', '<p>Hi {USER.NAME},</p><p>Your listing <a href=\"%{ITEM.URL}\">{
    ITEM.TITLE}</a> is about to expire at {WEB.LINK}.')", DB_TABLE_PREFIX, $comm->
    insertedId()));

    osc_set_preference('force_aspect_image', '0', 'osclass', 'BOOLEAN');
}

if( osc_version() < 321) {
    osc_calculate_location_slug( osc_subdomain_type() );
}

osc_changeVersionTo(322);

echo '<div class="well ui-rounded-corners separate-top-medium">';
echo '<p>...('Osclass &raquo; Updated correctly').</p>';
echo '<p>...('Osclass has been updated successfully. <a href="http://forums.osclass.org
    /">Need more help?</a>').</p>';
foreach($aMessages as $msg) {
    echo "<p>".$msg."</p>";
}
echo "</div>";

/**

```

```

* Convert alerts < 2.4, updating s_search with json encoded to based64.
*
* @param string $alert base64+serialized
*/
function convertAlert($alert)
{
    // decode search model
    $data = base64_decode($alert['s_search']);
    if (is_serialized($data)) { // don't attempt to unserialize data that wasn't
        serialized going in
        $data = unserialize($data);
        // if search model, convert alert
        if (get_class($data) == 'Search') {
            // get json
            $json = $data->toJson(true);
            // insert new alert with json
            $aCondition = array(
                's_email' => $alert['s_email'],
                'b_active' => $alert['b_active'],
                'e_type' => $alert['e_type']
            );
            if ($alert['fk_i_user_id'] != '') {
                $aCondition['fk_i_user_id'] = (int)$alert['fk_i_user_id'];
            }
            if ($alert['s_secret'] != '') {
                $aCondition['s_secret'] = $alert['s_secret'];
            }
            Alerts::newInstance()->update(array('s_search' => base64_encode($json)),
                $aCondition);
        }
    }
}

?>

<?php if ( !defined('ABS_PATH') ) exit('ABS_PATH is not loaded. Direct access is not allowed
. ');

/**
 * Oclass software for creating and publishing online classified advertising
 * platforms
 *
 * Copyright (C) 2012 OSCLASS
 *
 * This program is free software: you can redistribute it and/or modify it under the terms
 * of the GNU Affero General Public License as published by the Free Software Foundation,
 * either version 3 of the License, or (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful, but WITHOUT ANY
 * WARRANTY;
 * without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
 * PURPOSE.
 * See the GNU Affero General Public License for more details.
 *
 * You should have received a copy of the GNU Affero General Public
 * License along with this program. If not, see <http://www.gnu.org/licenses/>.
 */

Class UserActions
{
    var $is_admin;
    var $manager;

    function __construct($is_admin)
    {
        $this->is_admin = $is_admin;
        $this->manager = User::newInstance();
    }

    //add...
    function add()
    {
        $success = 0;
        $error = false;
        if( !$error && (osc_recaptcha_private_key() != '') && !$this->is_admin ) {
            if( !osc_check_recaptcha() ) {
                $error = 4;
            }
        }

        if( !$error && Params::getParam('s_password', false, false) == '' ) {
            $error = 6;
        }

        if( !$error && Params::getParam('s_password', false, false) != Params::getParam('
s_password2', false, false) ) {
            $error = 7;
        }

        $input = $this->prepareData(true);

        if( !$error && $input['s_name'] == '' ) {
            $error = 10;
        }

        if( !$error && !osc_validate_email($input['s_email']) ) {
            $error = 5;
        }
    }
}

```

```

$email_taken = $this->manager->findByEmail($input['s_email']);
if( !$error && $email_taken != null ) {
    osc_run_hook('register_email_taken', $input['s_email']);
    $error = 3;
}

if(!$error && $input['s_username']!= '') {
    $username_taken = $this->manager->findByUsername($input['s_username']);
    if( !$error && $username_taken != null ) {
        $error = 8;
    }
    if(osc_is_username_blacklisted($input['s_username'])) {
        $error = 9;
    }
}

// hook pre add or edit
osc_run_hook('pre-user-post');

if( is_numeric($error) && $error > 0) {
    osc_run_hook('user_register_failed', $error);
    return $error;
}

$this->manager->insert($input);
$userId = $this->manager->dao->insertedId();

if($input['s_username']== '') {
    $this->manager->update(
        array('s_username' => $userId)
        ,array('pk_i_id' => $userId)
    );
}

if ( is_array( Params::getParam('s_info') ) ) {
    foreach (Params::getParam('s_info') as $key => $value) {
        $this->manager->updateDescription($userId, $key, $value);
    }
}

Log::newInstance()->insertLog('user', 'add', $userId, $input['s_email'], $this->
    is_admin ? 'admin' : 'user', $this->is_admin ? osc_logged_admin_id() : $userId
);

// update items with s_contact.email the same as new user email
$aItems = Item::newInstance()->findByEmail( $input['s_email'] );
foreach( $aItems as $aux ) {
    if( Item::newInstance()->update(array('fk_i_user_id' => $userId, '
        s_contact_name' => $input['s_name']), array('pk_i_id' => $aux['pk_i_id']
    )) {
        $this->manager->increaseNumItems($userId);
    }
}

// update alerts user id with the same email
$aAlerts = Alerts::newInstance()->findByEmail( $input['s_email'] );
foreach( $aAlerts as $aux ) {
    Alerts::newInstance()->update(array('fk_i_user_id' => $userId), array('s_email
        ' => $input['s_email']));
}

$user = $this->manager->findByPrimaryKey($userId);

if( osc_notify_new_user() && !$this->is_admin ) {
    osc_run_hook('hook_email_admin_new_user', $user);
}

if( osc_user_validation_enabled() && !$this->is_admin ) {
    osc_run_hook('hook_email_user_validation', $user, $input);
    $success = 1;
} else {
    $this->manager->update(
        array('b_active' => '1')
        ,array('pk_i_id' => $userId)
    );
    $success = 2;
}

osc_run_hook('user_register_completed', $userId);
return $success;
}

//edit...
function edit($userId)
{
    $input = $this->prepareData(false);

    // hook pre add or edit
    osc_run_hook('pre-user-post');

    if($this->is_admin) {
        $user_email = $this->manager->findByEmail($input['s_email']);
        if(isset($user_email['pk_i_id']) && $user_email['pk_i_id']!= $userId) {
            return 3;
        }
    }

    if($input['s_name']== '') {

```

```

        return 10;
    }
    $this->manager->update($input, array('pk.i.id' => $userId));
    if($this->is_admin) {
        Item::newInstance()->update( array('s.contact.name' => $input['s.name'], '
            s.contact.email' => $input['s.email']), array('fk.i.user.id' => $userId) );
        ItemComment::newInstance()->update( array('s.author.name' => $input['s.name'],
            's.author.email' => $input['s.email']), array('fk.i.user.id' => $userId)
        );
        Alerts::newInstance()->update( array('s.email' => $input['s.email']), array('
            fk.i.user.id' => $userId) );
        Log::newInstance()->insertLog( 'user', 'edit', $userId, $input['s.email'],
            $this->is_admin ? 'admin' : 'user', $this->is_admin ? osc_logged_admin_id
            () : osc_logged_user_id() );
    } else {
        Item::newInstance()->update( array('s.contact.name' => $input['s.name'],
            array('fk.i.user.id' => $userId) );
        ItemComment::newInstance()->update( array('s.author.name' => $input['s.name'],
            array('fk.i.user.id' => $userId) );
        $user = $this->manager->findByPrimaryKey($userId);
        Log::newInstance()->insertLog('user', 'edit', $userId, $user['s.email'], $this
            ->is_admin ? 'admin' : 'user', $this->is_admin ? osc_logged_admin_id() :
            osc_logged_user_id() );
    }
    if(!$this->is_admin) {
        Session::newInstance()->_set('userName', $input['s.name']);
        $phone = ($input['s.phone.mobile']? $input['s.phone.mobile'] : $input['
            s.phone.land'];
        Session::newInstance()->_set('userPhone', $phone);
    }
    if ( is_array( Params::getParam('s.info') ) ) {
        foreach (Params::getParam('s.info') as $key => $value) {
            $this->manager->updateDescription($userId, $key, $value);
        }
    }
    osc_run_hook('user_edit_completed', $userId);
    if ( $this->is_admin ) {
        $iUpdated = 0;
        if( (Params::getParam("b.enabled") != '') && (Params::getParam("b.enabled") ==
            1) ) {
            $iUpdated += $this->manager->update( array('b.enabled' => 1), array('
                pk.i.id' => $userId) );
        } else {
            $iUpdated += $this->manager->update( array('b.enabled' => 0), array('
                pk.i.id' => $userId) );
        }
        if( (Params::getParam("b.active") != '') && (Params::getParam("b.active") ==
            1) ) {
            $iUpdated += $this->manager->update( array('b.active' => 1), array('
                pk.i.id' => $userId) );
        } else {
            $iUpdated += $this->manager->update( array('b.active' => 0), array('
                pk.i.id' => $userId) );
        }
        if($iUpdated > 0) {
            return 2;
        }
    }
    return 0;
}
function recover_password()
{
    $user = User::newInstance()->findByEmail( Params::getParam('s.email') );
    Session::newInstance()->_set( 'recover.time', time() );
    if ( ( osc_recaptcha_private_key() != '' ) ) {
        if( !osc_check_recaptcha() ) {
            return 2; // BREAK THE PROCESS, THE RECAPTCHA IS WRONG
        }
    }
    if( !$user || ($user['b.enabled'] == 0) ) {
        return 1;
    }
    $code = osc_genRandomPassword(30);
    $date = date('Y-m-d H:i:s');
    User::newInstance()->update(
        array('s.pass_code' => $code, 's.pass_date' => $date, 's.pass_ip' => $SERVER
            ['REMOTE_ADDR']),
        array('pk.i.id' => $user['pk.i.id']
    );
    $password_url = osc_forgot_user_password_confirm_url($user['pk.i.id'], $code);
    osc_run_hook('hook_email_user_forgot_password', $user, $password_url);
}

```

```

    return 0;
}

function prepareData($is_add)
{
    $input = array();

    if ( $is_add ) {
        $input['s_secret'] = osc_genRandomPassword();
        $input['dt_reg_date'] = date('Y-m-d H:i:s');
    } else {
        $input['dt_mod_date'] = date('Y-m-d H:i:s');
    }

    //only for administration, in the public website this two params are edited
    separately
    if ( $this->is_admin || $is_add ) {
        $input['s_email'] = Params::getParam('s_email');

        if( Params::getParam('s_password', false, false) != Params::getParam('s_password2', false, false) ) {
            return 1;
        }

        //if we want to change the password
        if( Params::getParam('s_password', false, false) != '' ) {
            $input['s_password'] = sha1( Params::getParam('s_password', false, false) );
        }
        $input['s_username'] = osc_sanitize_username(Params::getParam('s_username'));
    }

    $input['s_name'] = trim(Params::getParam('s_name'));
    $input['s_website'] = trim(Params::getParam('s_website'));
    $input['s_phone_land'] = trim(Params::getParam('s_phone_land'));
    $input['s_phone_mobile'] = trim(Params::getParam('s_phone_mobile'));

    //locations...
    $country = Country::newInstance()->findByCode( Params::getParam('countryId') );
    if(count($country) > 0) {
        $countryId = $country['pk_c_code'];
        $countryName = $country['s_name'];
    } else {
        $countryId = null;
        $countryName = Params::getParam('country');
    }

    if( intval( Params::getParam('regionId') ) ) {
        $region = Region::newInstance()->findByPrimaryKey( Params::getParam('regionId') );
        if( count($region) > 0 ) {
            $regionId = $region['pk_i_id'];
            $regionName = $region['s_name'];
        }
    } else {
        $regionId = null;
        $regionName = Params::getParam('region');
    }

    if( intval( Params::getParam('cityId') ) ) {
        $city = City::newInstance()->findByPrimaryKey( Params::getParam('cityId') );
        if( count($city) > 0 ) {
            $cityId = $city['pk_i_id'];
            $cityName = $city['s_name'];
        }
    } else {
        $cityId = null;
        $cityName = Params::getParam('city');
    }

    $input['fk_c_country_code'] = $countryId;
    $input['s_country'] = $countryName;
    $input['fk_i_region_id'] = $regionId;
    $input['s_region'] = $regionName;
    $input['fk_i_city_id'] = $cityId;
    $input['s_city'] = $cityName;
    $input['s_city_area'] = Params::getParam('cityArea');
    $input['s_address'] = Params::getParam('address');
    $input['s_zip'] = Params::getParam('zip');
    $input['d_coord_lat'] = (Params::getParam('d_coord_lat') != '') ? Params::getParam('d_coord_lat') : null;
    $input['d_coord_long'] = (Params::getParam('d_coord_long') != '') ? Params::getParam('d_coord_long') : null;
    $input['b_company'] = (Params::getParam('b_company') != '' && Params::getParam('b_company') != 0) ? 1 : 0;

    return($input);
}

public function activate($user_id)
{
    $user = $this->manager->findByPrimaryKey($user_id);

    if( !$user ) {
        return false;
    }

    $this->manager->update( array('b_active' => 1), array('pk_i_id' => $user_id) );
}

```



```

        if( !$this->is_admin ) {
            osc_run_hook('hook_email_admin_new_user', $user);
        }

        Log::newInstance()->insertLog('user', 'activate', $user_id, $user['s_email'],
            $this->is_admin ? 'admin' : 'user', $this->is_admin ? osc_logged_admin_id() :
            osc_logged_user_id());

        if( $user['b_enabled'] == 1 ) {
            $mItem = new ItemActions(true);
            $items = Item::newInstance()->findByUserID($user_id);
            foreach($items as $item) {
                $mItem->enable($item['pk_i_id']);
            }
        }
        osc_run_hook('activate_user', $user);

        return true;
    }

    public function deactivate($user_id)
    {
        $user = $this->manager->findByPrimaryKey($user_id);

        if( !$user ) {
            return false;
        }

        $this->manager->update( array('b_active' => 0), array('pk_i_id' => $user_id) );

        Log::newInstance()->insertLog('user', 'deactivate', $user_id, $user['s_email'],
            $this->is_admin ? 'admin' : 'user', $this->is_admin ? osc_logged_admin_id() :
            osc_logged_user_id());

        if( $user['b_enabled'] == 1 ) {
            $mItem = new ItemActions(true);
            $items = Item::newInstance()->findByUserID($user_id);
            foreach($items as $item) {
                $mItem->disable($item['pk_i_id']);
            }
        }
        osc_run_hook('deactivate_user', $user);

        return true;
    }

    public function enable($user_id)
    {
        $user = $this->manager->findByPrimaryKey($user_id);

        if( !$user ) {
            return false;
        }

        $this->manager->update( array('b_enabled' => 1), array('pk_i_id' => $user_id) );

        Log::newInstance()->insertLog('user', 'enable', $user_id, $user['s_email'], $this
            ->is_admin ? 'admin' : 'user', $this->is_admin ? osc_logged_admin_id() :
            osc_logged_user_id());

        if( $user['b_active'] == 1 ) {
            $mItem = new ItemActions(true);
            $items = Item::newInstance()->findByUserID($user_id);
            foreach($items as $item) {
                $mItem->enable($item['pk_i_id']);
            }
        }

        $this->fn_email_user_unsuspended($user);

        osc_run_hook('enable_user', $user);

        return true;
    }

    public function fn_email_user_unsuspended($user) {

        $account_name = $user['s_email'];
        $s_title = "Your Collab-Ads account has been unsuspended.";

        $web_link = '<a href="' . osc_base_url() . '">' . osc_page_title() . '</a>';
        $username = $user['s_name'];

        $s_text = "<p>Hi ". $username .",</p><p>Your account " . $account_name . " is no
            longer suspended. You may now be able to post ads again. Please observe site rules
            to prevent this from happening again.</p>"
            . "<p>Thank you!</p><p>Regards,</p><p>". $web_link . "</p>";

        $emailParams = array(
            'subject' => $s_title,
            'from' => osc_contact_email(),
            'to' => $user['s_email'],
            'to_name' => $user['s_name'],
            'body' => $s_text,
            'alt_body' => $s_text
        );
    }

```

```

osc_sendMail($emailParams);
//return 1;
}

public function disable($user_id)
{
    $user = $this->manager->findByPrimaryKey($user_id);

    if( !$user ) {
        return false;
    }

    $this->manager->update( array('b.enabled' => 0), array('pk.i.id' => $user_id) );

    Log::newInstance()->insertLog('user', 'disable', $user_id, $user['s_email'], $this
->is_admin ? 'admin' : 'user', $this->is_admin ? osc_logged_admin_id() :
    osc_logged_user_id());

    if( $user['b_active'] == 1 ) {
        $mItem = new ItemActions(true);
        $items = Item::newInstance()->findByPrimaryKey($user_id);
        foreach($items as $item) {
            $mItem->disable($item['pk.i.id']);
        }
    }

    $this->fn_email_user_suspended($user);
    osc_run_hook('disable-user', $user);

    return true;
}

public function fn_email_user_suspended($user) {
    $s_title = "Your Collab-Ads account has been suspended.";

    $web_link = '<a href="' . osc_base_url() . '">' . osc_page_title() . '</a>';
    $username = $user['s_name'];

    $s_text = "<p>Hi ". $username .",</p><p>Your account ". $account_name . " has been
suspended. You may have submitted an offensive, spam, duplicate, "
    . " or misclassified ad; or for abusing a user. Send us back an email if you
    think this was a mistake.</p>"
    . "<p>Thank you!</p><p>Regards,</p><p>". $web_link . "</p>";

    $emailParams = array(
        'subject' => $s_title ,
        'from' => osc_contact_email(),
        'to' => $user['s_email'],
        'to_name' => $user['s_name'],
        'body' => $s_text ,
        'alt_body' => $s_text
    );

    osc_sendMail($emailParams);
    return 1;
}

public function resend_activation($user_id)
{
    $user = $this->manager->findByPrimaryKey($user_id);
    $input['s_secret'] = $user['s_secret'];

    if( !$user || $user['b_active']==1 ) {
        return 0;
    }

    if( osc_user_validation_enabled() ) {
        osc_run_hook('hook_email_user_validation', $user, $input);
        return 1;
    }

    return 0;
}

public function bootstrap_login($user_id)
{
    $user = User::newInstance()->findByPrimaryKey($user_id);

    if( !$user ) {
        return 0;
    }

    if( !$user['b_active'] ) {
        return 1;
    }

    if( !$user['b_enabled'] ) {
        return 2;
    }

    //we are logged in... let's go!
    Session::newInstance()->set('userId', $user['pk.i.id']);
    Session::newInstance()->set('userName', $user['s_name']);
}

```

```

        Session::newInstance()->_set('userEmail', $user['s_email']);
        $phone = ($user['s_phone_mobile']) ? $user['s_phone_mobile'] : $user['s_phone_land
        '];
        Session::newInstance()->_set('userPhone', $phone);

        return 3;
    }
}

?>

<?php if ( ! defined('ABS_PATH')) exit('ABS_PATH is not loaded. Direct access is not allowed
. ');
/**
 * Oclass      software for creating and publishing online classified advertising
 *             platforms
 *
 * Copyright (C) 2012 OSCLASS
 *
 * This program is free software: you can redistribute it and/or modify it under the terms
 * of the GNU Affero General Public License as published by the Free Software Foundation,
 * either version 3 of the License, or (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful, but WITHOUT ANY
 * WARRANTY;
 * without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
 * PURPOSE.
 * See the GNU Affero General Public License for more details.
 *
 * You should have received a copy of the GNU Affero General Public
 * License along with this program. If not, see <http://www.gnu.org/licenses/>.
 */

$user_agents = array(
    'MSIE ([0-9].*)$',
    '^ Mozilla/[0-9].([\^c][\^o][\^m].)$',
    '^ Mozilla/5.*Gecko',
    '^ Mozilla/5.(PLAYSTATION.)',
    'Safari',
    'Omni',
    'iCab',
    'Opera',
    'Konqueror',
    'AOL',
    'MSN',
    'WebTV',
    'BlackBerry.*'
);

?>

<?php

/**
 * Oclass      software for creating and publishing online classified
 *             advertising platforms
 *
 * Copyright (C) 2012 OSCLASS
 *
 * This program is free software: you can redistribute it and/or
 * modify it under the terms of the GNU Affero General Public License
 * as published by the Free Software Foundation, either version 3 of
 * the License, or (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful, but
 * WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 * GNU Affero General Public License for more details.
 *
 * You should have received a copy of the GNU Affero General Public
 * License along with this program. If not, see <http://www.gnu.org/licenses/>.
 */

/**
 * check if the item is expired
 */
function osc_isExpired($dt_expiration) {
    $now      = date("YmdHis");

    $dt_expiration = str_replace(' ', '', $dt_expiration);
    $dt_expiration = str_replace('-', '', $dt_expiration);
    $dt_expiration = str_replace(':', '', $dt_expiration);

    if ($dt_expiration > $now) {
        return false;
    } else {
        return true;
    }
}

/**
 * Remove resources from disk
 * @param <type> $id
 * @param boolean $admin
 * @return boolean
 */
function osc_deleteResource( $id , $admin) {

```

```

if( defined('DEMO') ) {
    return false;
}
if( is_array( $id ) ){
    $id = $id[0];
}
$resource = ItemResource::newInstance()->findByPrimaryKey($id);
if( !is_null($resource) ){
    Log::newInstance()->insertLog('item', 'delete resource', $resource['pk_i_id'], $id,
        $admin?'admin':'user', $admin ? osc_logged_admin_id() : osc_logged_user_id());

    $backtracel = '';
    foreach(debug_backtrace() as $k=>$v){
        if($v['function'] == "include" || $v['function'] == "include_once" || $v['function']
            == "require_once" || $v['function'] == "require"){
            $backtracel .= "#".$k." ".$v['function']."' (".$v['args'][0].") called@ [".$v['file']
                .":". $v['line']."' / ";
        }else{
            $backtracel .= "#".$k." ".$v['function']."' called@ [".$v['file']:".":". $v['line']
                .":". $v['line']."' / ";
        }
    }

    Log::newInstance()->insertLog('item', 'delete resource backtrace', $resource['pk_i_id'],
        $backtracel, $admin?'admin':'user', $admin ? osc_logged_admin_id() :
        osc_logged_user_id());

    @unlink(osc_base_path() . $resource['s_path'] . $resource['pk_i_id'] . "." . $resource['s_extension']);
    @unlink(osc_base_path() . $resource['s_path'] . $resource['pk_i_id'] . "_original." . $resource['s_extension']);
    @unlink(osc_base_path() . $resource['s_path'] . $resource['pk_i_id'] . "_thumbnail." . $resource['s_extension']);
    @unlink(osc_base_path() . $resource['s_path'] . $resource['pk_i_id'] . "_preview." . $resource['s_extension']);
    osc_run_hook('delete_resource', $resource);
}
}

/**
 * Tries to delete the directory recursively.
 * @return true on success.
 */
function osc_deleteDir($path) {
    if(strpos($path, "../")!==false) {
        return false;
    }

    if (!is_dir($path)) {
        return false;
    }

    $fd = @opendir($path);
    if (!$fd) {
        return false;
    }

    while ($file = @readdir($fd)) {
        if ($file != '.' && $file != '..') {
            if (!is_dir($path . '/' . $file)) {
                if (!@unlink($path . '/' . $file)) {
                    closedir($fd);
                    return false;
                } else {
                    osc_deleteDir($path . '/' . $file);
                }
            } else {
                osc_deleteDir($path . '/' . $file);
            }
        }
    }
    closedir($fd);

    return @rmdir($path);
}

/**
 * Unpack a ZIP file into the specific path in the second parameter.
 * @DEPRECATED : TO BE REMOVED IN 3.3
 * @return true on success.
 */
function osc_packageExtract($zipPath, $path) {
    if(strpos($path, "../")!==false) {
        return false;
    }

    if(!file_exists($path)) {
        if (!@mkdir($path, 0666)) {
            return false;
        }
    }

    @chmod($path, 0777);

    $zip = new ZipArchive;
    if ($zip->open($zipPath) === true) {
        $zip->extractTo($path);
        $zip->close();
        return true;
    }
}

```

```

    } else {
        return false;
    }
}

/**
 * Fix the problem of symbolics links in the path of the file
 *
 * @param string $file The filename of plugin.
 * @return string The fixed path of a plugin.
 */
function osc_plugin_path($file) {
    // Sanitize windows paths and duplicated slashes
    $file = preg_replace('|/+|', '/', str_replace('\\', '/', $file));
    $plugin_path = preg_replace('|/+|', '/', str_replace('\\', '/', osc_plugins_path()));
    $file = $plugin_path . preg_replace('#^.*oc-content/plugins/#', '', $file);
    return $file;
}

/**
 * Fix the problem of symbolics links in the path of the file
 *
 * @param string $file The filename of plugin.
 * @return string The fixed path of a plugin.
 */
function osc_plugin_url($file) {
    // Sanitize windows paths and duplicated slashes
    $dir = preg_replace('|/+|', '/', str_replace('\\', '/', dirname($file)));
    $dir = osc_base_url() . 'oc-content/plugins/' . preg_replace('#^.*oc-content/plugins/#', '', $dir) . "/";
    return $dir;
}

/**
 * Fix the problem of symbolics links in the path of the file
 *
 * @param string $file The filename of plugin.
 * @return string The fixed path of a plugin.
 */
function osc_plugin_folder($file) {
    // Sanitize windows paths and duplicated slashes
    $dir = preg_replace('|/+|', '/', str_replace('\\', '/', dirname($file)));
    $dir = preg_replace('#^.*oc-content/plugins/#', '', $dir) . "/";
    return $dir;
}

/**
 * Serialize the data (usefull at plugins activation)
 * @return the data serialized
 */
function osc_serialize($data) {
    if (!is_serialized($data)) {
        if (is_array($data) || is_object($data)) {
            return serialize($data);
        }
    }
    return $data;
}

/**
 * Unserialize the data (usefull at plugins activation)
 * @return the data unserialized
 */
function osc_unserialize($data) {
    if (is_serialized($data)) { // don't attempt to unserialize data that wasn't serialized
        going in
        return @unserialize($data);
    }
    return $data;
}

/**
 * Checks is $data is serialized or not
 * @return bool False if not serialized and true if it was.
 */
function is_serialized($data) {
    // if it isn't a string, it isn't serialized
    if (!is_string($data))
        return false;
    $data = trim($data);
    if ('N;' == $data)
        return true;
    if (!preg_match('/^\s*([adObis]):/', $data, $badions))
        return false;
    switch ($badions[1]) {
        case 'a' :
        case 'O' :
        case 's' :
            if (preg_match("/^\s*{$badions[1]}:[0-9]+\s*;\s*$/", $data))
                return true;
            break;
        case 'b' :
        case 'i' :
        case 'd' :
            if (preg_match("/^\s*{$badions[1]}:[0-9.E-]+\s*;\s*$/", $data))
                return true;
    }
}

```

```

        break;
    }
    return false;
}

/**
 * VERY BASIC
 * Perform a POST request, so we could launch fake-cron calls and other core-system calls
 * without annoying the user
 */
function osc_doRequest($url, $_data) {
    if (function_exists('fsockopen')) {
        $data = http_build_query($_data);

        // parse the given URL
        $url = parse_url($url);

        // extract host and path:
        $host = $url['host'];
        $path = $url['path'];

        // open a socket connection on port 80
        // use localhost in case of issues with NATs (hairpinning)
        $fp = @fsockopen($host, 80);

        if ($fp !== false) {
            $out = "POST $path HTTP/1.1\r\n";
            $out .= "Host: $host\r\n";
            $out .= "Referer: Osclass (v.". osc_version() .")\r\n";
            $out .= "Content-type: application/x-www-form-urlencoded\r\n";
            $out .= "Content-Length: ".strlen($data)."\r\n";
            $out .= "Connection: Close\r\n\r\n";
            $out .= "$data";
            fwrite($fp, $out);
            fclose($fp);
        }
    }
}

function osc_sendMail($params) {
    // DO NOT send mail if it's a demo
    if( defined('DEMO') ) {
        return false;
    }

    $mail = new PHPMailer(true);
    $mail->ClearAddresses();
    $mail->ClearAllRecipients();
    $mail->ClearAttachments();
    $mail->ClearBCCs();
    $mail->ClearCCs();
    $mail->ClearCustomHeaders();
    $mail->ClearReplyTos();

    $mail = osc_apply_filter('init_send_mail', $mail);

    if( osc_mailserver_pop() ) {
        require_once osc_lib_path() . 'phpmailer/class.pop3.php';
        $pop = new POP3();

        $pop3_host = osc_mailserver_host();
        if( array_key_exists('host', $params) ) {
            $pop3_host = $params['host'];
        }

        $pop3_port = osc_mailserver_port();
        if( array_key_exists('port', $params) ) {
            $pop3_port = $params['port'];
        }

        $pop3_username = osc_mailserver_username();
        if( array_key_exists('username', $params) ) {
            $pop3_username = $params['username'];
        }

        $pop3_password = osc_mailserver_password();
        if( array_key_exists('password', $params) ) {
            $pop3_password = $params['password'];
        }

        $pop->Authorise($pop3_host, $pop3_port, 30, $pop3_username, $pop3_password, 0);
    }

    if( osc_mailserver_auth() ) {
        $mail->IsSMTP();
        $mail->SMTPAuth = true;
    } else if( osc_mailserver_pop() ) {
        $mail->IsSMTP();
    }

    $smtpSecure = osc_mailserver_ssl();
    if( array_key_exists('password', $params) ) {
        $smtpSecure = $params['ssl'];
    }
    if( $smtpSecure != '' ) {
        $mail->SMTPSecure = $smtpSecure;
    }

    $smtpUsername = osc_mailserver_username();

```

```

if( array_key_exists('username', $params) ) {
    $smtpUsername = $params['username'];
}
if( $smtpUsername != '' ) {
    $mail->Username = $smtpUsername;
}

$smtpPassword = osc_mailserver_password();
if( array_key_exists('password', $params) ) {
    $smtpPassword = $params['password'];
}
if( $smtpPassword != '' ) {
    $mail->Password = $smtpPassword;
}

$smtpHost = osc_mailserver_host();
if( array_key_exists('host', $params) ) {
    $smtpHost = $params['host'];
}
if( $smtpHost != '' ) {
    $mail->Host = $smtpHost;
}

$smtpPort = osc_mailserver_port();
if( array_key_exists('port', $params) ) {
    $smtpPort = $params['port'];
}
if( $smtpPort != '' ) {
    $mail->Port = $smtpPort;
}

$from = osc_mailserver_mail_from();
if(empty($from)) {
    $from = 'osclass@' . osc_get_domain();
}
if( array_key_exists('from', $params) ) {
    $from = $params['from'];
}

$from_name = osc_mailserver_name_from();
if(empty($from_name)) {
    $from_name = osc_page_title();
}
if( array_key_exists('from_name', $params) ) {
    $from_name = $params['from_name'];
}

$mail->From = osc_apply_filter('mail_from', $from);
$mail->FromName = osc_apply_filter('mail_from_name', $from_name);

$to = $params['to'];
$to_name = '';
if( array_key_exists('to_name', $params) ) {
    $to_name = $params['to_name'];
}

if( !is_array($to) ) {
    $to = array($to => $to_name);
}

foreach($to as $to_email => $to_name) {
    try {
        $mail->addAddress($to_email, $to_name);
    } catch (phpmailerException $e) {
        continue;
    }
}

if( array_key_exists('add_bcc', $params) ) {
    if( !is_array($params['add_bcc']) && $params['add_bcc'] != '' ) {
        $params['add_bcc'] = array($params['add_bcc']);
    }

    foreach($params['add_bcc'] as $bcc) {
        try {
            $mail->AddBCC($bcc);
        } catch (phpmailerException $e) {
            continue;
        }
    }
}

if( array_key_exists('reply_to', $params) ) {
    try {
        $mail->AddReplyTo($params['reply_to']);
    } catch (phpmailerException $e) {
        //continue;
    }
}

$mail->Subject = $params['subject'];
$mail->Body = $params['body'];

if( array_key_exists('attachment', $params) ) {
    if( !is_array($params['attachment']) ) {
        $params['attachment'] = array($params['attachment']);
    }

    foreach($params['attachment'] as $attachment) {

```

```

        try {
            $mail->AddAttachment($attachment);
        } catch (phpmailerException $e) {
            continue;
        }
    }
}

$mail->CharSet = 'utf-8';
$mail->IsHTML(true);

$mail = osc_apply_filter('pre-send-mail', $mail);

// send email!
try {
    $mail->Send();
} catch (phpmailerException $e) {
    return false;
}

return true;
}

function osc_mailBeauty($text, $params) {

    $text = str_ireplace($params[0], $params[1], $text);
    $keywords = array(
        '{WEB_URL}',
        '{WEB_TITLE}',
        '{WEB_LINK}',
        '{CURRENT_DATE}',
        '{HOUR}',
        '{IP_ADDRESS}'
    );
    $rwords = array(
        osc_base_url(),
        osc_page_title(),
        '<a href="' . osc_base_url() . '">' . osc_page_title() . '</a>',
        date('Y-m-d H:i:s'),
        date('H:i'),
        $_SERVER['REMOTE_ADDR']
    );
    $text = str_ireplace($keywords, $rwords, $text);

    return $text;
}

function osc_copy($source, $dest, $options=array('folderPermission'=>0755,'filePermission'=>0755)) {
    $result = true;
    if (is_file($source)) {
        if ($dest[strlen($dest)-1]!='/') {
            if (!file_exists($dest)) {
                cmfcDirectory::makeAll($dest, $options['folderPermission'], true);
            }
            $__dest=$dest."/".basename($source);
        } else {
            $__dest=$dest;
        }
        if(function_exists('copy')) {
            $result = @copy($source, $__dest);
        } else {
            $result=osc_copyemz($source, $__dest);
        }
        @chmod($__dest, $options['filePermission']);
    } elseif(is_dir($source)) {
        if ($dest[strlen($dest)-1]!='/') {
            if ($source[strlen($source)-1]!='/') {
                //Copy only contents
            } else {
                //Change parent itself and its contents
                $dest=$dest.basename($source);
                @mkdir($dest);
                @chmod($dest, $options['filePermission']);
            }
        } else {
            if ($source[strlen($source)-1]!='/') {
                //Copy parent directory with new name and all its content
                @mkdir($dest, $options['folderPermission']);
                @chmod($dest, $options['filePermission']);
            } else {
                //Copy parent directory with new name and all its content
                @mkdir($dest, $options['folderPermission']);
                @chmod($dest, $options['filePermission']);
            }
        }
    }

    $dirHandle=opendir($source);
    $result = true;
    while($file=readdir($dirHandle)) {
        if($file!="." && $file!="..") {
            if(!is_dir($source."/". $file)) {
                $__dest=$dest."/". $file;
            } else {
                $__dest=$dest."/". $file;
            }
            //echo "$source/$file ||| $__dest<br />";
        }
    }
}

```



```

        $data = osc_copy($source."/". $file , $_dest , $options);
        if($data==false) {
            $result = false;
        }
    }
    closedir($dirHandle);
} else {
    $result=true;
}
return $result;
}

function osc_copyemz($file1,$file2){
    $contentx=@file_get_contents($file1);
    $openedfile = fopen($file2, "w");
    fwrite($openedfile, $contentx);
    fclose($openedfile);
    if ($contentx == FALSE) {
        $status=false;
    } else {
        $status=true;
    }
    return $status;
}

/**
 * Dump osclass database into path file
 *
 * @param type $path
 * @param type $file
 * @return type
 */
function osc_dbdump($path, $file) {

    require_once LIB_PATH . 'osclass/model/Dump.php';
    if ( !is_writable($path) ) return -4;
    if($path == '') return -1;

    //checking connection
    $dump = Dump::newInstance();
    if (!$dump) return -2;

    $path .= $file;
    $result = $dump->showTables();

    if(!$result) {
        $_str = '';
        $_str .= '/* no tables in ' . DB_NAME . ' */';
        $_str .= "\n";

        $f = fopen($path, "a");
        fwrite($f, $_str);
        fclose($f);

        return -3;
    }

    $_str = '';
    $_str .= '/* OSCLASS MYSQL Autobackup (' . date('Y-m-d H:i:s') . ') */';
    $_str .= "\n";

    $f = fopen($path, "a");
    fwrite($f, $_str);
    fclose($f);

    $tables = array();
    foreach($result as $_table) {
        $tableName = current($_table);
        $tables[$tableName] = $tableName;
    }

    $tables_order = array('t_locale', 't_country', 't_currency', 't_region', 't_city', 't_city_area', 't_widget', 't_admin', 't_user', 't_user_description', 't_category', 't_category_description', 't_category_stats', 't_item', 't_item_description', 't_item_location', 't_item_stats', 't_item_resource', 't_item_comment', 't_preference', 't_user_preferences', 't_pages', 't_pages_description', 't_plugin_category', 't_cron', 't_alerts', 't_keywords', 't_meta_fields', 't_meta_categories', 't_item_meta');
    // Backup default Osclass tables in order, so no problem when importing them back
    foreach($tables_order as $table) {
        if(array_key_exists(DB_TABLE_PREFIX . $table, $tables)) {
            $dump->table_structure($path, DB_TABLE_PREFIX . $table);
            $dump->table_data($path, DB_TABLE_PREFIX . $table);
            unset($tables[DB_TABLE_PREFIX . $table]);
        }
    }

    // Backup the rest of tables
    foreach($tables as $table) {
        $dump->table_structure($path, $table);
        $dump->table_data($path, $table);
    }

    return 1;
}

```

```

// -----
/**
 * Returns true if there is curl on system environment
 *
 * @return type
 */
function testCurl() {
    if ( ! function_exists( 'curl_init' ) || ! function_exists( 'curl_exec' ) )
        return false;

    return true;
}

/**
 * Returns true if there is fsockopen on system environment
 *
 * @return type
 */
function testFsockopen() {
    if ( ! function_exists( 'fsockopen' ) )
        return false;

    return true;
}

/**
 * IF http-chunked-decode not exist implement here
 * @since 3.0
 */
if( !function_exists('http_chunked_decode') ) {
    /**
     * dechunk an http 'transfer-encoding: chunked' message
     *
     * @param string $chunk the encoded message
     * @return string the decoded message. If $chunk wasn't encoded properly it will be
     * returned unmodified.
     */
    function http_chunked_decode($chunk) {
        $pos = 0;
        $len = strlen($chunk);
        $dechunk = null;
        while(( $pos < $len)
            && ($chunkLenHex = substr($chunk,$pos, ($newlineAt = strpos($chunk,"\n",$pos+1))-
                $pos)))
        {
            if (! is_hex($chunkLenHex)) {
                trigger_error('Value is not properly chunk encoded', E_USER_WARNING);
                return $chunk;
            }

            $pos = $newlineAt + 1;
            $chunkLen = hexdec(rtrim($chunkLenHex,"\r\n"));
            $dechunk .= substr($chunk, $pos, $chunkLen);
            $pos = strpos($chunk, "\n", $pos + $chunkLen) + 1;
        }
        return $dechunk;
    }
}

/**
 * determine if a string can represent a number in hexadecimal
 *
 * @since 3.0
 * @param string $hex
 * @return boolean true if the string is a hex, otherwise false
 */
function is_hex($hex) {
    // regex is for weenies
    $hex = strtolower(trim(ltrim($hex,"0")));
    if (empty($hex)) { $hex = 0; };
    $dec = hexdec($hex);
    return ($hex == dechex($dec));
}

/**
 * Process response and return headers and body
 *
 * @since 3.0
 * @param type $content
 * @return type
 */
function processResponse($content)
{
    $res = explode("\r\n\r\n", $content);
    $headers = $res[0];
    $body = isset($res[1]) ? $res[1] : '';

    if (!is_string($headers)) {
        return array();
    }

    return array('headers' => $headers, 'body' => $body);
}

/**
 * Parse headers and return into array format
 *

```

```

* @param type $headers
* @return type
*/
function processHeaders($headers)
{
    $headers = str_replace("\r\n", "\n", $headers);
    $headers = preg_replace('/\n[ \t]/', ' ', $headers);
    $headers = explode("\n", $headers);
    $tmpHeaders = $headers;
    $headers = array();

    foreach ($tmpHeaders as $aux) {
        if (preg_match('/^(.*):\s(.*)$/ ', $aux, $matches)) {
            $headers[strtolower($matches[1])] = $matches[2];
        }
    }
    return $headers;
}

/**
* Download file using fsockopen
*
* @since 3.0
* @param type $sourceFile
* @param type $fileout
*/
function download_fsockopen($sourceFile, $fileout = null)
{
    // parse URL
    $aUrl = parse_url($sourceFile);
    $host = $aUrl['host'];
    if ('localhost' == strtolower($host))
        $host = '127.0.0.1';

    $link = $aUrl['path'] . ( isset($aUrl['query']) ? '?' . $aUrl['query'] : '' );

    if (empty($link))
        $link .= '/';

    $fp = @fsockopen($host, 80, $errno, $errstr, 30);
    if (!$fp) {
        return false;
    } else {
        $ua = $_SERVER['HTTP_USER_AGENT'] . ' Osclass (v.' . osc_version() . ')';
        $out = "GET $link HTTP/1.1\r\n";
        $out .= "Host: $host\r\n";
        $out .= "User-Agent: $ua\r\n";
        $out .= "Connection: Close\r\n\r\n";
        $out .= "\r\n";
        fwrite($fp, $out);

        $contents = '';
        while (!feof($fp)) {
            $contents .= fgets($fp, 1024);
        }

        fclose($fp);

        // check redirections ?
        // if (redirections) then do request again
        $aResult = processResponse($contents);
        $headers = processHeaders($aResult['headers']);

        $location = @$headers['location'];
        if (isset($location) && $location != "") {
            $aUrl = parse_url($headers['location']);

            $host = $aUrl['host'];
            if ('localhost' == strtolower($host))
                $host = '127.0.0.1';

            $requestPath = $aUrl['path'] . ( isset($aUrl['query']) ? '?' . $aUrl['query'] : '' );

            if (empty($requestPath))
                $requestPath .= '/';

            download_fsockopen($host, $requestPath, $fileout);
        } else {
            $body = $aResult['body'];
            $transferEncoding = @$headers['transfer-encoding'];
            if ($transferEncoding == 'chunked') {
                $body = http_chunked_decode($aResult['body']);
            }
            if ($fileout != null) {
                $ff = @fopen($fileout, 'w+');
                if ($ff !== FALSE) {
                    fwrite($ff, $body);
                    fclose($ff);
                    return true;
                } else {
                    return false;
                }
            } else {
                return $body;
            }
        }
    }
}
}

```

```

function osc_downloadFile($sourceFile, $downloadedFile)
{
    if (strpos($downloadedFile, "../")!==false) {
        return false;
    }

    if ( testCurl() ) {
        @set_time_limit(0);
        $fp = @fopen (osc_content_path() . 'downloads/' . $downloadedFile, 'w+');
        if($fp) {
            $ch = curl_init($sourceFile);
            @curl_setopt($ch, CURLOPT_TIMEOUT, 50);
            curl_setopt($ch, CURLOPT_USERAGENT, $_SERVER['HTTP_USER_AGENT'] . ' Osclass (v.' .
                osc_version() . ')');
            curl_setopt($ch, CURLOPT_FILE, $fp);
            @curl_setopt($ch, CURLOPT_FOLLOWLOCATION, true);
            curl_setopt($ch, CURLOPT_REFERER, osc_base_url());
            curl_exec($ch);
            curl_close($ch);
            fclose($fp);
            return true;
        } else {
            return false;
        }
    } else if (testFsockopen()) { // test curl/fsockopen
        $downloadedFile = osc_content_path() . 'downloads/' . $downloadedFile;
        download_fsockopen($sourceFile, $downloadedFile);
        return true;
    }
    return false;
}

function osc_file_get_contents($url)
{
    if( testCurl() ) {
        $ch = curl_init();
        curl_setopt($ch, CURLOPT_URL, $url);
        curl_setopt($ch, CURLOPT_USERAGENT, $_SERVER['HTTP_USER_AGENT'] . ' Osclass (v.' .
            osc_version() . ')');
        if( !defined('CURLOPT_RETURNTRANSFER') ) define('CURLOPT_RETURNTRANSFER', 1);
        @curl_setopt($ch, CURLOPT_FOLLOWLOCATION, 1);
        curl_setopt($ch, CURLOPT_REFERER, osc_base_url());
        curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
        $data = curl_exec($ch);
        curl_close($ch);
    } else if( testFsockopen() ) {
        $data = download_fsockopen($url);
    }
    return $data;
}

// -----
/**
 * Check if we loaded some specific module of apache
 *
 * @param string $mod
 *
 * @return bool
 */
function apache_mod_loaded($mod) {
    if(function_exists('apache_get_modules')) {
        $modules = apache_get_modules();
        if(in_array($mod, $modules)) { return true; }
    } else if(function_exists('phpinfo')) {
        ob_start();
        phpinfo(INFO_MODULES);
        $content = ob_get_contents();
        ob_end_clean();
        if(strpos($content, $mod)!==FALSE) { return true; }
    }
    return false;
}

/**
 * Change version to param number
 *
 * @param mixed version
 */
function osc_changeVersionTo($version = null) {
    if($version != null) {
        osc_set_preference('version', $version);
        osc_reset_preferences();
    }
}

function strip_slashes_extended($array) {
    if(is_array($array)) {
        foreach($array as $k => &$v) {
            $v = strip_slashes_extended($v);
        }
    } else {
        $array = stripslashes($array);
    }
    return $array;
}

/**

```

```

* Unzip's a specified ZIP file to a location
*
* @param string $file Full path of the zip file
* @param string $to Full path where it is going to be unzipped
* @return int
* 0 - destination folder not writable (or not exist and cannot be created)
* 1 - everything was OK
* 2 - zip is empty
* -1 : file could not be created (or error reading the file from the zip)
*/
function osc_unzip_file($file, $to) {
    if (strpos($to, "../")!==false) {
        return 0;
    }

    if (!file_exists($to)) {
        if (!mkdir($to, 0766)) {
            return 0;
        }
    }

    @chmod($to, 0777);

    if (!is_writable($to)) {
        return 0;
    }

    if (class_exists('ZipArchive')) {
        return _unzip_file_ziparchive($file, $to);
    }

    // if ZipArchive class doesn't exist, we use PclZip
    return _unzip_file_pclzip($file, $to);
}

/**
 * We assume that the $to path is correct and can be written. It unzips an archive using the
 * PclZip library.
 *
 * @param string $file Full path of the zip file
 * @param string $to Full path where it is going to be unzipped
 * @return int
 */
function _unzip_file_ziparchive($file, $to) {
    if (strpos($to, "../")!==false) {
        return 0;
    }

    $zip = new ZipArchive();
    $zipopen = $zip->open($file, 4);

    if ($zipopen !== true) {
        return 2;
    }

    // The zip is empty
    if ($zip->numFiles==0) {
        return 2;
    }

    for ($i = 0; $i < $zip->numFiles; $i++) {
        $file = $zip->statIndex($i);

        if (!$file) {
            return -1;
        }

        if (substr($file['name'], 0, 9) === '_MACOSX/') {
            continue;
        }

        if (substr($file['name'], -1) === '/') {
            @mkdir($to . $file['name'], 0777);
            continue;
        }

        $content = $zip->getFromIndex($i);
        if ($content === false) {
            return -1;
        }

        $fp = @fopen($to . $file['name'], 'w');
        if (!$fp) {
            return -1;
        }

        @fwrite($fp, $content);
        @fclose($fp);
    }

    $zip->close();

    return 1;
}

/**
 * We assume that the $to path is correct and can be written. It unzips an archive using the
 * PclZip library.
 *

```

```

* @param string $zip_file Full path of the zip file
* @param string $to Full path where it is going to be unzipped
* @return int
*/
function _unzip_file_pclzip($zip_file, $to) {
    if (strpos($to, "../") !== false) {
        return false;
    }

    // first, we load the library
    require_once LIB_PATH . 'pclzip/pclzip.lib.php';

    $archive = new PclZip($zip_file);
    if (($files = $archive->extract(PCLZIP_OPT_EXTRACT_AS_STRING)) === false) {
        return 2;
    }

    // check if the zip is not empty
    if (count($files) == 0) {
        return 2;
    }

    // Extract the files from the zip
    foreach ($files as $file) {
        if (substr($file['filename'], 0, 9) === '_MACOSX/') {
            continue;
        }

        if ($file['folder']) {
            @mkdir($to . $file['filename'], 0777);
            continue;
        }

        $fp = @fopen($to . $file['filename'], 'w');
        if (!$fp) {
            return -1;
        }

        @fwrite($fp, $file['content']);
        @fclose($fp);
    }

    return 1;
}

/**
 * Common interface to zip a specified folder to a file using ziparchive or pclzip
 *
 * @param string $archive_folder full path of the folder
 * @param string $archive_name full path of the destination zip file
 * @return int
 */
function osc_zip_folder($archive_folder, $archive_name) {
    if (strpos($archive_folder, "../") !== false || strpos($archive_name, "../") !== false) {
        return false;
    }

    if (class_exists('ZipArchive')) {
        return _zip_folder_ziparchive($archive_folder, $archive_name);
    }
    // if ZipArchive class doesn't exist, we use PclZip
    return _zip_folder_pclzip($archive_folder, $archive_name);
}

/**
 * Zips a specified folder to a file
 *
 * @param string $archive_folder full path of the folder
 * @param string $archive_name full path of the destination zip file
 * @return int
 */
function _zip_folder_ziparchive($archive_folder, $archive_name) {
    if (strpos($archive_folder, "../") !== false || strpos($archive_name, "../") !== false) {
        return false;
    }

    $zip = new ZipArchive;
    if ($zip -> open($archive_name, ZipArchive::CREATE) === TRUE) {
        $dir = preg_replace('/[\\/]2}/', '/', $archive_folder."/");

        $dirs = array($dir);
        while (count($dirs) > 0) {
            $dir = current($dirs);
            $zip -> addEmptyDir(str_replace(ABS_PATH, '', $dir));

            $dh = opendir($dir);
            while (false !== ($_file = readdir($dh))) {
                if ($_file != '.' && $_file != '..' && strpos($_file, 'Osclass_backup.') === FALSE) {
                    if (is_file($dir . $_file)) {
                        $zip -> addFile($dir . $_file, str_replace(ABS_PATH, '', $dir . $_file));
                    } elseif (is_dir($dir . $_file)) {
                        $dirs[] = $dir . $_file."/";
                    }
                }
            }
        }
        closedir($dh);
    }
}

```

```

        array_shift($dirs);
    }
    $zip -> close();
    return true;
} else {
    return false;
}
}

/**
 * Zips a specified folder to a file
 *
 * @param string $archive_folder full path of the folder
 * @param string $archive_name full path of the destination zip file
 * @return int
 */
function _zip_folder_pclzip($archive_folder, $archive_name) {
    if(strpos($archive_folder, "../")!==false || strpos($archive_name, "../")!==false) {
        return false;
    }

    // first, we load the library
    require_once LIB_PATH . 'pclzip/pclzip.lib.php';

    $zip = new PclZip($archive_name);
    if($zip) {
        $dir = preg_replace('/[\\\/]{2,}/', '/', $archive_folder."/");

        $v_dir = osc_base_path();
        $v_remove = $v_dir;

        // To support windows and the C: root you need to add the
        // following 3 lines, should be ignored on linux
        if (substr($v_dir, 1,1) == ':') {
            $v_remove = substr($v_dir, 2);
        }
        $v_list = $zip->create($dir, PCLZIP_OPT_REMOVE_PATH, $v_remove);
        if ($v_list == 0) {
            return false;
        }
        return true;
    } else {
        return false;
    }
}

function osc_check_recaptcha() {

    require_once osc_lib_path() . 'recaptchalib.php';
    if ( Params::getParam("recaptcha_challenge_field") != '' ) {
        $resp = recaptcha_check_answer (osc_recaptcha_private_key()
            ,$_SERVER["REMOTE_ADDR"]
            ,Params::getParam("recaptcha_challenge_field")
            ,Params::getParam("recaptcha_response_field"));

        return $resp->is_valid;
    }

    return false;
}

function osc_check_dir_writable( $dir = ABS_PATH ) {
    if(strpos($dir, "../")!==false) {
        return false;
    }

    clearstatcache();
    if ($dh = opendir($dir)) {
        while (($file = readdir($dh)) !== false) {
            if ($file!="." && $file!="..") {
                if(is_dir(str_replace("//", "/", $dir . "/" . $file))) {
                    if(str_replace("//", "/", $dir)==(ABS_PATH . "oc-content/themes")) {
                        if($file=="modern" || $file=="index.php") {
                            $res = osc_check_dir_writable( str_replace("//", "/", $dir . "/" .
                                $file));
                            if(!$res) { return false; };
                        }
                    } else if(str_replace("//", "/", $dir)==(ABS_PATH . "oc-content/plugins"))
                    {
                        if($file=="google_maps" || $file=="google_analytics" || $file=="index.
                            php") {
                            $res = osc_check_dir_writable( str_replace("//", "/", $dir . "/" .
                                $file));
                            if(!$res) { return false; };
                        }
                    } else if(str_replace("//", "/", $dir)==(ABS_PATH . "oc-content/languages
                        ")) {
                        if($file=="en-US" || $file=="index.php") {
                            $res = osc_check_dir_writable( str_replace("//", "/", $dir . "/" .
                                $file));
                            if(!$res) { return false; };
                        }
                    } else if(str_replace("//", "/", $dir)==(ABS_PATH . "oc-content/downloads
                        ")) {
                    } else if(str_replace("//", "/", $dir)==(osc_uploads_path())) {
                    } else {
                        $res = osc_check_dir_writable( str_replace("//", "/", $dir . "/" .

```

```

        $file));
        if (!$res) { return false; };
    } else {
        return is_writable( str_replace("//", "/", $dir . "/" . $file));
    }
}
}
closedir($dh);
}
return true;
}

function osc_change_permissions( $dir = ABS_PATH ) {
    if (strpos($dir, "../")!==false) {
        return false;
    }

    clearstatcache();
    if ($dh = opendir($dir)) {
        while (($file = readdir($dh)) !== false) {
            if ($file!="." && $file!=".." && substr($file,0,1)!=".") {
                if (is_dir(str_replace("//", "/", $dir . "/" . $file))) {
                    if (!is_writable(str_replace("//", "/", $dir . "/" . $file))) {
                        $res = @chmod( str_replace("//", "/", $dir . "/" . $file), 0777);
                        if (!$res) { return false; };
                    }
                    if (str_replace("//", "/", $dir)==(ABS_PATH . "oc-content/themes")) {
                        if ($file=="modern" || $file=="index.php") {
                            $res = osc_change_permissions( str_replace("//", "/", $dir . "/" . $file));
                            if (!$res) { return false; };
                        }
                    } else if (str_replace("//", "/", $dir)==(ABS_PATH . "oc-content/plugins")) {
                        if ($file=="google_maps" || $file=="google_analytics" || $file=="index.php") {
                            $res = osc_change_permissions( str_replace("//", "/", $dir . "/" . $file));
                            if (!$res) { return false; };
                        }
                    } else if (str_replace("//", "/", $dir)==(ABS_PATH . "oc-content/languages")) {
                        if ($file=="en_US" || $file=="index.php") {
                            $res = osc_change_permissions( str_replace("//", "/", $dir . "/" . $file));
                            if (!$res) { return false; };
                        }
                    } else if (str_replace("//", "/", $dir)==(ABS_PATH . "oc-content/downloads")) {
                    } else if (str_replace("//", "/", $dir)==(osc_uploads_path())) {
                    } else {
                        $res = osc_change_permissions( str_replace("//", "/", $dir . "/" . $file));
                        if (!$res) { return false; };
                    }
                } else {
                    if (!is_writable(str_replace("//", "/", $dir . "/" . $file))) {
                        return @chmod( str_replace("//", "/", $dir . "/" . $file), 0777);
                    } else {
                        return true;
                    }
                }
            }
        }
    }
    closedir($dh);
}
return true;
}

function osc_save_permissions( $dir = ABS_PATH ) {
    if (strpos($dir, "../")!==false) {
        return false;
    }

    $perms = array();
    $perms[$dir] = fileperms($dir);
    clearstatcache();
    if ($dh = opendir($dir)) {
        while (($file = readdir($dh)) !== false) {
            if ($file!="." && $file!="..") {
                if (is_dir(str_replace("//", "/", $dir . "/" . $file))) {
                    $res = osc_save_permissions( str_replace("//", "/", $dir . "/" . $file));
                    foreach($res as $k => $v) {
                        $perms[$k] = $v;
                    }
                } else {
                    $perms[str_replace("//", "/", $dir . "/" . $file)] = fileperms(
                        str_replace("//", "/", $dir . "/" . $file));
                }
            }
        }
    }
    closedir($dh);
}
return $perms;
}

```



```

function osc_prepare_price($price) {
    return $price/1000000;
}

/**
 * Recursive glob function
 *
 * @param string $pattern
 * @param string $flags
 * @param string $path
 * @return array of files
 */
function rglob($pattern, $flags = 0, $path = '') {
    if (!$path && ($dir = dirname($pattern)) != '.') {
        if ($dir == '\\\' || $dir == '/') $dir = '';
        return rglob(basename($pattern), $flags, $dir . '/');
    }
    $paths = glob($path . '*', GLOB_ONLYDIR | GLOB_NOSORT);
    $files = glob($path . $pattern, $flags);
    foreach ($paths as $p) $files = array_merge($files, rglob($pattern, $flags, $p . '/'));
    return $files;
}

/**
 * Market util functions
 */

function osc_check_plugin_update($update_uri, $version = null) {
    $suri = _get_market_url('plugins', $update_uri);
    if ($suri != false) {
        return _need_update($suri, $version);
    }
    return false;
}

function osc_check_theme_update($update_uri, $version = null) {
    $suri = _get_market_url('themes', $update_uri);
    if ($suri != false) {
        return _need_update($suri, $version);
    }
    return false;
}

function osc_check_language_update($update_uri, $version = null) {
    $suri = _get_market_url('languages', $update_uri);
    if ($suri != false) {
        if (false===($json=@osc_file_get_contents($suri))) {
            return false;
        } else {
            $data = json_decode($json, true);
            if (isset($data['s_version'])) {
                $result = version_compare2($version, $data['s_version']);
                if ($result == -1) {
                    // market have a newer version of this language
                    $result = version_compare2($data['s_version'], OSCLASS_VERSION);
                    if ($result == 0 || $result == -1) {
                        // market version is compatible with current osclass version
                        return true;
                    }
                }
            }
        }
    }
    return false;
}

function _get_market_url($type, $update_uri) {
    if ($update_uri == null) {
        return false;
    }

    if (in_array($type, array('plugins', 'themes', 'languages'))) {
        $suri = '';
        if (stripos($update_uri, "http://")===FALSE) {
            // OSCLASS OFFICIAL REPOSITORY
            $suri = osc_market_url($type, $update_uri);
        } else {
            // THIRD PARTY REPOSITORY
            if (!osc_market_external_sources()) {
                return false;
            }
            $suri = $update_uri;
        }
        return $suri;
    } else {
        return false;
    }
}

function _need_update($suri, $version) {
    if (false===($json=@osc_file_get_contents($suri))) {
        return false;
    } else {
        $data = json_decode($json, true);
        if (isset($data['s_version'])) {
            $result = version_compare2($data['s_version'], $version);
            if ($result == 1) {

```

```

        return true;
    }
}
}
// END -- Market util functions

/**
 * Returns
 * 0 if both are equal,
 * 1 if A > B, and
 * -1 if B < A.
 *
 * @param type $a -> from market
 * @param type $b -> installed version
 * @return type
 */
function version_compare2($a, $b)
{
    $a = explode(".", rtrim($a, ".0")); //Split version into pieces and remove trailing .0
    $b = explode(".", rtrim($b, ".0")); //Split version into pieces and remove trailing .0
    foreach ($a as $depth => $aVal)
    { //Iterate over each piece of A
        if (isset($b[$depth]))
        { //If B matches A to this depth, compare the values
            if ($aVal > $b[$depth]) return 1; //Return A > B
            else if ($aVal < $b[$depth]) return -1; //Return B > A
            //An equal result is inconclusive at this point
        }
        else
        { //If B does not match A to this depth, then A comes after B in sort order
            return 1; //so return A > B
        }
    }
    //At this point, we know that to the depth that A and B extend to, they are equivalent.
    //Either the loop ended because A is shorter than B, or both are equal.
    return (count($a) < count($b)) ? -1 : 0;
}

function _recursive_category_stats(&$aux, &$categoryTotal)
{
    $count_items = Item::newInstance()->numItems($aux, true, true);
    if(is_array($aux['categories'])) {
        foreach($aux['categories'] as &$cat) {
            $count_items += _recursive_category_stats($cat, $categoryTotal);
        }
    }
    $categoryTotal[$aux['pk_i-id']] = $count_items;
    return $count_items;
}

/**
 * Update category stats
 *
 * @param string $update_uri
 * @return boolean
 */
function osc_update_cat_stats() {
    $categoryTotal = array();
    $aCategories = Category::newInstance()->toTreeAll();

    foreach($aCategories as &$category) {
        if( is_null($category['fk_i-parent_id']) ) {
            _recursive_category_stats($category, $categoryTotal);
        }
    }

    $sql = 'REPLACE INTO '.DB_TABLE_PREFIX.'t_category_stats (fk_i-category-id, i-num-items)
VALUES ';
    $aValues = array();
    foreach($categoryTotal as $k => $v) {
        array_push($aValues, "($k, $v)" );
    }
    $sql .= implode(',', $aValues);
    $result = CategoryStats::newInstance()->dao->query($sql);
}

/**
 * Recount items for a given a category id
 *
 * @param int $id
 */
function osc_update_cat_stats_id($id)
{
    // get sub categorias
    $aCategories = Category::newInstance()->findSubcategories($id);
    $categoryTotal = 0;
    $category = Category::newInstance()->findByPrimaryKey($id);

    if( count($aCategories) > 0 ) {
        // sumar items de la categor a
        foreach($aCategories as $subcategory) {
            $total = Item::newInstance()->numItems($subcategory, true, true);
            $categoryTotal += $total;
        }
        $categoryTotal += Item::newInstance()->numItems($category, true, true);
    } else {
        $total = Item::newInstance()->numItems($category, true, true);
    }
}

```

```

        $categoryTotal += $total;
    }

    $sql = 'REPLACE INTO '.DB.TABLE.PREFIX.'t_category_stats (fk_i_category_id , i_num_items)
VALUES ' ;
    $sql .= " (".Sid.", ". $categoryTotal.")";
    $result = CategoryStats::newInstance()->dao->query($sql);

    if($category['fk_i_parent_id']!=0) {
        osc_update_cat_stats_id($category['fk_i_parent_id']);
    }
}

/**
 * Update locations stats. I moved this function from cron.daily.php:update_location_stats
 *
 * @since 3.1
 */
function osc_update_location_stats($force = false , $limit = 1000) {

    $loctmp = LocationsTmp::newInstance();
    $workToDo = $loctmp->count();

    if( $workToDo > 0 ) {
        // there are work to do
        if( $limit=='auto' ) {
            $total_cities = City::newInstance()->count();
            $limit = max(1000, ceil($total_cities/22));
        }
        $aLocations = $loctmp->getLocations($limit);
        foreach($aLocations as $location) {
            $id = $location['id_location'];
            $type = $location['e_type'];
            $data = 0;
            // update locations stats
            switch ( $type ) {
                case 'COUNTRY':
                    $numItems = CountryStats::newInstance()->calculateNumItems( $id );
                    $data = CountryStats::newInstance()->setNumItems($id , $numItems);
                    unset($numItems);
                    break;
                case 'REGION' :
                    $numItems = RegionStats::newInstance()->calculateNumItems( $id );
                    $data = RegionStats::newInstance()->setNumItems($id , $numItems);
                    unset($numItems);
                    break;
                case 'CITY' :
                    $numItems = CityStats::newInstance()->calculateNumItems( $id );
                    $data = CityStats::newInstance()->setNumItems($id , $numItems);
                    unset($numItems);
                    break;
                default :
                    break;
            }
            if($data >= 0) {
                $loctmp->delete(array('e_type' => $location['e_type'], 'id_location' =>
                    $location['id_location']));
            }
        }
    }
    } else if($force) {
        // we need populate location tmp table
        $aCountry = Country::newInstance()->listAll();

        foreach($aCountry as $country) {
            $aRegionsCountry = Region::newInstance()->findByCountry( $country['pk_c_code'] );
            $loctmp->insert(array('id_location' => $country['pk_c_code'], 'e_type' => 'COUNTRY'
                ));
            foreach($aRegionsCountry as $region) {
                $aCitiesRegion = City::newInstance()->findByRegion( $region['pk_i_id'] );
                $loctmp->insert(array('id_location' => $region['pk_i_id'], 'e_type' => 'REGION'
                    ));
                $batchCities = array();
                foreach($aCitiesRegion as $city) {
                    $batchCities[] = $city['pk_i_id'];
                }
                unset($aCitiesRegion);
                $loctmp->batchInsert($batchCities, 'CITY');
                unset($batchCities);
            } unset($aRegionsCountry);
        } unset($aCountry);
        osc_set_preference('location_todo', LocationsTmp::newInstance()->count());
    }
    return LocationsTmp::newInstance()->count();
}

/** Translate current categories to new locale
 *
 * @since 3.2.1
 */
function osc_translate_categories($locale) {
    $old_locale = Session::newInstance()->get('adminLocale');
    Session::newInstance()->_set('adminLocale', $locale);
    Translation::newInstance()->load(osc_translations_path().$locale.'/core.mo', 'cat_'.$locale);
    $catManager = Category::newInstance();
    $old_categories = $catManager->_findNameIDByLocale($old_locale);
    $tmp_categories = $catManager->_findNameIDByLocale($locale);
}

```

```

    foreach($tmp_categories as $category) {
        $new_categories[$category['pk_i_id']] = $category['s_name'];
    }
    unset($tmp_categories);
    foreach($old_categories as $category) {
        if(!isset($new_categories[$category['pk_i_id']])) {
            $fieldsDescription['s_name'] = --($category['s_name'], 'cat_'. $locale);
            $fieldsDescription['s_description'] = '';
            $fieldsDescription['fk_i_category_id'] = $category['pk_i_id'];
            $fieldsDescription['fk_c_locale_code'] = $locale;
            $slug_tmp = $slug = osc_sanitizeString(osc_apply_filter('slug', $fieldsDescription
                ['s_name']));
            $slug_unique = 1;
            while(true) {
                if(!$catManager->findBySlug($slug)) {
                    break;
                } else {
                    $slug = $slug_tmp . "-" . $slug_unique;
                    $slug_unique++;
                }
            }
            $fieldsDescription['s_slug'] = $slug;
            $catManager->insertDescription($fieldsDescription);
        }
    }
    Session::newInstance()->.set('adminLocale', $old_locale);
}

function get_ip() {
    if( !empty($_SERVER['HTTP_CLIENT_IP']) ) {
        return $_SERVER['HTTP_CLIENT_IP'];
    }

    if( !empty($_SERVER['HTTP_X_FORWARDED_FOR']) ) {
        $ip_array = explode(',', $_SERVER['HTTP_X_FORWARDED_FOR']);
        foreach($ip_array as $ip) {
            return trim($ip);
        }
    }

    return $_SERVER['REMOTE_ADDR'];
}

/*****
 * CSRFGUARD functions *
 *****/
function osc_csrfguard_generate_token() {
    $token_name = Session::newInstance()->.get('token_name');
    if($token_name!='' && Session::newInstance()->.get($token_name)!='') {
        return array($token_name, Session::newInstance()->.get($token_name));
    }
    $unique_token_name = osc_csrf_name()."-" . mt_rand(0, mt_getrandmax());
    if(function_exists("hash_algos") and in_array("sha512", hash_algos())) {
        $token = hash("sha512", mt_rand(0, mt_getrandmax()));
    } else {
        $token = '';
        for ($i=0; $i<128; ++$i) {
            $r=mt_rand(0,35);
            if ($r<26) {
                $c=chr(ord('a')+$r);
            } else {
                $c=chr(ord('0')+$r-26);
            }
            $token.=$c;
        }
    }
    Session::newInstance()->.set('token_name', $unique_token_name);
    Session::newInstance()->.set($unique_token_name, $token);
    return array($unique_token_name, $token);
}

function osc_csrfguard_validate_token($unique_form_name, $token_value) {
    $name = Session::newInstance()->.get('token_name');
    $token = Session::newInstance()->.get($unique_form_name);
    if($name==$unique_form_name && $token==$token_value) {
        return true;
    } else {
        return false;
    }
    return $result;
}

function osc_csrfguard_replace_forms($form_data_html) {
    $count = preg_match_all("</form(.*)>/is", $form_data_html, $matches, PREG_SET_ORDER);
    if(is_array($matches)) {
        foreach ($matches as $m) {
            if (strpos($m[1], "nocsrf")!==false) { continue; }
            $form_data_html=str_replace($m[0], "<form{$m[1]}>" . osc_csrf_token_form(),
                $form_data_html);
        }
    }
    return $form_data_html;
}
}

```

```

function osc_csrfguard_inject() {
    $data = ob_get_clean();
    $data = osc_csrfguard_replace_forms($data);
    echo $data;
}

function osc_csrfguard_start() {
    ob_start();
    $functions = osc_apply_filter('shutdown_functions', array('osc_csrfguard_inject'));
    foreach($functions as $f) {
        register_shutdown_function($f);
    }
}

function osc_redirect_to($url) {
    if(ob_get_length()>0) {
        ob_end_flush();
    }
    header("Location: ".$url);
    exit;
}

function osc_calculate_location_slug($type) {
    $field = 'pk_i_id';
    switch($type) {
        case 'country':
            $manager = Country::newInstance();
            $field = 'pk_c_code';
            break;
        case 'region':
            $manager = Region::newInstance();
            break;
        case 'city':
            $manager = City::newInstance();
            break;
        default:
            return false;
            break;
    }
    $locations = $manager->listByEmptySlug();
    $locations_changed = 0;
    foreach($locations as $location) {
        $slug_tmp = $slug = osc_sanitizeString($location['s_name']);
        $slug_unique = 1;
        while(true) {
            $location_slug = $manager->findBySlug($slug);
            if(!isset($location_slug[$field])) {
                break;
            } else {
                $slug = $slug_tmp . '-' . $slug_unique;
                $slug_unique++;
            }
        }
        $locations_changed += $manager->update(array('s_slug' => $slug), array($field =>
            $location[$field]));
    }

    return $locations_changed;
}

function osc_prune_array(&$input) {
    foreach ($input as $key => &$value) {
        if(is_array($value)) {
            osc_prune_array($value);
            if(empty($input[$key])) {
                unset($input[$key]);
            }
        } else if($value===' ' || $value===false || $value===null) {
            unset($input[$key]);
        }
    }
}

?>

<?php if ( ! defined('ABS_PATH')) exit('ABS_PATH is not loaded. Direct access is not allowed
. ');

/**
 * Oclass      software for creating and publishing online classified advertising
 *              platforms
 *
 * Copyright (C) 2012 OSCLASS
 *
 * This program is free software: you can redistribute it and/or modify it under the terms
 * of the GNU Affero General Public License as published by the Free Software Foundation,
 * either version 3 of the License, or (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful, but WITHOUT ANY
 * WARRANTY;
 * without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
 * PURPOSE.
 * See the GNU Affero General Public License for more details.
 *
 * You should have received a copy of the GNU Affero General Public
 * License along with this program. If not, see <http://www.gnu.org/licenses/>.

```

```

*/
class WebThemes extends Themes
{
    private static $instance;

    private $pages = array( '404',
                            'contact',
                            'alert-form',
                            'custom',
                            'footer',
                            'functions',
                            'head',
                            'header',
                            'inc.search',
                            'index',
                            'item-contact',
                            'item-edit',
                            'item-post',
                            'item-send-friend',
                            'item',
                            'main',
                            'page',
                            'search',
                            'search_gallery',
                            'search_list',
                            'user-alerts',
                            'user-change_email',
                            'user-change_password',
                            'user-dashboard',
                            'user-forgot_password',
                            'user-items',
                            'user-login',
                            'user-profile',
                            'user-recover',
                            'user-register',
                            );

    public static function newInstance()
    {
        if(!self::$instance instanceof self) {
            self::$instance = new self;
        }
        return self::$instance;
    }

    public function __construct()
    {
        parent::__construct();
        $this->path = osc_themes_path();

        if( Params::getParam('theme') != '' && Session::newInstance()->_get('adminId') !=
            '' ) {
            $this->setCurrentTheme( Params::getParam('theme') );
        } else {
            $this->setCurrentTheme( osc_theme() );
        }

        $functions_path = $this->getCurrentThemePath() . 'functions.php';
        if( file_exists($functions_path) ) {
            require_once $functions_path;
        }
    }

    public function setCurrentThemePath()
    {
        if ( file_exists( $this->path . $this->theme . '/' ) ) {
            $this->theme_exists = true;
            $this->theme_path = $this->path . $this->theme . '/';
        } else {
            $this->theme_exists = false;
            $this->theme_path = osc_lib_path() . 'osclass/gui/';
        }
    }

    public function setCurrentThemeUrl()
    {
        if ( $this->theme_exists ) {
            $this->theme_url = osc_base_url() . str_replace(osc_base_path(), '', $this->
                theme_path);
        } else {
            $this->theme_url = osc_base_url() . 'oc-includes/osclass/gui/';
        }
    }

    /* PUBLIC */
    public function setPath($path)
    {
        if( file_exists($path) ) {
            $this->path = $path;
            return true;
        }

        return false;
    }

    public function setCurrentTheme($theme)
    {
        $this->theme = $theme;
    }
}

```

```

        $this->setCurrentThemePath();
        $this->setCurrentThemeUrl();
    }

    public function setGuiTheme()
    {
        $this->theme = '';

        $this->theme_exists = false;
        $this->theme_path = osc_lib_path() . 'osclass/gui/';
        $this->theme_url = osc_base_url() . 'oc-includes/osclass/gui/';

        $functions_path = $this->getCurrentThemePath() . 'functions.php';
        if( file_exists($functions_path) ) {
            require_once $functions_path;
        }
    }

    /**
     * This function returns an array of themes (those copied in the oc-content/themes
     * folder)
     * @return <type>
     */
    public function getListThemes()
    {
        $themes = array();
        $dir = opendir( $this->path );
        while ( $file = readdir($dir) ) {
            if (preg_match('/^[a-zA-Z0-9_]+$/ ', $file)) {
                $themes[] = $file;
            }
        }
        closedir($dir);
        return $themes;
    }

    /**
     * @param <type> $theme
     * @return <type>
     */
    function loadThemeInfo($theme)
    {
        $path = $this->path . $theme . '/index.php';
        if( !file_exists($path) ) {
            return false;
        }

        // NEW CODE FOR THEME INFO
        $s_info = file_get_contents($path);
        $info = array();
        if( preg_match('|Theme Name:([^\r\t\n]*)|i', $s_info, $match) ) {
            $info['name'] = trim($match[1]);
        } else {
            $info['name'] = "";
        }

        if( preg_match('|Theme URI:([^\r\t\n]*)|i', $s_info, $match) ) {
            $info['theme_uri'] = trim($match[1]);
        } else {
            $info['theme_uri'] = "";
        }

        if( preg_match('|Theme update URI:([^\r\t\n]*)|i', $s_info, $match) ) {
            $info['theme_update_uri'] = trim($match[1]);
        } else {
            $info['theme_update_uri'] = "";
        }

        if( preg_match('|Description:([^\r\t\n]*)|i', $s_info, $match) ) {
            $info['description'] = trim($match[1]);
        } else {
            $info['description'] = "";
        }

        if( preg_match('|Version:([^\r\t\n]*)|i', $s_info, $match) ) {
            $info['version'] = trim($match[1]);
        } else {
            $info['version'] = "";
        }

        if( preg_match('|Author:([^\r\t\n]*)|i', $s_info, $match) ) {
            $info['author_name'] = trim($match[1]);
        } else {
            $info['author_name'] = "";
        }

        if( preg_match('|Author URI:([^\r\t\n]*)|i', $s_info, $match) ) {
            $info['author_url'] = trim($match[1]);
        } else {
            $info['author_url'] = "";
        }

        if( preg_match('|Widgets:([^\r\t\n]*)|i', $s_info, $match) ) {
            $info['locations'] = explode(", ", str_replace(" ", "", $match[1]));
        } else {
            $info['locations'] = array();
        }
        $info['filename'] = $path;
    }

```

```

        $info['int_name'] = $theme;

        if($info['name']!= '') {
            return $info;
        }

        // OLD CODE INFO
        require_once $path;
        $fxName = $theme . '_theme-info';
        if (!function_exists($fxName)) {
            return false;
        }
        $result = call_user_func($fxName);
        $result['int_name'] = $theme;

        return $result;
    }

    function isValidPage($internal_name)
    {
        return !in_array($internal_name, $this->pages);
    }

    function getAvailableTemplates($theme = null)
    {
        if($theme==null) { $theme = $this->theme; };

        $templates = array();
        $dir = opendir( $this->path . $theme . "/" );
        while ( $file = readdir($dir) ) {
            if (preg_match('/^template-[a-zA-Z0-9-\.\_]+$/ ', $file)) {
                $templates[] = $file;
            }
        }
        closedir($dir);
        return $templates;
    }

}

}

/* file end: ./oc-includes/osclass/WebThemes.php */

<?php
// meta tag robots
osc_add_hook('header', 'bender_nofollow_construct');
bender_add_body_class('error not-found');
osc_current_web_theme_path('header.php');
?>
<div class="flashmessage-404">
<h1><?php _e(" Sorry but I can't find the page you're looking for", 'bender') ; ?></h1>

<p><?php _e(" Let us help you, we have got a few tips for you to find it.", 'bender') ;
?></p>
<ul>
<li>
<?php _e("<strong>Search</strong> for it:", 'bender') ; ?>
<form action="<?php echo osc_base_url(true) ; ?>" method="get" class="search">
<input type="hidden" name="page" value="search" />
<fieldset class="main">
<input type="text" name="sPattern" id="query" value="" />
<button type="submit" class="ui-button ui-button-middle"><?php _e('Search
', 'bender') ; ?></button>
</fieldset>
</form>
</li>
<li><?php _e("<strong>Look</strong> for it in the most popular categories.", 'bender')
; ?>
<div class="categories">
<?php osc_goto_first_category() ; ?>
<?php while ( osc_has_categories() ) { ?>
<h2><a class="category" <?php echo osc_category_slug() ; ?>" href="<?
php echo osc_search_category_url() ; ?>"><?php echo
osc_category_name() ; ?></a> <span><?php echo
osc_category_total_items() ; ?></h2>
<?php if ( osc_count_subcategories() > 0 ) { ?>
<?php while ( osc_has_subcategories() ) { ?>
<?php if( osc_category_total_items() > 0 ) { ?>
<h3><a class="category" <?php echo osc_category_slug() ;
?>" href="<?php echo osc_search_category_url() ;
?>"><?php echo osc_category_name() ; ?></a> <span><?
php echo osc_category_total_items() ; ?></h3>
<?php } ?>
<?php } ?>
<?php } ?>
</div>
<div class="clear"></div>
</li>
</ul>
</div>
<?php osc_current_web_theme_path('footer.php') ; ?>

<?php
/*
*           Osclass      software for creating and publishing online classified
*                       advertising platforms
*
*/

```



```

*
*                                     Copyright (C) 2013 OSCLASS
*
*      This program is free software: you can redistribute it and/or
*      modify it under the terms of the GNU Affero General Public License
*      as published by the Free Software Foundation, either version 3 of
*      the License, or (at your option) any later version.
*
*      This program is distributed in the hope that it will be useful, but
*      WITHOUT ANY WARRANTY; without even the implied warranty of
*      MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
*      GNU Affero General Public License for more details.
*
*      You should have received a copy of the GNU Affero General Public
*      License along with this program. If not, see <http://www.gnu.org/licenses/>.
*/
?>
<script type="text/javascript">
$(document).ready(function(){
    $("#sub-button").click(function(){
        $.post('<?php echo osc_base_url(true); ?>', {email:$("#alert_email").val(), userid:$
            ($("#alert_userId").val(), alert:$("#alert").val(), page:"ajax", action:"alerts"},
            function(data){
                if(data==1) { alert('<?php echo osc_esc_js(--('You have sucessfully subscribed
                    to the alert', 'bender'))); ?>'); }
                else if(data==-1) { alert('<?php echo osc_esc_js(--('Invalid email address', '
                    bender'))); ?>'); }
                else { alert('<?php echo osc_esc_js(--('There was a problem with the alert', '
                    bender'))); ?>'); }
            });
        return false;
    });
});

var sQuery = '<?php echo osc_esc_js(AlertForm::default_email_text()); ?>';

if($('#input[name=alert_email]').val() == sQuery) {
    $('#input[name=alert_email]').css('color', 'gray');
}
$('#input[name=alert_email]').click(function(){
    if($('#input[name=alert_email]').val() == sQuery) {
        $('#input[name=alert_email]').val('');
        $('#input[name=alert_email]').css('color', '');
    }
});
$('#input[name=alert_email]').blur(function(){
    if($('#input[name=alert_email]').val() == '') {
        $('#input[name=alert_email]').val(sQuery);
        $('#input[name=alert_email]').css('color', 'gray');
    }
});
$('#input[name=alert_email]').keypress(function(){
    $('#input[name=alert_email]').css('background', '');
});
});
</script>

<div class="alert_form">
<h3>
<strong><?php _e('Subscribe to this search', 'bender'); ?></strong>
</h3>
<form action="<?php echo osc_base_url(true); ?>" method="post" name="sub_alert" id="
sub_alert" class="nocrsf">
    <?php AlertForm::page_hidden(); ?>
    <?php AlertForm::alert_hidden(); ?>

    <?php if(osc_is_web_user_logged_in()) { ?>
        <?php AlertForm::user_id_hidden(); ?>
        <?php AlertForm::email_hidden(); ?>

    <?php } else { ?>
        <?php AlertForm::user_id_hidden(); ?>
        <?php AlertForm::email_text(); ?>

    <?php }; ?>
    <button type="submit" class="sub-button" ><?php _e('Subscribe now', 'bender');
    ?>!</button>
</form>
</div>

<?php
/*
*      Oclass      software for creating and publishing online classified
*                  advertising platforms
*
*                  Copyright (C) 2013 OSCLASS
*
*      This program is free software: you can redistribute it and/or
*      modify it under the terms of the GNU Affero General Public License
*      as published by the Free Software Foundation, either version 3 of
*      the License, or (at your option) any later version.
*
*      This program is distributed in the hope that it will be useful, but
*      WITHOUT ANY WARRANTY; without even the implied warranty of
*      MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
*      GNU Affero General Public License for more details.
*
*      You should have received a copy of the GNU Affero General Public
*      License along with this program. If not, see <http://www.gnu.org/licenses/>.
*/

```

```

// meta tag robots
osc_add_hook('header', 'bender_nofollow_construct');

bender_add_body_class('contact');
osc_enqueue_script('jquery-validate');
osc_current_web_theme_path('header.php');
?>
<div class="form-container form-horizontal form-container-box">
  <div class="header">
    <h1><?php _e('Contact us', 'bender'); ?></h1>
  </div>
  <div class="resp-wrapper">
    <ul id="error_list"></ul>
    <form name="contact_form" action="<?php echo osc_base_url(true); ?>" method="post" >
      <input type="hidden" name="page" value="contact" />
      <input type="hidden" name="action" value="contact-post" />
      <div class="control-group">
        <label class="control-label" for="yourName">
          <?php _e('Your name', 'bender'); ?>
          (<?php _e('optional', 'bender'); ?>)</label>
        <div class="controls">
          <?php ContactForm::your_name(); ?></div>
        </div>
        <div class="control-group">
          <label class="control-label" for="yourEmail">
            <?php _e('Your email address', 'bender'); ?></label>
          <div class="controls">
            <?php ContactForm::your_email(); ?></div>
          </div>
        <div class="control-group">
          <label class="control-label" for="subject">
            <?php _e('Subject', 'bender'); ?>
            (<?php _e('optional', 'bender'); ?>)</label>
          <div class="controls">
            <?php ContactForm::the_subject(); ?></div>
          </div>
        <div class="control-group">
          <label class="control-label" for="message">
            <?php _e('Message', 'bender'); ?></label>
          <div class="controls textarea">
            <?php ContactForm::your_message(); ?></div>
          </div>
        <div class="control-group">
          <div class="controls">
            <?php osc_run_hook('contact_form'); ?>
            <?php osc_show_recaptcha(); ?>
            <button type="submit" class="ui-button ui-button-middle ui-button-main"><?php _e('Send', 'bender'); ?></button>
            <?php osc_run_hook('admin_contact_form'); ?>
          </div>
        </div>
      </form>
      <?php ContactForm::js_validation(); ?>
    </div>
  </div>
  <?php osc_current_web_theme_path('footer.php') ; ?>

<?php
/*
 *      Oclass      software for creating and publishing online classified
 *                  advertising platforms
 *
 *      Copyright (C) 2013 OSCLASS
 *
 *      This program is free software: you can redistribute it and/or
 *      modify it under the terms of the GNU Affero General Public License
 *      as published by the Free Software Foundation, either version 3 of
 *      the License, or (at your option) any later version.
 *
 *      This program is distributed in the hope that it will be useful, but
 *      WITHOUT ANY WARRANTY; without even the implied warranty of
 *      MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 *      GNU Affero General Public License for more details.
 *
 *      You should have received a copy of the GNU Affero General Public
 *      License along with this program. If not, see <http://www.gnu.org/licenses/>.
 */

bender_add_body_class('custom');
osc_current_web_theme_path('header.php') ;

?>
<?php osc_render_file(); ?>
<?php osc_current_web_theme_path('footer.php') ; ?>

<?php
/*
 *      Oclass      software for creating and publishing online classified
 *                  advertising platforms
 *
 *      Copyright (C) 2013 OSCLASS
 *
 *      This program is free software: you can redistribute it and/or
 *      modify it under the terms of the GNU Affero General Public License
 *      as published by the Free Software Foundation, either version 3 of
 *      the License, or (at your option) any later version.
 *
 *      This program is distributed in the hope that it will be useful, but

```

```

*          WITHOUT ANY WARRANTY; without even the implied warranty of
*          MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
*          GNU Affero General Public License for more details.
*
*          You should have received a copy of the GNU Affero General Public
*          License along with this program. If not, see <http://www.gnu.org/licenses/>.
*/
?>
</div><!-- content -->
<?php osc_run_hook('after-main'); ?>
</div>
<div id="responsive-trigger"></div>
<!-- footer -->
<div class="clear"></div>
<div id="footer">
  <div class="wrapper">
    <ul class="resp-toogle">
      <?php if( osc_users_enabled() ) { ?>
        <?php if( osc_is_web_user_logged_in() ) { ?>
          <li>
            <?php echo sprintf('--('Hi %s', 'bender'), osc_logged_user_name() . '!');
            ?> &middot;
            <strong><a href="<?php echo osc_user_dashboard_url(); ?>"><?php _e('My
            account', 'bender'); ?></a></strong> &middot;
            <a href="<?php echo osc_user_logout_url(); ?>"><?php _e('Logout', 'bender
            '); ?></a>
          </li>
          <?php } else { ?>
            <li><a href="<?php echo osc_user_login_url(); ?>"><?php _e('Login', 'bender');
            ?></a></li>
            <?php if(osc_user_registration_enabled()) { ?>
              <li>
                <a href="<?php echo osc_register_account_url(); ?>"><?php _e('Register
                for a free account', 'bender'); ?></a>
              </li>
            <?php } ?>
          <?php } ?>
          <?php } ?>
          <?php if( osc_users_enabled() || ( !osc_users_enabled() && !osc_reg_user_post() ) )
          { ?>
            <li class="publish">
              <a href="<?php echo osc_item_post_url_in_category(); ?>"><?php _e(" Publish
              your ad for free", 'bender');?></a>
            </li>
          <?php } ?>
        </ul>
        <ul>
          <?php
          osc_reset_static_pages();
          while( osc_has_static_pages() ) { ?>
            <li>
              <a href="<?php echo osc_static_page_url(); ?>"><?php echo
              osc_static_page_title(); ?></a>
            </li>
          <?php
          }
          ?>
            <li>
              <a href="<?php echo osc_contact_url(); ?>"><?php _e('Contact', 'bender'); ?></
              a>
            </li>
          </ul>
          <?php
          if( osc_get_preference('footer_link', 'bender_theme') !== '0') {
            echo '<div>Collab-Ads is a website featured with collaborative tagging and
            collaborative filtering. 2012</div>';
          }
          ?>
          <?php if ( osc_count_web_enabled_locales() > 1 ) { ?>
            <?php osc_goto_first_locale(); ?>
            <strong><?php _e('Language:', 'bender'); ?></strong>
            <?php $i = 0; ?>
            <?php while ( osc_has_web_enabled_locales() ) { ?>
              <span><a id="<?php echo osc_locale_code(); ?>" href="<?php echo
              osc_change_language_url ( osc_locale_code() ); ?>"><?php echo osc_locale_name
              (); ?></a></span><?php if( $i == 0 ) { echo " &middot; "; } ?>
              <?php $i++; ?>
            <?php } ?>
          <?php } ?>
        </div>
      </div>
      <?php osc_run_hook('footer'); ?>
    </body></html>

<?php
/*
*          Oclass      software for creating and publishing online classified
*          advertising platforms
*
*          Copyright (C) 2013 OSCLASS
*
*          This program is free software: you can redistribute it and/or
*          modify it under the terms of the GNU Affero General Public License
*          as published by the Free Software Foundation, either version 3 of
*          the License, or (at your option) any later version.
*
*          This program is distributed in the hope that it will be useful, but
*          WITHOUT ANY WARRANTY; without even the implied warranty of
*          MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
*/

```

```

*           GNU Affero General Public License for more details.
*
*           You should have received a copy of the GNU Affero General Public
* License along with this program. If not, see <http://www.gnu.org/licenses/>.
*/

/**
DEFINES

*/
define('BENDER_THEME_VERSION', '1.0');
if( !osc_get_preference('keyword_placeholder', 'bender_theme') ) {
    osc_set_preference('keyword_placeholder', '--(ie. PHP Programmer', 'bender'), '
        bender_theme');
}
osc_register_script('fancybox', osc_current_web_theme_url('js/fancybox/jquery.fancybox.
    pack.js'), array('jquery'));
osc_enqueue_style('fancybox', osc_current_web_theme_url('js/fancybox/jquery.fancybox.css')
);
osc_enqueue_script('fancybox');
// used for date/dateinterval custom fields
osc_enqueue_script('php-date');

/**

FUNCTIONS

*/

// install update options
if( !function_exists('benderBodyClass_theme_install') ) {
    function bender_theme_install() {
        osc_set_preference('keyword_placeholder', Params::getParam('keyword_placeholder'),
            'bender_theme');
        osc_set_preference('version', BENDER_THEME_VERSION, 'bender_theme');
        osc_set_preference('footer_link', '1', 'bender_theme');
        osc_set_preference('donation', '0', 'bender_theme');
        osc_set_preference('default_logo', '1', 'bender_theme');
        osc_set_preference('defaultShowAs@all', 'list', 'bender_theme');
        osc_set_preference('defaultShowAs@search', 'list');
        osc_reset_preferences();
    }
}

if(!function_exists('check_install_bender_theme')) {
    function check_install_bender_theme() {
        $current_version = osc_get_preference('version', 'bender_theme');
        //check if current version is installed or need an update<
        if( !$current_version ) {
            bender_theme_install();
        }
    }
}

if(!function_exists('bender_add_body_class_construct')) {
    function bender_add_body_class_construct($classes){
        $benderBodyClass = benderBodyClass::newInstance();
        $classes = array_merge($classes, $benderBodyClass->get());
        return $classes;
    }
}

if(!function_exists('bender_body_class')) {
    function bender_body_class($echo = true){
        /**
        * Print body classes.
        *
        * @param string $echo Optional parameter.
        * @return print string with all body classes concatenated
        */
        osc_add_filter('bender_bodyClass', 'bender_add_body_class_construct');
        $classes = osc_apply_filter('bender_bodyClass', array());
        if($echo && count($classes)){
            echo 'class="'.implode(' ', $classes).'";';
        } else {
            return $classes;
        }
    }
}

if(!function_exists('bender_add_body_class')) {
    function bender_add_body_class($class){
        /**
        * Add new body class to body class array.
        *
        * @param string $class required parameter.
        */
        $benderBodyClass = benderBodyClass::newInstance();
        $benderBodyClass->add($class);
    }
}

if(!function_exists('bender_nofollow_construct')) {
    /**
    * Hook for header, meta tags robots nofollos
    */
    function bender_nofollow_construct() {
        echo '<meta name="robots" content="noindex, nofollow, noarchive" />' . PHP_EOL;
        echo '<meta name="googlebot" content="noindex, nofollow, noarchive" />' . PHP_EOL;
    }
}

```

```

}
if( !function_exists('bender_follow_construct') ) {
    /**
     * Hook for header, meta tags robots follow
     */
    function bender_follow_construct() {
        echo '<meta name="robots" content="index, follow" />' . PHP_EOL;
        echo '<meta name="googlebot" content="index, follow" />' . PHP_EOL;
    }
}
/* logo */
if( !function_exists('logo_header') ) {
    function logo_header() {
        $html = '<a href="' . osc_base_url().'"></a>';
        if( file_exists( WebThemes::newInstance()->getCurrentThemePath() . 'images/logo.
            jpg' ) ) {
            return $html;
        } else {
            return '<a href="' . osc_base_url().'">' . osc_page_title(). '</a>';
        }
    }
}
/*
if( !function_exists('bender_draw_item') ) {
    function bender_draw_item($class = false, $admin = false, $premium = false) {
        $premiumSlug = '';
        if($premium){
            $premiumSlug = '-premium';
            View::newInstance()->exportVariableToView("item", View::newInstance()->
                _current('premiums'));
        } else {
            View::newInstance()->exportVariableToView("item", View::newInstance()->
                _current('latestItems'));
        }
        require WebThemes::newInstance()->getCurrentThemePath(). 'loop-single.php';
        View::newInstance()->_erase('item');
    }
}*/
if( !function_exists('bender_draw_item') ) {
    function bender_draw_item($class = false, $admin = false, $premium = false) {
        $filename = 'loop-single';
        if($premium){
            $filename .= '-premium';
        }
        require WebThemes::newInstance()->getCurrentThemePath(). $filename . '.php';
    }
}
if( !function_exists('bender_show_as') ){
    function bender_show_as(){
        $p_sShowAs = Params::getParam('sShowAs');
        $aValidShowAsValues = array('list', 'gallery');
        if (!in_array($p_sShowAs, $aValidShowAsValues)) {
            $p_sShowAs = bender_default_show_as();
        }
        return $p_sShowAs;
    }
}
if( !function_exists('bender_default_show_as') ){
    function bender_default_show_as(){
        return getPreference('defaultShowAs@all', 'bender_theme');
    }
}
if( !function_exists('bender_draw_categories_list') ) {
    function bender_draw_categories_list(){ ?>
    <?php if(!osc_is_home_page()){ echo '<div class="resp-wrapper">'; } ?>
    <?php
    //cell_3
    $total_categories = osc_count_categories();
    $coll_max_cat = ceil($total_categories/3);

    osc_goto_first_category();
    $i = 0;

    while ( osc_has_categories() ) {
    ?>
    <?php
    if($i%$coll_max_cat == 0){
        if($i > 0) { echo '</div>'; }
        if($i == 0) {
            echo '<div class="cell_3 first_cel">';
        } else {
            echo '<div class="cell_3">';
        }
    }
    ?>
    <ul class="r-list">
    <li>
        <h1><a class="category <?php echo osc_category_slug() ; ?>" href="<?php echo
            osc_search_category_url() ; ?>"><?php echo osc_category_name() ; ?></a> <
            span><?php echo osc_category_total_items() ; ?></span></h1>
        <?php /**/if ( osc_count_subcategories() > 0 ) { ?>
            <ul>
                <?php while ( osc_has_subcategories() ) { ?>
                    <li>
                        <?php if( osc_category_total_items() > 0 ) { ?><a class="category

```



```

        $iTotalItems = count($aItems);
        if( $iTotalItems > 0 ) {
            View::newInstance()->exportVariableToView('items', $aItems);
            return $iTotalItems;
        }
        unset($mSearch);

        return 0;
    }
}

if( !function_exists('osc_is_contact_page') ) {
    function osc_is_contact_page() {
        if( Rewrite::newInstance()->get_location() === 'contact' ) {
            return true;
        }

        return false;
    }
}

if( !function_exists('get_breadcrumb_lang') ) {
    function get_breadcrumb_lang() {
        $lang = array();
        $lang['item_add'] = --('Publish a listing', 'bender');
        $lang['item_edit'] = --('Edit your listing', 'bender');
        $lang['item_send_friend'] = --('Send to a friend', 'bender');
        $lang['item_contact'] = --('Contact publisher', 'bender');
        $lang['search'] = --('Search results', 'bender');
        $lang['search_pattern'] = --('Search results: %s', 'bender');
        $lang['user_dashboard'] = --('Dashboard', 'bender');
        $lang['user_dashboard_profile'] = --('%s's profile', 'bender');
        $lang['user_account'] = --('Account', 'bender');
        $lang['user_items'] = --('Listings', 'bender');
        $lang['user_alerts'] = --('Alerts', 'bender');
        $lang['user_profile'] = --('Update account', 'bender');
        $lang['user_change_email'] = --('Change email', 'bender');
        $lang['user_change_username'] = --('Change username', 'bender');
        $lang['user_change_password'] = --('Change password', 'bender');
        $lang['login'] = --('Login', 'bender');
        $lang['login_recover'] = --('Recover password', 'bender');
        $lang['login_forgot'] = --('Change password', 'bender');
        $lang['register'] = --('Create a new account', 'bender');
        $lang['contact'] = --('Contact', 'bender');
        return $lang;
    }
}

if( !function_exists('user_dashboard_redirect') ) {
    function user_dashboard_redirect() {
        $page = Params::getParam('page');
        $action = Params::getParam('action');
        if($page=='user' && $action=='dashboard') {
            if(ob_get_length()>0) {
                ob_end_flush();
            }
            header("Location: ".osc_user_list_items_url(), TRUE,301);
        }
    }
    osc_add_hook('init', 'user_dashboard_redirect');
}

if( !function_exists('get_user_menu') ) {
    function get_user_menu() {
        $options = array();
        $options[] = array(
            'name' => --('Listings', 'bender'),
            'url' => osc_user_list_items_url(),
            'class' => 'opt_items'
        );
        $options[] = array(
            'name' => --('Alerts', 'bender'),
            'url' => osc_user_alerts_url(),
            'class' => 'opt_alerts'
        );
        $options[] = array(
            'name' => --('Account', 'bender'),
            'url' => osc_user_profile_url(),
            'class' => 'opt_account'
        );
        $options[] = array(
            'name' => --('Change email', 'bender'),
            'url' => osc_change_user_email_url(),
            'class' => 'opt_change_email'
        );
        $options[] = array(
            'name' => --('Change username', 'bender'),
            'url' => osc_change_user_username_url(),
            'class' => 'opt_change_username'
        );
        $options[] = array(
            'name' => --('Change password', 'bender'),
            'url' => osc_change_user_password_url(),
            'class' => 'opt_change_password'
        );
        $options[] = array(
            'name' => --('Delete account', 'bender'),
            'url' => '#',
            'class' => 'opt_delete_account'
        );
    }
}

```





```

        break;
    }
}

function bender_redirect_user_dashboard()
{
    if( (Rewrite::newInstance()->get_location() === 'user') && (Rewrite::newInstance()->
        get_section() === 'dashboard') ) {
        header('Location: ' . osc_user_list_items_url());
        exit;
    }
}
osc_add_hook('init', 'bender_redirect_user_dashboard', 2);
osc_add_hook('init_admin', 'theme_bender_actions_admin');
osc_admin_menu_appearance('--('Header logo', 'bender'), osc_admin_render_theme_url('oc-
content/themes/bender/admin/header.php'), 'header_bender');
osc_admin_menu_appearance('--('Theme settings', 'bender'), osc_admin_render_theme_url('oc-
content/themes/bender/admin/settings.php'), 'settings_bender');

/**
TRIGGER FUNCTIONS

*/
check_install_bender_theme();
if(osc_is_home_page()){
    osc_add_hook('inside-main', 'bender_draw_categories_list');
} else if( osc_is_static_page() || osc_is_contact_page() ){
    osc_add_hook('before-content', 'bender_draw_categories_list');
}

if(osc_is_home_page() || osc_is_search_page()){
    bender_add_body_class('has-searchbox');
}

function bender_sidebar_category_search($catId = null)
{
    $aCategories = array();
    if($catId==null) {
        $aCategories[] = Category::newInstance()->findRootCategoriesEnabled();
    } else {
        // if parent category, only show parent categories
        $aCategories = Category::newInstance()->toRootTree($catId);
        end($aCategories);
        $cat = current($aCategories);
        // if is parent of some category
        $childCategories = Category::newInstance()->findSubcategories($cat['pk_i_id']);
        if(count($childCategories) > 0) {
            $aCategories[] = $childCategories;
        }
    }

    if(count($aCategories) == 0) {
        return "";
    }

    bender_print_sidebar_category_search($aCategories, $catId);
}

function bender_print_sidebar_category_search($aCategories, $current_category = null, $i = 0)
{
    $class = '';
    if(!isset($aCategories[$i])) {
        return null;
    }

    if($i==0) {
        $class = 'class="category"';
    }

    $c = $aCategories[$i];
    $i++;
    if(!isset($c['pk_i_id'])) {
        echo '<ul ' . $class . '>';
        if($i==1) {
            echo '<li><a href="' . osc_esc_html(osc_update_search_url(array('sCategory'=>null)))
                . '>' . _('--('All categories')) . '</a></li>';
        }
        foreach($c as $key => $value) {
            ?>
                <li>
                    <a id="cat_<?php echo osc_esc_html($value['pk_i_id']);?>" href="<?php echo
                        osc_esc_html(osc_update_search_url(array('sCategory'=> $value['pk_i_id']))
                            ); ?>">
                        <?php if(isset($current_category) && $current_category == $value['pk_i_id']){
                            echo '<strong>' . $value['s_name'] . '</strong>'; }
                        else{ echo $value['s_name']; } ?>
                    </a>
                </li>
            <?php
                }
                if($i==1) {
                    echo "</ul>";
                } else {
                    echo "</ul>";
                }
            }
        } else {
            ?>

```

```

<ul <?php echo $class;?>>
  <?php if($i==1) { ?>
  <li><a href="<?php echo osc_esc_html(osc_update_search_url(array('sCategory'=>null)));
    ?>"><?php _e('All categories '); ?></a></li>
  <?php } ?>
  <li>
    <a id="cat-<?php echo osc_esc_html($c['pk_i_id']);?>" href="<?php echo
      osc_esc_html(osc_update_search_url(array('sCategory'=> $c['pk_i_id']));
        ?>">
    <?php if(isset($current_category) && $current_category == $c['pk_i_id']){ echo
      '<strong>'. $c['s_name']. '</strong>'; }
      else{ echo $c['s_name']; } ?>
    </a>
    <?php bender_print_sidebar_category_search($aCategories, $current_category, $i
      ); ?>
  </li>
  <?php if($i==1) { ?>
  <?php } ?>
</ul>
<?php
}
}
/**
CLASSES
*/
class benderBodyClass
{
  /**
   * Custom Class for add, remove or get body classes.
   *
   * @param string $instance used for singleton.
   * @param array $class.
   */
  private static $instance;
  private $class;

  private function __construct()
  {
    $this->class = array();
  }

  public static function newInstance()
  {
    if ( !self::$instance instanceof self)
    {
      self::$instance = new self;
    }
    return self::$instance;
  }

  public function add($class)
  {
    $this->class[] = $class;
  }
  public function get()
  {
    return $this->class;
  }
}
?>

<?php
/**
 * Osclass software for creating and publishing online classified
 * advertising platforms
 *
 * Copyright (C) 2013 OSCLASS
 *
 * This program is free software: you can redistribute it and/or
 * modify it under the terms of the GNU Affero General Public License
 * as published by the Free Software Foundation, either version 3 of
 * the License, or (at your option) any later version.
 *
 * This program is distributed in the hope that it will be useful, but
 * WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 * GNU Affero General Public License for more details.
 *
 * You should have received a copy of the GNU Affero General Public
 * License along with this program. If not, see <http://www.gnu.org/licenses/>.
 */
?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/
DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" dir="ltr" lang="<?php echo str_replace('-', '- ',
osc_current_user_locale()); ?>">
<head>
  <?php osc_current_web_theme_path('common/head.php') ; ?>
</head>
<body <?php bender_body_class(); ?>>
<div id="header">
  <div class="wrapper">
    <div id="logo">
      <?php echo logo_header(); ?>
      <span id="description"><?php echo osc_page_description(); ?></span>
    </div>

```

```

<ul class="nav">
  <?php if( osc_is_static_page() || osc_is_contact_page() ){ ?>
    <li class="search"><a class="ico-search icons" data-bclass-toggle="display-search"></a></li>
    <li class="cat"><a class="ico-menu icons" data-bclass-toggle="display-cat"></a></li>
  <?php } ?>
  <?php if( osc_users_enabled() ) { ?>
  <?php if( osc_is_web_user_logged_in() ) { ?>
    <li class="first logged">
      <span><?php echo sprintf(__( 'Hi %s', 'bender' ), osc_logged_user_name() . '!' ); ?> &middot;</span>
      <strong><a href="<?php echo osc_user_dashboard_url(); ?>"><?php _e( 'My account', 'bender' ); ?></a></strong> &middot;
      <a href="<?php echo osc_user_logout_url(); ?>"><?php _e( 'Logout', 'bender' ); ?></a>
    </li>
  <?php } else { ?>
    <li><a id="login_open" href="<?php echo osc_user_login_url(); ?>"><?php _e( 'Login', 'bender' ); ?></a></li>
    <?php if( osc_user_registration_enabled() ) { ?>
      <li><a href="<?php echo osc_register_account_url(); ?>"><?php _e( 'Register for a free account', 'bender' ); ?></a></li>
    <?php }; ?>
  <?php } ?>
  <?php } ?>
  <?php if( osc_users_enabled() || ( !osc_users_enabled() && !osc_reg_user_post() ) ) { ?>
    <?php if( osc_is_web_user_logged_in() ) { ?>
    <li class="publish"><a href="<?php echo osc_item_post_url_in_category(); ?>"><?php _e( "Post your ad", 'bender' ); ?></a></li>
      <?php } ?>
      <?php if( !osc_is_web_user_logged_in() ) { ?>
        <li class="publish"><a href="<?php echo osc_user_login_url(); ?>"><?php _e( 'Post your ad', 'hierarchy' ); ?></a></li>
        <?php } ?>
      <?php } ?>
    </ul>
  </div>
  <?php if( osc_is_home_page() || osc_is_static_page() || osc_is_contact_page() ) { ?>
  <form action="<?php echo osc_base_url( true ); ?>" method="get" class="search nocsrf" <?php
  /* onsubmit="javascript: return doSearch();"*/ ?>>
  <input type="hidden" name="page" value="search"/>
  <div class="main-search">
    <div class="cell">
      <input type="text" name="sPattern" id="query" class="input-text" value=""
      placeholder="<?php echo osc_esc_html( __( osc_get_preference( 'keyword_placeholder', 'bender-theme' ), 'bender' ); ?> ) />
    </div>
    <?php if ( osc_count_categories() ) { ?>
      <div class="cell selector">
        <?php osc_categories_select( 'sCategory', null, __( 'Select a category', 'bender' ); ?>
        </div>
      <div class="cell reset-padding">
        <?php } else { ?>
        <div class="cell">
          <?php } ?>
          <button class="ui-button ui-button-big js-submit"><?php _e( "Search", 'bender' ); ?></button>
        </div>
      </div>
    </div>
  <div id="message-search"></div>
  </form>
  <?php } ?>
</div>
<div class="wrapper wrapper-flash">
  <?php
  $breadcrumb = osc_breadcrumb( '&raquo;', false, get_breadcrumb_lang() );
  if( $breadcrumb != '' ) { ?>
  <div class="breadcrumb">
    <?php echo $breadcrumb; ?>
  <div class="clear"></div>
  </div>
  <?php
  }
  ?>
  <?php osc_show_flash_message(); ?>
</div>
<?php osc_run_hook( 'before-content' ); ?>
<div class="wrapper" id="content">
  <?php osc_run_hook( 'before-main' ); ?>
  <div id="main">
    <?php osc_run_hook( 'inside-main' ); ?>
  </div>
</div>
<?php
/*
 *      Oclass      software for creating and publishing online classified
 *                  advertising platforms
 *
 *
 *      Copyright (C) 2013 OSCLASS
 *
 *      This program is free software: you can redistribute it and/or
 *      modify it under the terms of the GNU Affero General Public License
 *      as published by the Free Software Foundation, either version 3 of
 *      the License, or (at your option) any later version.
 *
 *      This program is distributed in the hope that it will be useful, but
 *      WITHOUT ANY WARRANTY; without even the implied warranty of
 */

```

```

*          MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
*          GNU Affero General Public License for more details.
*
* You should have received a copy of the GNU Affero General Public
* License along with this program. If not, see <http://www.gnu.org/licenses/>.
*/

/*
Theme Name: bender
Theme URI: http://www.osclass.org/
Description: bender
Version: 1.0
Author: Osclass team
Author URI: http://www.osclass.org/
Widgets:
Theme update URI:
*/

function bender_theme_info() {
    return array(
        'name' => 'bender',
        'version' => '1.0',
        'description' => 'bender',
        'author_name' => 'Osclass Team',
        'author_url' => 'http://osclass.org',
        'locations' => array()
    );
}

?>

<?php
/*
*          Osclass software for creating and publishing online classified
*          advertising platforms
*
*          Copyright (C) 2013 OSCLASS
*
*          This program is free software: you can redistribute it and/or
*          modify it under the terms of the GNU Affero General Public License
*          as published by the Free Software Foundation, either version 3 of
*          the License, or (at your option) any later version.
*
*          This program is distributed in the hope that it will be useful, but
*          WITHOUT ANY WARRANTY; without even the implied warranty of
*          MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
*          GNU Affero General Public License for more details.
*
*          You should have received a copy of the GNU Affero General Public
*          License along with this program. If not, see <http://www.gnu.org/licenses/>.
*/

// meta tag robots
osc_add_hook('header', 'bender_follow_construct');

osc_enqueue_script('fancybox');
osc_enqueue_style('fancybox', osc_current_web_theme_url('js/fancybox/jquery.fancybox.css'));
osc_enqueue_script('jquery-validate');

bender_add_body_class('item');
osc_add_hook('after-main', 'sidebar');
function sidebar(){
    osc_current_web_theme_path('item-sidebar.php');
}

$location = array();
if( osc_item_city_area() !== '' ) {
    $location[] = osc_item_city_area();
}
if( osc_item_city() !== '' ) {
    $location[] = osc_item_city();
}
if( osc_item_region() !== '' ) {
    $location[] = osc_item_region();
}
if( osc_item_country() !== '' ) {
    $location[] = osc_item_country();
}

osc_current_web_theme_path('header.php');

?>
<div id="item-content">
<h1><?php if( osc_price_enabled_at_items() ) { ?><span class="price"><?php echo
    osc_item_formatted_price(); ?></span> <?php } ?><strong><?php echo osc_item_title()
    . ' ' . osc_item_city(); ?></strong></h1>
<div class="item-header">
<div>
<?php if ( osc_item_pub_date() !== '' ) { printf( __( '<strong class="publish">
    Published date</strong>: %1$s', 'bender' ), osc_format_date(
    osc_item_pub_date() ) ); } ?>
</div>
<div>
<?php if ( osc_item_mod_date() !== '' ) { printf( __( '<strong class="update">
    Modified date:</strong> %1$s', 'bender' ), osc_format_date(
    osc_item_mod_date() ) ); } ?>
</div>
<ul id="item.location">
<li><strong><?php _e("Location", 'bender'); ?></strong>: <?php echo implode('
    ', $location); ?></li>

```

```

        </ul>
    </div>
    <?php if( osc_is_web_user_logged_in() && osc_logged_user_id()==osc_item_user_id() ) { ?>
        <p id="edit_item_view">
            <strong>
                <a href="<?php echo osc_item_edit_url(); ?>" rel="nofollow"><?php _e('Edit
                    item', 'bender'); ?></a>
            </strong>
        </p>
    <?php } ?>

    <?php if( osc_images_enabled_at_items() ) { ?>
        <?php
            if( osc_count_item_resources() > 0 ) {
                $i = 0;
            ?>
            <div class="item-photos">
                <a href="<?php echo osc_resource_url(); ?>" class="main-photo" title="<?php _e('
                    Image', 'bender'); ?> <?php echo $i+1;?> / <?php echo osc_count_item_resources
                    ();?>">
                    " title="<?php echo osc_item_title(); ?>" />
                </a>
                <div class="thumbs">
                    <?php for ( $i = 0; osc_has_item_resources(); $i++ ) { ?>
                        <a href="<?php echo osc_resource_url(); ?>" rel="image_group" title="<?php _e
                            ('Image', 'bender'); ?> <?php echo $i+1;?> / <?php echo
                                osc_count_item_resources ();?>">
                            " title="<?php echo osc_item_title(); ?>"
                                />
                        </a>
                    <?php } ?>
                </div>
            </div>
        <?php } ?>
    </div>
    <?php } ?>
    <?php } ?>
    <div id="description">
        <p><?php echo osc_item_description(); ?></p>
        <div id="custom_fields">
            <?php if( osc_count_item_meta() >= 1 ) { ?>
                <br />
                <div class="meta_list">
                    <?php while ( osc_has_item_meta() ) { ?>
                        <?php if( osc_item_meta_value() != '' ) { ?>
                            <div class="meta">
                                <strong><?php echo osc_item_meta_name(); ?></strong> <?php
                                    echo osc_item_meta_value(); ?>
                            </div>
                        <?php } ?>
                    <?php } ?>
                </div>
            <?php } ?>
        </div>
        <?php osc_run_hook('item_detail', osc_item()); ?>
        <p class="contact_button">
            <?php if( !osc_item_is_expired() ) { ?>
                <?php if( !( ( osc_logged_user_id() == osc_item_user_id() ) && osc_logged_user_id
                    () != 0 ) ) { ?>
                    <?php
                        if( osc_reg_user_can_contact() && osc_is_web_user_logged_in() || !
                            osc_reg_user_can_contact() ) { ?>
                            <a href="#contact" class="ui-button ui-button-middle ui-button-main resp-
                                toogle"><?php _e('Contact seller', 'bender'); ?></a>
                        <?php
                            } ?>
                    <?php
                        } ?>
                <?php } ?>
                <a href="<?php echo osc_item_send_friend_url(); ?>" rel="nofollow" class="ui-button
                    ui-button-middle"><?php _e('Share', 'bender'); ?></a>
            </p>
            <?php osc_run_hook('location'); ?>
        </div>
        <!-- plugins -->
        <div id="useful_info" class="bordered-box">
            <h2><?php _e('Useful information', 'bender'); ?></h2>
            <ul>
                <li><?php _e('Avoid scams by acting locally or paying with PayPal', 'bender');
                    ?></li>
                <li><?php _e('Never pay with Western Union, Moneygram or other anonymous payment
                    services', 'bender'); ?></li>
                <li><?php _e('Don\'t buy or sell outside of your country. Don\'t accept cashier
                    cheques from outside your country', 'bender'); ?></li>
                <li><?php _e('This site is never involved in any transaction, and does not handle
                    payments, shipping, guarantee transactions, provide escrow services, or offer
                    "buyer protection" or "seller certification"', 'bender'); ?></li>
            </ul>
        </div>

        <?php
            if( function_exists('echo_recommended_ads') ) {
                echo_recommended_ads();
            }
        ?>

        <?php related_listings(); ?>
        <?php if( osc_count_items() > 0 ) { ?>
            <div class="similar_ads">
                <h2><?php _e('Related listings', 'bender'); ?></h2>

```

```

<?php
View::newInstance()->.exportVariableToView("listType", 'items');
osc_current_web_theme_path('loop.php');
?>
<div class="clear"></div>
</div>
<?php } ?>
<?php if( osc_comments_enabled() ) { ?>
<?php if( osc_reg_user_post_comments() && osc_is_web_user_logged_in() || !
osc_reg_user_post_comments() ) { ?>
<div id="comments">
<h2><?php _e('Comments', 'bender'); ?></h2>
<?php if( osc_count_item_comments() == 0 ) { _e('No comments', 'bender'); } ?>
<ul id="comment_error_list"></ul>
<?php CommentForm::js_validation(); ?>
<?php if( osc_count_item_comments() >= 1 ) { ?>
<div class="comments-list">
<?php while ( osc_has_item_comments() ) { ?>
<div class="comment">
<h3><strong><?php echo osc_comment_title(); ?></strong> <em><?php
_e("by", 'bender'); ?> <?php echo osc_comment_author_name();
?> <?php _e("on", 'bender'); ?> &nbsp; <?php
Timestamp = new DateTime(
osc_comment_pub_date());
echo date_format(Timestamp,
F d, Y h:i A');
?>
<?php _e(":", 'bender'); ?></em></h3>
<p><?php echo nl2br( osc_comment_body() ); ?> </p>
<?php if ( osc_comment_user_id() && (osc_comment_user_id() ==
osc_logged_user_id()) ) { ?>
<p>
<a rel="nofollow" href="<?php echo osc_delete_comment_url();
?" title="<?php _e('Delete your comment', 'bender');
?"><?php _e('Delete', 'bender'); ?></a>
</p>
<?php } ?>
</div>
<?php } ?>
<div class="paginate" style="text-align: right;">
<?php echo osc_comments_pagination(); ?>
</div>
</div>
<?php } ?>
<?php if(osc_is_web_user_logged_in()) { ?>
<div class="form-container form-horizontal">
<div class="header">
<h3><?php _e('Leave your comment (spam and offensive messages will be
removed)', 'bender'); ?></h3>
</div>
<div class="resp-wrapper">
<form action="<?php echo osc_base_url(true); ?>" method="post" name="
comment_form" id="comment_form">
<fieldset>
<input type="hidden" name="action" value="add_comment" />
<input type="hidden" name="page" value="item" />
<input type="hidden" name="id" value="<?php echo osc_item_id();
?" />
<?php if(osc_is_web_user_logged_in()) { ?>
<input type="hidden" name="authorName" value="<?php echo
osc_esc_html( osc_logged_user_name() ); ?>" />
<input type="hidden" name="authorEmail" value="<?php echo
osc_logged_user_email(); ?>" />
<?php } else { ?>
<div class="control-group">
<label class="control-label" for="authorName"><?php _e('
Your name', 'bender'); ?></label>
<div class="controls">
<?php CommentForm::author_input_text(); ?>
</div>
</div>
<div class="control-group">
<label class="control-label" for="authorEmail"><?php _e('
Your e-mail', 'bender'); ?></label>
<div class="controls">
<?php CommentForm::email_input_text(); ?>
</div>
</div>
<?php } ?>
<div class="control-group">
<label class="control-label" for="title"><?php _e('Title', '
bender'); ?></label>
<div class="controls">
<?php CommentForm::title_input_text(); ?>
</div>
</div>
<div class="control-group">
<label class="control-label" for="body"><?php _e('Comment', '
bender'); ?></label>
<div class="controls textarea">
<?php CommentForm::body_input_textarea(); ?>
</div>
</div>
<div class="actions">
<button type="submit"><?php _e('Send', 'bender'); ?></button>
</div>

```

```

        </fieldset>
    </form>
</div>
</div>
        <?php } else { ?>
        <a href="<?php echo osc_user_login_url(); ?>"><?php _e('Login
        to post your comments', 'hierarchy'); ?></a>
        <?php } ?>
    </div>
    <?php } ?>
</div>
<?php osc_current_web_theme_path('footer.php') ; ?>

<?php
/*
 *      Oclass      software for creating and publishing online classified
 *                  advertising platforms
 *
 *                  Copyright (C) 2013 OSCLASS
 *
 *      This program is free software: you can redistribute it and/or
 *      modify it under the terms of the GNU Affero General Public License
 *      as published by the Free Software Foundation, either version 3 of
 *      the License, or (at your option) any later version.
 *
 *      This program is distributed in the hope that it will be useful, but
 *      WITHOUT ANY WARRANTY; without even the implied warranty of
 *      MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 *      GNU Affero General Public License for more details.
 *
 *      You should have received a copy of the GNU Affero General Public
 *      License along with this program. If not, see <http://www.gnu.org/licenses/>.
 */

// meta tag robots
osc_add_hook('header', 'bender_nofollow_construct');

osc_current_web_theme_path('item-post.php');
?>

<?php
/*
 *      Oclass      software for creating and publishing online classified
 *                  advertising platforms
 *
 *                  Copyright (C) 2013 OSCLASS
 *
 *      This program is free software: you can redistribute it and/or
 *      modify it under the terms of the GNU Affero General Public License
 *      as published by the Free Software Foundation, either version 3 of
 *      the License, or (at your option) any later version.
 *
 *      This program is distributed in the hope that it will be useful, but
 *      WITHOUT ANY WARRANTY; without even the implied warranty of
 *      MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 *      GNU Affero General Public License for more details.
 *
 *      You should have received a copy of the GNU Affero General Public
 *      License along with this program. If not, see <http://www.gnu.org/licenses/>.
 */

// meta tag robots
osc_add_hook('header', 'bender_nofollow_construct');

osc_enqueue_script('jquery-validate');
bender_add_body_class('item item-post');
$action = 'item_add_post';
$edit = false;
if(Params::getParam('action') == 'item-edit'){
    $action = 'item_edit_post';
    $edit = true;
}
?>
<?php osc_current_web_theme_path('header.php') ; ?>
<?php ItemForm::location_javascript_new(); ?>
<?php if(osc_images_enabled_at_items()) ItemForm::photos_javascript(); ?>
<div class="form-container form-horizontal">
    <div class="resp-wrapper">
        <div class="header">
            <h1><?php _e('Publish a listing', 'bender'); ?></h1>
        </div>
        <ul id="error_list"></ul>
        <form name="item" action="<?php echo osc_base_url(true);?>" method="post"
            enctype="multipart/form-data" id="item-post">
            <fieldset>
                <input type="hidden" name="action" value="<?php echo $action; ?>" />
                <input type="hidden" name="page" value="item" />
                <?php if($edit){ ?>
                    <input type="hidden" name="id" value="<?php echo osc_item_id();?>" />
                    <input type="hidden" name="secret" value="<?php echo osc_item_secret()
                    ;?>" />
                <?php } ?>
                <h2><?php _e('General Information', 'bender'); ?></h2>
            <div class="control-group">

```

```

        <label class="control-label" for="select_1"><?php _e('Category', '
            bender'); ?></label>
        <div class="controls">
            <?php ItemForm::category_select(null, null, _('Select a
                category', 'bender')); ?>
        </div>
    </div>
    <div class="control-group">
        <label class="control-label" for="title[<?php echo osc_locale_code
            (); ?>]"><?php _e('Title', 'bender'); ?></label>
        <div class="controls">
            <?php ItemForm::title_input('title', osc_locale_code(),
                osc_esc_html( bender_item_title() )); ?>
        </div>
    </div>
    <div class="control-group">
        <label class="control-label" for="description[<?php echo
            osc_locale_code(); ?>]"><?php _e('Description', 'bender');
            ?></label>
        <div class="controls">
            <?php ItemForm::description_textarea('description',
                osc_locale_code(), osc_esc_html( bender_item_description()
                    )); ?>
        </div>
    </div>
    <?php if( osc_price_enabled_at_items() ) { ?>
    <div class="control-group">
        <label class="control-label" for="price"><?php _e('Price', 'bender
            '); ?></label>
        <div class="controls">
            <?php ItemForm::price_input_text(); ?>
            <?php ItemForm::currency_select(); ?>
        </div>
    </div>
    <?php } ?>
    <?php if( osc_images_enabled_at_items() ) { ?>
    <div class="box_photos">
        <h2><?php _e('Photos', 'bender'); ?></h2>
        <div class="control-group">
            <label class="control-label" for="photos[]"><?php _e('Photos',
                'bender'); ?></label>
            <div class="controls">
                <div id="photos">
                    <?php ItemForm::photos(); ?>
                </div>
            </div>
            <div class="controls">
                <a href="#" onclick="addNewPhoto(); return false;"><?php
                    _e('Add new photo', 'bender'); ?></a>
            </div>
        </div>
    </div>
    <?php } ?>
    <div class="box_location">
        <h2><?php _e('Listing Location', 'bender'); ?></h2>
        <div class="control-group">
            <label class="control-label" for="country"><?php _e('Country',
                'bender'); ?></label>
            <div class="controls">
                <?php ItemForm::country_select(osc_get_countries(),
                    osc_user()); ?>
            </div>
        </div>
        <div class="control-group">
            <label class="control-label" for="region"><?php _e('Region',
                'bender'); ?></label>
            <div class="controls">
                <?php ItemForm::region_text(osc_user()); ?>
            </div>
        </div>
        <div class="control-group">
            <label class="control-label" for="city"><?php _e('City',
                'bender'); ?></label>
            <div class="controls">
                <?php ItemForm::city_text(osc_user()); ?>
            </div>
        </div>
        <div class="control-group">
            <label class="control-label" for="cityArea"><?php _e('City
                Area', 'bender'); ?></label>
            <div class="controls">
                <?php ItemForm::city_area_text(osc_user()); ?>
            </div>
        </div>
        <div class="control-group">
            <label class="control-label" for="address"><?php _e('Address',
                'bender'); ?></label>
            <div class="controls">
                <?php ItemForm::address_text(osc_user()); ?>
            </div>
        </div>
    </div>
    <!-- seller info -->
    <?php if(!osc_is_web_user_logged_in() ) { ?>
    <div class="box_seller_info">
        <h2><?php _e("Seller's information", 'bender'); ?></h2>
        <div class="control-group">
            <label class="control-label" for="contactName"><?php _e('Name

```



```

        , 'bender'); ?></label>
        <div class="controls">
            <?php ItemForm::contact_name_text(); ?>
        </div>
    </div>
    <div class="control-group">
        <label class="control-label" for="contactEmail"><?php _e('E-
        mail', 'bender'); ?></label>
        <div class="controls">
            <?php ItemForm::contact_email_text(); ?>
        </div>
    </div>
    <div class="control-group">
        <div class="controls checkbox">
            <?php ItemForm::show_email_checkbox(); ?> <label for="
            showEmail"><?php _e('Show e-mail on the listing page',
            'bender'); ?></label>
        </div>
    </div>
</div>
<?php
}
if($edit) {
    ItemForm::plugin_edit_item();
} else {
    ItemForm::plugin_post_item();
}
?>
<div class="control-group">
    <?php if( osc_recaptcha_items_enabled() ) { ?>
        <div class="controls">
            <?php osc_show_recaptcha(); ?>
        </div>
    <?php ?>
    <div class="controls">
        <button type="submit" class="ui-button ui-button-middle ui-
        button-main"><?php if($edit) { _e("Update", 'bender'); }
        else { _e("Publish", 'bender'); } ?></button>
    </div>
</div>
</div>
</form>
</div>
<script type="text/javascript">
<?php if(osc_locale_thousands_sep()!='' || osc_locale_dec_point() != '') { ?>
$(function() {
    $("#price").blur(function(event) {
        var price = $("#price").prop("value");
        <?php if(osc_locale_thousands_sep()!='') { ?>
            while(price.indexOf('<?php echo osc_esc_js(osc_locale_thousands_sep()); ?>')!=-1)
            {
                price = price.replace('<?php echo osc_esc_js(osc_locale_thousands_sep());
                ?>', '');
            }
            <?php ?>
            <?php if(osc_locale_dec_point()!='') { ?>
                var tmp = price.split('<?php echo osc_esc_js(osc_locale_dec_point())?>');
                if(tmp.length>2) {
                    price = tmp[0]+'<?php echo osc_esc_js(osc_locale_dec_point())?>'+tmp[1];
                }
            <?php ?>
            $("#price").prop("value", price);
        });
    });
</script>
<?php osc_current_web_theme_path('footer.php'); ?>

<?php
/*
 *      Oclass      software for creating and publishing online classified
 *                  advertising platforms
 *
 *
 *                  Copyright (C) 2013 OSCLASS
 *
 *
 *      This program is free software: you can redistribute it and/or
 *      modify it under the terms of the GNU Affero General Public License
 *      as published by the Free Software Foundation, either version 3 of
 *      the License, or (at your option) any later version.
 *
 *
 *      This program is distributed in the hope that it will be useful, but
 *      WITHOUT ANY WARRANTY; without even the implied warranty of
 *      MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 *      GNU Affero General Public License for more details.
 *
 *
 *      You should have received a copy of the GNU Affero General Public
 *      License along with this program. If not, see <http://www.gnu.org/licenses/>.
 */

// meta tag robots
osc_add_hook('header', 'bender_nofollow_construct');

osc_enqueue_script('jquery-validate');
bender_add_body_class('contact');
osc_current_web_theme_path('header.php');
?>
<div class="form-container form-horizontal form-container-box">
    <div class="header">

```

```

        <h1><?php _e('Send to a friend', 'bender'); ?></h1>
    </div>
    <div class="resp-wrapper">
        <ul id="error_list"></ul>
        <form name="sendfriend" action="<?php echo osc_base_url(true); ?>" method="post" >
            <input type="hidden" name="action" value="send_friend_post" />
            <input type="hidden" name="page" value="item" />
            <input type="hidden" name="id" value="<?php echo osc_item_id(); ?>" />
            <?php if(osc_is_web_user_logged_in()) { ?>
                <input type="hidden" name="yourName" value="<?php echo
                    osc_esc_html( osc_logged_user_name() ); ?>" />
                <input type="hidden" name="yourEmail" value="<?php echo
                    osc_logged_user_email();?>" />
            <?php } else { ?>
                <div class="control-group">
                    <label class="control-label" for="yourName"><?php _e("Your name", 'bender');
                        ?></label>
                    <div class="controls">
                        <?php SendFriendForm::your_name(); ?>
                    </div>
                </div>
                <div class="control-group">
                    <label class="control-label" for="yourEmail"><?php _e("Your e-mail", 'bender');
                        ?></label>
                    <div class="controls">
                        <?php SendFriendForm::your_email(); ?>
                    </div>
                </div>
                <?php } ?>
                <div class="control-group">
                    <label class="control-label" for="friendName"><?php _e("Your friend's name",
                        'bender'); ?></label>
                    <div class="controls">
                        <?php SendFriendForm::friend_name(); ?>
                    </div>
                </div>
                <div class="control-group">
                    <label for="friendEmail"><?php _e("Your friend's e-mail address", 'bender');
                        ?></label> </label>
                    <div class="controls">
                        <?php SendFriendForm::friend_email(); ?>
                    </div>
                </div>
                <div class="control-group">
                    <label class="control-label" for="subject">
                        <?php _e('Subject (optional)', 'bender'); ?>
                    </label>
                    <div class="controls">
                        <?php ContactForm::the_subject(); ?>
                    </div>
                </div>
                <div class="control-group">
                    <label class="control-label" for="message">
                        <?php _e('Message', 'bender'); ?></label>
                    <div class="controls textarea">
                        <?php SendFriendForm::your_message(); ?>
                    </div>
                </div>
                <div class="control-group">
                    <div class="controls">
                        <?php osc_run_hook('contact_form'); ?>
                        <?php osc_show_recaptcha(); ?>
                        <button type="submit" class="ui-button ui-button-middle ui-button-main"><?
                            php _e("Send", 'bender'); ?></button>
                        <?php osc_run_hook('admin.contact_form'); ?>
                    </div>
                </div>
            </form>
            <?php SendFriendForm::js_validation(); ?>
        </div>
    </div>
    <?php osc_current_web_theme_path('footer.php'); ?>

<?php
/*
 *      Oclass      software for creating and publishing online classified
 *                  advertising platforms
 *
 *                  Copyright (C) 2013 OSCLASS
 *
 *      This program is free software: you can redistribute it and/or
 *      modify it under the terms of the GNU Affero General Public License
 *      as published by the Free Software Foundation, either version 3 of
 *      the License, or (at your option) any later version.
 *
 *      This program is distributed in the hope that it will be useful, but
 *      WITHOUT ANY WARRANTY; without even the implied warranty of
 *      MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 *      GNU Affero General Public License for more details.
 *
 *      You should have received a copy of the GNU Affero General Public
 *      License along with this program. If not, see <http://www.gnu.org/licenses/>.
 */
?>
<div id="sidebar">
    <?php //if(!osc_is_web_user_logged_in() || osc_logged_user_id()!=osc_item_user_id()) { ?>
        <?php if( osc_is_web_user_logged_in() && osc_logged_user_id()!=osc_item_user_id()) {
            ?>
        }
    }

```

```

<form action="<?php echo osc_base_url(true); ?>" method="post" name="mask_as_form" id
="mask_as_form">
  <input type="hidden" name="id" value="<?php echo osc_item_id(); ?>" />
  <input type="hidden" name="as" value="spam" />
  <input type="hidden" name="action" value="mark" />
  <input type="hidden" name="page" value="item" />
  <select name="as" id="as" class="mark_as">
    <option><?php _e("Mark as...", 'bender'); ?></option>
    <option value="spam"><?php _e("Mark as spam", 'bender'); ?></option>
    <option value="badcat"><?php _e("Mark as misclassified", 'bender'); ?></
    option>
    <option value="repeated"><?php _e("Mark as duplicated", 'bender'); ?></
    option>
    <option value="expired"><?php _e("Mark as expired", 'bender'); ?></option>
    <option value="offensive"><?php _e("Mark as offensive", 'bender'); ?></
    option>
  </select>
</form>
<?php } ?>
<div id="contact" class="widget-box form-container form-vertical">
  <h2><?php _e("Contact publisher", 'bender'); ?></h2>
  <?php if( osc_item_is_expired() ) { ?>
    <p>
      <?php _e("The listing is expired. You can't contact the publisher.", 'bender')
      ; ?>
    </p>
  <?php } else if( ( osc_logged_user_id() == osc_item_user_id() ) && osc_logged_user_id
  () != 0 ) { ?>
    <p>
      <?php _e("It's your own listing, you can't contact the publisher.", 'bender');
      ?>
    </p>
  <?php //} else if( osc_reg_user_can_contact() && !osc_is_web_user_logged_in() ) { ?>
    <?php //} else if( !osc_is_web_user_logged_in() ) { ?>
    <!--<p>
      <?php //_e("You must log in or register a new account to contact the
      advertiser", 'bender'); ?>
    </p>
    <p class="contact-button">
      <strong><a href="<?php //echo osc_user_login_url(); ?>"><?php //_e('Login', '
      bender'); ?></a></strong>
      <strong><a href="<?php //echo osc_register_account_url(); ?>"><?php //_e('
      Register for a free account', 'bender'); ?></a></strong>
    </p-->
  <?php } else { ?>
    <?php if( osc_item_user_id() != null ) { ?>
      <p class="name"><?php _e('Name', 'bender') ?>: <a href="<?php echo
      osc_user_public_profile_url( osc_item_user_id() ); ?>"><?php echo
      osc_item_contact_name(); ?></a></p>
    <?php } else { ?>
      <p class="name"><?php printf(__( 'Name: %s', 'bender' ), osc_item_contact_name(
      ) ); ?></p>
    <?php } ?>
    <?php if( osc_item_show_email() ) { ?>
      <p class="email"><?php printf(__( 'E-mail: %s', 'bender' ),
      osc_item_contact_email() ); ?></p>
    <?php } ?>
    <?php if ( osc_user_phone() != '' ) { ?>
      <p class="phone"><?php printf(__( "Phone: %s", 'bender' ), osc_user_phone() );
      ?></p>
    <?php } ?>
    <ul id="error_list"></ul>
    <form action="<?php echo osc_base_url(true); ?>" method="post" name="contact_form"
    id="contact_form" <?php if(osc_item_attachment()) { echo 'enctype="multipart/
    form-data" '; }?> >
      <?php osc_prepare_user_info(); ?>
      <input type="hidden" name="action" value="contact_post" />
      <input type="hidden" name="page" value="item" />
      <input type="hidden" name="id" value="<?php echo osc_item_id(); ?>" />
      <div class="control-group">
        <label class="control-label" for="yourName"><?php _e('Your name', 'bender
        '); ?></label>
        <div class="controls"><?php ContactForm::your_name(); ?></div>
      </div>
      <div class="control-group">
        <label class="control-label" for="yourEmail"><?php _e('Your e-mail address
        ', 'bender'); ?></label>
        <div class="controls"><?php ContactForm::your_email(); ?></div>
      </div>
      <div class="control-group">
        <label class="control-label" for="phoneNumber"><?php _e('Phone number', '
        bender'); ?> (<?php _e('optional', 'bender'); ?>)</label>
        <div class="controls"><?php ContactForm::your_phone_number(); ?></div>
      </div>
      <div class="control-group">
        <label class="control-label" for="message"><?php _e('Message', 'bender')
        ?></label>
        <div class="controls textarea"><?php ContactForm::your_message(); ?></div>
      </div>
      <?php if(osc_item_attachment()) { ?>
        <div class="control-group">
          <label class="control-label" for="attachment"><?php _e('Attachment', '
          bender'); ?></label>
          <div class="controls"><?php ContactForm::your_attachment(); ?></div>
        </div>
      <?php } ?>
  </div>
</div>

```

```

<div class="control-group">
  <div class="controls">
    <?php osc_run_hook('item_contact_form', osc_item_id()); ?>
    <?php if( osc_recaptcha_public_key() ) { ?>
    <script type="text/javascript">
      var RecaptchaOptions = {
        theme : 'custom',
        custom_theme_widget: 'recaptcha_widget'
      };
    </script>
    <style type="text/css"> div#recaptcha_widget, div#recaptcha_image >
      img { width:280px; } </style>
    <div id="recaptcha_widget">
      <div id="recaptcha_image"><img /></div>
      <span class="recaptcha_only_if_image"><?php _e('Enter the words
        above', 'bender'); ?></span>
      <input type="text" id="recaptcha_response_field" name="
        recaptcha_response_field" />
      <div><a href="javascript:Recaptcha.showhelp()"><?php _e('Help', '
        bender'); ?></a></div>
    </div>
    <?php } ?>
    <?php osc_show_recaptcha(); ?>
    <button type="submit" class="ui-button ui-button-middle ui-button-main
      "><?php _e("Send", 'bender');?></button>
  </div>
</div>
</form>
<?php ContactForm::js_validation(); ?>
<?php } ?>
</div>
</div><!-- /sidebar -->

<?php
$loopClass = '';
$type = 'items';
if(View::newInstance()->_exists('listType')){
  $type = View::newInstance()->_get('listType');
}
if(View::newInstance()->_exists('listClass')){
  $loopClass = View::newInstance()->_get('listClass');
}
?>
<ul class="listing-card-list <?php echo $loopClass; ?>" id="listing-card-list">
  <?php
  $i = 0;

  if($type == 'latestItems'){
    while ( osc_has_latest_items() ) {
      $class = '';
      if($i%3 == 0){
        $class = 'first';
      }
      bender_draw_item($class);
      $i++;
    }
  } elseif($type == 'premiums'){
    while ( osc_has_premiums() ) {
      $class = '';
      if($i%3 == 0){
        $class = 'first';
      }
      bender_draw_item($class, false, true);
      $i++;
      if($i == 3){
        break;
      }
      $i++;
    }
  } else {
    while(osc_has_items()) {
      $i++;
      $class = false;
      if($i%4 == 0){
        $class = 'last';
      }
      $admin = false;
      if(View::newInstance()->_exists("listAdmin")){
        $admin = true;
      }
      bender_draw_item($class, $admin);
    }
  }
?>
</ul>

<?php $size = explode('x', osc_thumbnail_dimensions()); ?>
<li class="listing-card <?php echo $class; if(osc_item_is_premium()){ echo ' premium'; } ?>">
  <?php if( osc_images_enabled_at_items() ) { ?>
  <?php if(osc_count_item_resources()) { ?>
    <a class="listing-thumb" href="<?php echo osc_item_url(); ?>" title="<?php echo
      osc_item_title(); ?>">" width="<?php echo $size[0];
      ?>" height="<?php echo $size[1]; ?>"></a>
  <?php } else { ?>
    <a class="listing-thumb" href="<?php echo osc_item_url(); ?>" title="<?php echo
      osc_item_title(); ?>">" width="<?
        php echo $size[0]; ?>" height="<?php echo $size[1]; ?>"></a>
    <?php } ?>
<?php } ?>
<div class="listing-detail">
    <div class="listing-cell">
        <div class="listing-data">
            <div class="listing-basicinfo">
                <a href="<?php echo osc_item_url() ; ?>" class="title" title="<?php echo
                osc_item_title() ; ?>"><?php echo osc_item_title() ; ?></a>
            <div class="listing-attributes">
                <span class="category"><?php echo osc_item_category() ; ?></span> -
                <span class="location"><?php echo osc_item_city(); ?> (<?php echo
                osc_item_region(); ?>)</span> <span class="g-hide"></span> <?php
                echo osc_format_date(osc_item_pub_date()); ?>
                <?php if( osc_price_enabled_at_items() ) { ?><span class="currency-
                value"><?php echo osc_format_price(osc_item_price()); ?></span><?
                php } ?>
            </div>
            <p><?php echo osc_highlight( strip_tags( osc_item_description() ),250) ;
            ?></p>
        </div>
        <?php if($admin){ ?>
            <span class="admin-options">
                <a href="<?php echo osc_item_edit_url(); ?>" rel="nofollow"><?php _e('
                Edit item', 'bender'); ?></a>
                <span></span>
                <a class="delete" onclick="javascript:return confirm('<?php echo
                osc_esc_js(_('This action can not be undone. Are you sure you
                want to continue?', 'bender')); ?>)" href="<?php echo
                osc_item_delete_url();?>" ><?php _e('Delete', 'bender'); ?></a>
                <?php if(osc_item_is_inactive()) {?>
                <span></span>
                <a href="<?php echo osc_item_activate_url();?>" ><?php _e('Activate',
                'bender'); ?></a>
                <?php } ?>
            </span>
        <?php } ?>
    </div>
</div>
</li>

<?php $size = explode('x', osc_thumbnail_dimensions()); ?>
<li class="listing-card <?php echo $class; ?> premium">
    <?php if( osc_images_enabled_at_items() ) { ?>
        <?php if(osc_count_premium_resources()) { ?>
            <a class="listing-thumb" href="<?php echo osc_premium_url() ; ?>" title="<?php
            echo osc_premium_title() ; ?>">" width="<?php echo
            $size[0]; ?>" height="<?php echo $size[1]; ?>"></a>
        <?php } else { ?>
            <a class="listing-thumb" href="<?php echo osc_premium_url() ; ?>" title="<?php
            echo osc_premium_title() ; ?>">" width="<?php echo $size[0]; ?>" height="<?php echo $size[1]; ?>"></a>
        <?php } ?>
    <?php } ?>
    <div class="listing-detail">
        <div class="listing-cell">
            <div class="listing-data">
                <div class="listing-basicinfo">
                    <a href="<?php echo osc_premium_url() ; ?>" class="title" title="<?php
                    echo osc_premium_title() ; ?>"><?php echo osc_premium_title() ; ?></a>
                <div class="listing-attributes">
                    <span class="category"><?php echo osc_premium_category() ; ?></span> -
                    <span class="location"><?php echo osc_premium_city(); ?> (<?php echo
                    osc_premium_region(); ?>)</span> <span class="g-hide"></span> <?
                    php echo osc_format_date(osc_premium_pub_date()); ?>
                    <?php if( osc_price_enabled_at_items() ) { ?><span class="currency-
                    value"><?php echo osc_format_price(osc_premium_price()); ?></span>
                    <?php } ?>
                </div>
                <p><?php echo osc_highlight( strip_tags( osc_premium_description() ),250)
                ; ?></p>
            </div>
            <?php if($admin){ ?>
                <span class="admin-options">
                    <a href="<?php echo osc_premium_edit_url(); ?>" rel="nofollow"><?php
                    _e('Edit item', 'bender'); ?></a>
                    <span></span>
                    <a class="delete" onclick="javascript:return confirm('<?php echo
                    osc_esc_js(_('This action can not be undone. Are you sure you
                    want to continue?', 'bender')); ?>)" href="<?php echo
                    osc_premium_delete_url();?>" ><?php _e('Delete', 'bender'); ?></a>
                    <?php if(osc_premium_is_inactive()) {?>
                    <span></span>
                    <a href="<?php echo osc_premium_activate_url();?>" ><?php _e('Activate
                    ', 'bender'); ?></a>
                    <?php } ?>
                </span>
            <?php } ?>
        </div>
    </div>
</li>

<?php

```

```

/*
 *      Oclass      software for creating and publishing online classified
 *                  advertising platforms
 *
 *                  Copyright (C) 2013 OSCLASS
 *
 *      This program is free software: you can redistribute it and/or
 *      modify it under the terms of the GNU Affero General Public License
 *      as published by the Free Software Foundation, either version 3 of
 *      the License, or (at your option) any later version.
 *
 *      This program is distributed in the hope that it will be useful, but
 *      WITHOUT ANY WARRANTY; without even the implied warranty of
 *      MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 *      GNU Affero General Public License for more details.
 *
 *      You should have received a copy of the GNU Affero General Public
 *      License along with this program. If not, see <http://www.gnu.org/licenses/>.
 */

// meta tag robots
osc_add_hook('header', 'bender_follow_construct');

bender_add_body_class('home');

$buttonClass = '';
$listClass = '';
if(bender_show_as() == 'gallery'){
    $listClass = 'listing-grid';
    $buttonClass = 'active';
}
?>
<?php osc_current_web_theme_path('header.php'); ?>
<div class="clear"></div>
<div class="latest-ads">
<h1><strong><?php _e('Latest Listings', 'bender'); ?></strong></h1>
<?php if( osc_count_latest_items() == 0) { ?>
<div class="clear"></div>
<p class="empty"><?php _e("There aren't listings available at this moment", 'bender');
?></p>
<?php } else { ?>
<div class="actions">
<span class="doublebutton <?php echo $buttonClass; ?>">
<a href="<?php echo osc_base_url(true); ?>?sShowAs=list" class="list-button" data-
class-toggle="listing-grid" data-destination="#listing-card-list"><span><?php
_e('List', 'bender'); ?></span></a>
<a href="<?php echo osc_base_url(true); ?>?sShowAs=gallery" class="grid-button"
data-class-toggle="listing-grid" data-destination="#listing-card-list"><span><?>
php _e('Grid', 'bender'); ?></span></a>
</span>
</div>
<?php
View::newInstance()->exportVariableToView("listType", 'latestItems');
View::newInstance()->exportVariableToView("listClass", $listClass);
osc_current_web_theme_path('loop.php');
?>
<div class="clear"></div>
<?php if( osc_count_latest_items() == osc_max_latest_items() ) { ?>
<p class="see-more-link"><a href="<?php echo osc_search_show_all_url(); ?>">
<strong><?php _e('See all listings', 'bender'); ?> &raquo;</strong></a>
</p>
<?php } ?>
<?php } ?>
</div>
</div><!-- main -->
<div id="sidebar">
<div class="widget-box">
<?php if(osc_count_list_regions() > 0 ) { ?>
<div class="box location">
<h3><strong><?php _e("Location", 'bender'); ?></strong></h3>
<ul>
<?php while(osc_has_list_regions() ) { ?>
<li><a href="<?php echo osc_list_region_url(); ?>"><?php echo
osc_list_region_name(); ?> <em><?php echo osc_list_region_items(); ?>
</em></a></li>
<?php } ?>
</ul>
</div>
<?php } ?>
</div>
</div>
</div>
<div class="clear"><!-- do not close, use main closing tag for this case -->
<?php osc_current_web_theme_path('footer.php'); ?>

<?php
/*
 *      Oclass      software for creating and publishing online classified
 *                  advertising platforms
 *
 *                  Copyright (C) 2013 OSCLASS
 *
 *      This program is free software: you can redistribute it and/or
 *      modify it under the terms of the GNU Affero General Public License
 *      as published by the Free Software Foundation, either version 3 of
 *      the License, or (at your option) any later version.
 *
 *      This program is distributed in the hope that it will be useful, but
 *      WITHOUT ANY WARRANTY; without even the implied warranty of

```

```

*          MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
*          GNU Affero General Public License for more details.
*
* You should have received a copy of the GNU Affero General Public
* License along with this program. If not, see <http://www.gnu.org/licenses/>.
*/

// meta tag robots
osc_add_hook('header', 'bender_nofollow_construct');

bender_add_body_class('page');
osc_current_web_theme_path('header.php');
?>
<h1><?php echo osc_static_page_title(); ?></h1>
<?php echo osc_static_page_text(); ?>
<?php osc_current_web_theme_path('footer.php'); ?>

<?php
/*
*          Oclass      software for creating and publishing online classified
*                      advertising platforms
*
*                      Copyright (C) 2013 OSCLASS
*
* This program is free software: you can redistribute it and/or
* modify it under the terms of the GNU Affero General Public License
* as published by the Free Software Foundation, either version 3 of
* the License, or (at your option) any later version.
*
* This program is distributed in the hope that it will be useful, but
* WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
* GNU Affero General Public License for more details.
*
* You should have received a copy of the GNU Affero General Public
* License along with this program. If not, see <http://www.gnu.org/licenses/>.
*/

// meta tag robots
if( osc_count_items() == 0 || stripos($_SERVER['REQUEST_URI'], 'search') ) {
    osc_add_hook('header', 'bender_nofollow_construct');
} else {
    osc_add_hook('header', 'bender_follow_construct');
}

bender_add_body_class('search');
$listClass = '';
$buttonClass = '';
if(osc_search_show_as() == 'gallery'){
    $listClass = 'listing-grid';
    $buttonClass = 'active';
}
osc_add_hook('before-main', 'sidebar');
function sidebar(){
    osc_current_web_theme_path('search-sidebar.php');
}
osc_add_hook('footer', 'autocompleteCity');
function autocompleteCity(){ ?>
<script type="text/javascript">
$(function() {
    function log( message ) {
        $( "<div/>" ).text( message ).prependTo( "#log" );
        $( "#log" ).attr( "scrollTop", 0 );
    }

    $( "#sCity" ).autocomplete({
        source: "<?php echo osc_base_url(true); ?>?page=ajax&action=location",
        minLength: 2,
        select: function( event, ui ) {
            $( "#sRegion" ).attr( "value", ui.item.region );
            log( ui.item ?
                "<?php _e('Selected', 'modern'); ?>: " + ui.item.value + " aka
                " + ui.item.id :
                "<?php _e('Nothing selected, input was', 'modern'); ?> " +
                this.value );
        }
    });
});
</script>
<?php
}

?>
<?php osc_current_web_theme_path('header.php'); ?>
<div class="list-header">
    <div class="resp-wrapper">
        <?php osc_run_hook('search_ads_listing_top'); ?>
        <h1><?php echo search_title(); ?></h1>

        <?php if(osc_count_items() == 0) { ?>
            <p class="empty"><?php printf(--('There are no results matching "%s"', '
            bender'), osc_search_pattern()); ?></p>
        <?php } else { ?>
            <span class="counter-search"><?php
                $search_number = bender_search_number();
                printf(--('%1$d - %2$d of %3$d listings', 'bender'), $search_number['from'],
                $search_number['to'], $search_number['of']);
            ?></span>
            <div class="actions">

```

```

<a href="#" data-bclass-toggle="display-filters" class="resp-toogle show-filters
-btn"><?php _e('Show filters ', 'bender'); ?></a>
<span class="doublebutton <?php echo $buttonClass; ?>">
  <a href="<?php echo osc_esc_html(osc_update_search_url(array('showAs'=> '
list ')); ?>" class="list-button" data-class-toggle="listing-grid" data
-destination="#listing-card-list"><span><?php _e('List ', 'bender'); ?></
span></a>
  <a href="<?php echo osc_esc_html(osc_update_search_url(array('showAs'=> '
gallery ')); ?>" class="grid-button" data-class-toggle="listing-grid"
data-destination="#listing-card-list"><span><?php _e('Grid ', 'bender');
?></span></a>
</span>
<!-- START sort by -->
<span class="see-by">
<span><?php _e('Sort by ', 'bender'); ?></span>
<?php
$order = osc_list_orders();
$current = '';
foreach($order as $label => $params) {
  $orderType = ($params['iOrderType'] == 'asc') ? '0' : '1';
  if(osc_search_order() == $params['sOrder'] && osc_search_order_type() ==
  $orderType) {
    $current = $label;
  }
}
?>
<label><?php echo $current; ?><b class="arrow-envelope"><b class="arrow-down"></
b></b></label>
<?php $i = 0; ?>
<ul>
<?php
foreach($order as $label => $params) {
  $orderType = ($params['iOrderType'] == 'asc') ? '0' : '1'; ?>
  <?php if(osc_search_order() == $params['sOrder'] &&
  osc_search_order_type() == $orderType) { ?>
    <li><a class="current" href="<?php echo osc_esc_html(
    osc_update_search_url($params)); ?>"><?php echo $label; ?></a></
    li>
    <?php } else { ?>
    <li><a href="<?php echo osc_esc_html(osc_update_search_url($params))
    ; ?>"><?php echo $label; ?></a></li>
    <?php } ?>
    <?php $i++; ?>
  <?php } ?>
</ul>
</span>
<!-- END sort by -->
</div>
<?php } ?>
</div>
<?php
$i = 0;
osc_get_premiums();
if(osc_count_premiums() > 0) {
echo '<h5>'. _e('Premium listings ', 'bender'). '</h5>';
View::newInstance()->exportVariableToView("listType", 'premiums');
View::newInstance()->exportVariableToView("listClass", $listClass.' premium-list ');
osc_current_web_theme_path('loop.php');
}
?>
<?php if(osc_count_items() > 0) {
echo '<h5>'. _e('Listings ', 'bender'). '</h5>';
View::newInstance()->exportVariableToView("listType", 'items');
View::newInstance()->exportVariableToView("listClass", $listClass);
osc_current_web_theme_path('loop.php');
}
?>
<div class="clear"></div>
<?php
if(osc_rewrite_enabled()){
$footerLinks = osc_search_footer_links();
if(count($footerLinks)>0) {
?>
<div id="related-searches">
<h5><?php _e('Other searches that may interest you ', 'bender'); ?></h5>
<ul class="footer-links">
<?php foreach($footerLinks as $f) { View::newInstance()->exportVariableToView('
footer_link ', $f); ?>
<?php if($f['total'] < 3) continue; ?>
<li><a href="<?php echo osc_footer_link_url(); ?>"><?php echo
osc_footer_link_title(); ?></a></li>
<?php } ?>
</ul>
</div>
<?php }
} ?>
<div class="paginate" >
<?php echo osc_search_pagination(); ?>
</div>
<?php } ?>
<?php osc_current_web_theme_path('footer.php') ; ?>
<?php
/*
*      Osclass      software for creating and publishing online classified
*                    advertising platforms

```



```

*
*
* Copyright (C) 2013 OSCLASS
*
* This program is free software: you can redistribute it and/or
* modify it under the terms of the GNU Affero General Public License
* as published by the Free Software Foundation, either version 3 of
* the License, or (at your option) any later version.
*
* This program is distributed in the hope that it will be useful, but
* WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
* GNU Affero General Public License for more details.
*
* You should have received a copy of the GNU Affero General Public
* License along with this program. If not, see <http://www.gnu.org/licenses/>.
*/
$category = --get("category");
if(!isset($category['pk.i.id'])) {
    $category['pk.i.id'] = null;
}

?>
<div id="sidebar">
<?php osc_alert_form(); ?>
<div class="filters">
    <form action="<?php echo osc_base_url(true); ?>" method="get" class="nocsrf">
        <input type="hidden" name="page" value="search"/>
        <input type="hidden" name="sOrder" value="<?php echo osc_search_order(); ?>" />
        <input type="hidden" name="iOrderType" value="<?php $allowedTypesForSorting = Search::
            getAllowedTypesForSorting(); echo $allowedTypesForSorting[osc_search_order_type()
            ]; ?>" />
        <?php foreach(osc_search_user() as $userId) { ?>
        <input type="hidden" name="sUser[]" value="<?php echo $userId; ?>" />
        <?php } ?>
        <fieldset class="first">
            <h3><?php _e('Your search', 'bender'); ?></h3>
            <div class="row">
                <input class="input-text" type="text" name="sPattern" id="query" value="<?php
                    echo osc_esc_html(osc_search_pattern()); ?>" />
            </div>
        </fieldset>
        <fieldset>
            <h3><?php _e('City', 'bender'); ?></h3>
            <div class="row">
                <input class="input-text" type="text" id="sCity" name="sCity" value="<?php
                    echo osc_esc_html(osc_search_city()); ?>" />
            </div>
        </fieldset>
        <?php if( osc_images_enabled_at_items() ) { ?>
        <fieldset>
            <h3><?php _e('Show only', 'bender') ; ?></h3>
            <div class="row">
                <input type="checkbox" name="bPic" id="withPicture" value="1" <?php echo (
                    osc_search_has_pic() ? 'checked' : '' ); ?> />
                <label for="withPicture"><?php _e('listings with pictures', 'bender') ; ?></
                    label>
            </div>
        </fieldset>
        <?php } ?>
        <?php if( osc_price_enabled_at_items() ) { ?>
        <fieldset>
            <div class="row price-slice">
                <h3><?php _e('Price', 'bender') ; ?></h3>
                <span><?php _e('Min', 'bender') ; ?></span>
                <input class="input-text" type="text" id="priceMin" name="sPriceMin" value="<?
                    php echo osc_esc_html(osc_search_price_min()); ?>" size="6" maxlength="6"
                    />
                <span><?php _e('Max', 'bender') ; ?></span>
                <input class="input-text" type="text" id="priceMax" name="sPriceMax" value="<?
                    php echo osc_esc_html(osc_search_price_max()); ?>" size="6" maxlength="6"
                    />
            </div>
        </fieldset>
        <?php } ?>
        <div class="plugin-hooks">
            <?php
                if(osc_search_category_id()) {
                    osc_run_hook('search-form', osc_search_category_id());
                } else {
                    osc_run_hook('search-form');
                }
            ?>
        </div>
        <?php
            $aCategories = osc_search_category();
            foreach($aCategories as $cat_id) { ?>
                <input type="hidden" name="sCategory[]" value="<?php echo osc_esc_html($cat_id);
                    ?>" />
            <?php } ?>
        <div class="actions">
            <button type="submit"><?php _e('Apply', 'bender') ; ?></button>
        </div>
    </form>
</fieldset>
    <div class="row">
        <h3><?php _e('Refine category', 'bender') ; ?></h3>
        <?php bender_sidebar_category_search($category['pk.i.id']); ?>
    </div>
</fieldset>

```

```

</div>
</div>

<?php
/*
 *      Oclass      software for creating and publishing online classified
 *                  advertising platforms
 *
 *                  Copyright (C) 2013 OSCLASS
 *
 *      This program is free software: you can redistribute it and/or
 *      modify it under the terms of the GNU Affero General Public License
 *      as published by the Free Software Foundation, either version 3 of
 *      the License, or (at your option) any later version.
 *
 *      This program is distributed in the hope that it will be useful, but
 *      WITHOUT ANY WARRANTY; without even the implied warranty of
 *      MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 *      GNU Affero General Public License for more details.
 *
 *      You should have received a copy of the GNU Affero General Public
 *      License along with this program. If not, see <http://www.gnu.org/licenses/>.
 */

// meta tag robots
osc_add_hook('header', 'bender_nofollow_construct');

bender_add_body_class('user user-profile');
osc_add_hook('before-main', 'sidebar');
function sidebar(){
    osc_current_web_theme_path('user-sidebar.php');
}
osc_add_filter('meta_title_filter', 'custom_meta_title');
function custom_meta_title($data){
    return __('Alerts', 'bender');
}
osc_current_web_theme_path('header.php');
$osc_user = osc_user();
?>
<h1><?php _e('Alerts', 'bender'); ?></h1>
<?php if(osc_count_alerts() == 0) { ?>
<h3><?php _e('You do not have any alerts yet', 'bender'); ?></h3>
<?php } else { ?>
<?php
    $i = 1;
    while(osc_has_alerts()) { ?>
        <div class="userItem" >
            <div class="title -has-actions">
                <h3><?php _e('Alert', 'bender'); ?> <?php echo $i; ?></h3> <a onclick="
                    javascript: return confirm('<?php echo osc_esc_js(__('This action can\'t be
                    undone. Are you sure you want to continue?', 'benderw')); ?>');" href="<?
                    php echo osc_user_unsubscribe_alert_url(); ?>"><?php _e('Delete this alert
                    ', 'bender'); ?></a><div class="clear"></div></div>

                <div>
                    <?php osc_current_web_theme_path('loop.php') ; ?>
                    <?php if(osc_count_items() == 0) { ?>
                        <br />
                        0 <?php _e('Listings', 'bender'); ?>
                    <?php } ?>
                </div>
            </div>
            <br />
        <?php
            $i++;
        }
    }
?>
<?php } ?>
<?php osc_current_web_theme_path('footer.php') ; ?>

<?php
/*
 *      Oclass      software for creating and publishing online classified
 *                  advertising platforms
 *
 *                  Copyright (C) 2013 OSCLASS
 *
 *      This program is free software: you can redistribute it and/or
 *      modify it under the terms of the GNU Affero General Public License
 *      as published by the Free Software Foundation, either version 3 of
 *      the License, or (at your option) any later version.
 *
 *      This program is distributed in the hope that it will be useful, but
 *      WITHOUT ANY WARRANTY; without even the implied warranty of
 *      MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 *      GNU Affero General Public License for more details.
 *
 *      You should have received a copy of the GNU Affero General Public
 *      License along with this program. If not, see <http://www.gnu.org/licenses/>.
 */

// meta tag robots
osc_add_hook('header', 'bender_nofollow_construct');

osc_enqueue_script('jquery-validate');
bender_add_body_class('user user-profile');
osc_add_hook('before-main', 'sidebar');
function sidebar(){
    osc_current_web_theme_path('user-sidebar.php');
}

```

```

    }
    osc_add_filter('meta_title_filter', 'custom_meta_title');
    function custom_meta_title($data){
        return __('Change e-mail', 'bender');
    }
    osc_current_web_theme_path('header.php') ;
    $osc_user = osc_user();
?>
<h1><?php _e('Change e-mail', 'bender'); ?></h1>
<div class="form-container form-horizontal">
    <div class="resp-wrapper">
        <ul id="error_list"></ul>
        <form action="<?php echo osc_base_url(true); ?>" method="post">
            <input type="hidden" name="page" value="user" />
            <input type="hidden" name="action" value="change_email_post" />
            <div class="control-group">
                <label for="email"><?php _e('Current e-mail', 'bender'); ?></label>
                <div class="controls">
                    <?php echo osc_logged_user_email(); ?>
                </div>
            </div>
            <div class="control-group">
                <label class="control-label" for="new_email"><?php _e('New e-mail', 'bender');
                    ?> *</label>
                <div class="controls">
                    <input type="text" name="new_email" id="new_email" value="" />
                </div>
            </div>
            <div class="control-group">
                <div class="controls">
                    <button type="submit" class="ui-button ui-button-middle ui-button-main"><?
                        php _e("Update", 'bender');?></button>
                </div>
            </div>
        </form>
    </div>
</div>
<script type="text/javascript">
    $(document).ready(function() {
        $('form#change-email').validate({
            rules: {
                new_email: {
                    required: true,
                    email: true
                }
            },
            messages: {
                new_email: {
                    required: '?php echo osc_esc_js(__( "Email: this field is required", "
                        bender")); ?>.',
                    email: '<?php echo osc_esc_js(__( "Invalid email address", "bender")); ?>.'
                }
            },
            errorLabelContainer: "#error_list",
            wrapper: "li",
            invalidHandler: function(form, validator) {
                $('html, body').animate({ scrollTop: $('h1').offset().top }, { duration: 250,
                    easing: 'swing' });
            },
            submitHandler: function(form) {
                $('button[type=submit], input[type=submit]').attr('disabled', 'disabled');
                form.submit();
            }
        });
    });
</script>
<?php osc_current_web_theme_path('footer.php') ; ?>

<?php
/*
 *      Oclass      software for creating and publishing online classified
 *                  advertising platforms
 *
 *                  Copyright (C) 2013 OSCLASS
 *
 *      This program is free software: you can redistribute it and/or
 *      modify it under the terms of the GNU Affero General Public License
 *      as published by the Free Software Foundation, either version 3 of
 *      the License, or (at your option) any later version.
 *
 *      This program is distributed in the hope that it will be useful, but
 *      WITHOUT ANY WARRANTY; without even the implied warranty of
 *      MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 *      GNU Affero General Public License for more details.
 *
 *      You should have received a copy of the GNU Affero General Public
 *      License along with this program. If not, see <http://www.gnu.org/licenses/>.
 */

// meta tag robots
osc_add_hook('header', 'bender_nofollow_construct');

bender_add_body_class('user user-profile');
osc_add_hook('before-main', 'sidebar');
function sidebar(){
    osc_current_web_theme_path('user-sidebar.php');
}
osc_add_filter('meta_title_filter', 'custom_meta_title');
function custom_meta_title($data){

```

```

        return --('Change password', 'bender');
    }
    osc_current_web_theme_path('header.php');
    $osc_user = osc_user();
?>
<h1><?php _e('Change password', 'bender'); ?></h1>
<div class="form-container form-horizontal">
    <div class="resp-wrapper">
        <ul id="error_list"></ul>
        <form action="<?php echo osc_base_url(true); ?>" method="post">
            <input type="hidden" name="page" value="user" />
            <input type="hidden" name="action" value="change-password-post" />
            <div class="control-group">
                <label class="control-label" for="password"><?php _e('Current password', '
                bender'); ?> *</label>
                <div class="controls">
                    <input type="password" name="password" id="password" value="" />
                </div>
            </div>
            <div class="control-group">
                <label class="control-label" for="new_password"><?php _e('New password', '
                bender'); ?> *</label>
                <div class="controls">
                    <input type="password" name="new_password" id="new_password" value="" />
                </div>
            </div>
            <div class="control-group">
                <label class="control-label" for="new_password2"><?php _e('Repeat new password
                ', 'bender'); ?> *</label>
                <div class="controls">
                    <input type="password" name="new_password2" id="new_password2" value="" />
                </div>
            </div>
            <div class="control-group">
                <div class="controls">
                    <button type="submit" class="ui-button ui-button-middle ui-button-main"><?
                    php _e("Update", 'bender'); ?></button>
                </div>
            </div>
        </form>
    </div>
</div>
<?php osc_current_web_theme_path('footer.php'); ?>

<?php
/*
 *      Oclass      software for creating and publishing online classified
 *                  advertising platforms
 *
 *      Copyright (C) 2013 OSCLASS
 *
 *      This program is free software: you can redistribute it and/or
 *      modify it under the terms of the GNU Affero General Public License
 *      as published by the Free Software Foundation, either version 3 of
 *      the License, or (at your option) any later version.
 *
 *      This program is distributed in the hope that it will be useful, but
 *      WITHOUT ANY WARRANTY; without even the implied warranty of
 *      MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 *      GNU Affero General Public License for more details.
 *
 *      You should have received a copy of the GNU Affero General Public
 *      License along with this program. If not, see <http://www.gnu.org/licenses/>.
 */

// meta tag robots
osc_add_hook('header', 'bender_nofollow_construct');

osc_enqueue_script('jquery-validate');

bender_add_body_class('user user-profile');
osc_add_hook('before-main', 'sidebar');
function sidebar(){
    osc_current_web_theme_path('user-sidebar.php');
}
osc_add_filter('meta_title_filter', 'custom_meta_title');
function custom_meta_title($data){
    return --('Change username', 'bender');
}
osc_current_web_theme_path('header.php');
$osc_user = osc_user();
?>
<h1><?php _e('Change username', 'bender'); ?></h1>
<script type="text/javascript">
$(document).ready(function() {
    $('#form#change-username').validate({
        rules: {
            s_username: {
                required: true
            }
        },
        messages: {
            s_username: {
                required: '<?php echo osc_esc_js(--("Username: this field is required", "
                bender")); ?>.'
            }
        }
    },
    errorLabelContainer: "#error_list",
    wrapper: "li",

```

```

        invalidHandler: function(form, validator) {
            $('html,body').animate({ scrollTop: $('hl').offset().top }, { duration: 250,
                easing: 'swing' });
        },
        submitHandler: function(form) {
            $('#button[type=submit], input[type=submit]').attr('disabled', 'disabled');
            form.submit();
        }
    });

    var cInterval;
    $('#s_username').keydown(function(event) {
        if($('#s_username').attr("value")!='') {
            clearInterval(cInterval);
            cInterval = setInterval(function() {
                $.getJSON(
                    "<?php echo osc_base_url(true); ?>?page=ajax&action=
                    check_username_availability",
                    { "s_username": $('#s_username').attr("value") },
                    function(data) {
                        clearInterval(cInterval);
                        if(data.exists==0) {
                            $('#available').text('<?php echo osc_esc_js(--("The username is
                            available", "bender")); ?>');
                        } else {
                            $('#available').text('<?php echo osc_esc_js(--("The username is
                            NOT available", "bender")); ?>');
                        }
                    }
                );
            }, 1000);
        }
    });
}
</script>
<div class="form-container form-horizontal">
    <div class="resp-wrapper">
        <ul id="error_list"></ul>
        <form action="<?php echo osc_base_url(true); ?>" method="post" id="change-username">
            <input type="hidden" name="page" value="user" />
            <input type="hidden" name="action" value="change_username_post" />
            <div class="control-group">
                <label class="control-label" for="s_username"><?php _e('Username', 'bender');
                ?></label>
                <div class="controls">
                    <input type="text" name="s_username" id="s_username" value="" />
                    <div id="available"></div>
                </div>
            </div>
            <div class="control-group">
                <div class="controls">
                    <button type="submit" class="ui-button ui-button-middle ui-button-main"><?
                    php _e("Update", 'bender');?></button>
                </div>
            </div>
        </form>
    </div>
</div>
<?php osc_current_web_theme_path('footer.php') ; ?>

<?php
/*
 *      Oclass      software for creating and publishing online classified
 *                  advertising platforms
 *
 *      Copyright (C) 2013 OSCLASS
 *
 *      This program is free software: you can redistribute it and/or
 *      modify it under the terms of the GNU Affero General Public License
 *      as published by the Free Software Foundation, either version 3 of
 *      the License, or (at your option) any later version.
 *
 *      This program is distributed in the hope that it will be useful, but
 *      WITHOUT ANY WARRANTY; without even the implied warranty of
 *      MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 *      GNU Affero General Public License for more details.
 *
 *      You should have received a copy of the GNU Affero General Public
 *      License along with this program. If not, see <http://www.gnu.org/licenses/>.
 */

// meta tag robots
osc_add_hook('header', 'bender_nofollow_construct');

bender_add_body_class('user_user-custom');
osc_add_hook('before-main', 'sidebar');
function sidebar() {
    osc_current_web_theme_path('user-sidebar.php');
}
osc_current_web_theme_path('header.php') ;

osc_render_file();

osc_current_web_theme_path('footer.php');
?>

<?php

```

```

/*
 *   Oclass      software for creating and publishing online classified
 *               advertising platforms
 *
 *               Copyright (C) 2013 OSCLASS
 *
 *   This program is free software: you can redistribute it and/or
 *   modify it under the terms of the GNU Affero General Public License
 *   as published by the Free Software Foundation, either version 3 of
 *   the License, or (at your option) any later version.
 *
 *   This program is distributed in the hope that it will be useful, but
 *   WITHOUT ANY WARRANTY; without even the implied warranty of
 *   MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 *   GNU Affero General Public License for more details.
 *
 *   You should have received a copy of the GNU Affero General Public
 *   License along with this program. If not, see <http://www.gnu.org/licenses/>.
 */

// meta tag robots
osc_current_web_theme_path('user-items.php') ;
?>

<?php
/*
 *   Oclass      software for creating and publishing online classified
 *               advertising platforms
 *
 *               Copyright (C) 2013 OSCLASS
 *
 *   This program is free software: you can redistribute it and/or
 *   modify it under the terms of the GNU Affero General Public License
 *   as published by the Free Software Foundation, either version 3 of
 *   the License, or (at your option) any later version.
 *
 *   This program is distributed in the hope that it will be useful, but
 *   WITHOUT ANY WARRANTY; without even the implied warranty of
 *   MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 *   GNU Affero General Public License for more details.
 *
 *   You should have received a copy of the GNU Affero General Public
 *   License along with this program. If not, see <http://www.gnu.org/licenses/>.
 */

// meta tag robots
osc_add_hook('header', 'bender_nofollow_construct');

bender_add_body_class('forgot');
osc_current_web_theme_path('header.php');
?>
<div class="form-container form-horizontal form-container-box">
  <div class="header">
    <h1><?php _e('Recover your password', 'bender'); ?></h1>
  </div>
  <div class="resp-wrapper">
    <form action="<?php echo osc_base_url(true); ?>" method="post" >
      <input type="hidden" name="page" value="login" />
      <input type="hidden" name="action" value="forgot_post" />
      <input type="hidden" name="userId" value="<?php echo osc_esc_html(Params::getParam('userId')); ?>" />
      <input type="hidden" name="code" value="<?php echo osc_esc_html(Params::getParam('code')); ?>" />
      <div class="control-group">
        <label class="control-label" for="new-password"><?php _e('New password', 'bender'); ?></label>
        <div class="controls">
          <input type="password" name="new-password" value="" />
        </div>
      </div>
      <div class="control-group">
        <label class="control-label" for="new-password2"><?php _e('Repeat new password', 'bender'); ?></label>
        <div class="controls">
          <input type="password" name="new-password2" value="" />
        </div>
      </div>
      <div class="control-group">
        <div class="controls">
          <button type="submit" class="ui-button ui-button-middle ui-button-main"><?php _e("Change password", 'bender'); ?></button>
        </div>
      </div>
    </form>
  </div>
</div>
<?php osc_current_web_theme_path('footer.php') ; ?>

<?php
/*
 *   Oclass      software for creating and publishing online classified
 *               advertising platforms
 *
 *               Copyright (C) 2013 OSCLASS
 *
 *   This program is free software: you can redistribute it and/or
 *   modify it under the terms of the GNU Affero General Public License
 *   as published by the Free Software Foundation, either version 3 of

```

```

*           the License, or (at your option) any later version.
*
*   This program is distributed in the hope that it will be useful, but
*   WITHOUT ANY WARRANTY; without even the implied warranty of
*   MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
*   GNU Affero General Public License for more details.
*
*   You should have received a copy of the GNU Affero General Public
*   License along with this program. If not, see <http://www.gnu.org/licenses/>.
*/

// meta tag robots
osc_add_hook('header', 'bender_nofollow_construct');

bender_add_body_class('user user-items');
osc_add_hook('before-main', 'sidebar');
function sidebar(){
    osc_current_web_theme_path('user-sidebar.php');
}
osc_current_web_theme_path('header.php');

$listClass = '';
$buttonClass = '';
if(Params::getParam('ShowAs') == 'gallery'){
    $listClass = 'listing-grid';
    $buttonClass = 'active';
}
?>
<div class="list-header">
<?php osc_run_hook('search-ads-listing-top'); ?>
<h1><?php _e('My listings', 'bender'); ?></h1>
<?php if(osc_count_items() == 0) { ?>
    <p class="empty"><?php _e('No listings have been added yet', 'bender'); ?></p>
<?php } else { ?>
    <div class="actions">
        <span class="doublebutton <?php echo $buttonClass; ?>">
            <a href="<?php echo osc_user_list_items_url(); ?>?ShowAs=list" class="list-button" data-class-toggle="listing-grid" data-destination="#listing-card-list"><span>Lista</span></a>
            <a href="<?php echo osc_user_list_items_url(); ?>?ShowAs=gallery" class="grid-button" data-class-toggle="listing-grid" data-destination="#listing-card-list"><span>Grid</span></a>
        </span>
    </div>
</div>
<?php
View::newInstance()->exportVariableToView("listClass", $listClass);
View::newInstance()->exportVariableToView("listAdmin", true);
osc_current_web_theme_path('loop.php');
?>
<div class="clear"></div>
<?php
if(osc_rewrite_enabled()){
    $footerLinks = osc_search_footer_links();
?>
<ul class="footer-links">
    <?php foreach($footerLinks as $f) { View::newInstance()->exportVariableToView('
        footer_link', $f); ?>
        <?php if($f['total'] < 3) continue; ?>
        <li><a href="<?php echo osc_footer_link_url(); ?>"><?php echo
            osc_footer_link_title(); ?></a></li>
    <?php } ?>
</ul>
<?php } ?>
<div class="paginate" >
    <?php echo osc_search_pagination(); ?>
</div>
<?php } ?>
<?php osc_current_web_theme_path('footer.php'); ?>

<?php
/*
*   Oclass software for creating and publishing online classified
*   advertising platforms
*
*   Copyright (C) 2013 OSCLASS
*
*   This program is free software: you can redistribute it and/or
*   modify it under the terms of the GNU Affero General Public License
*   as published by the Free Software Foundation, either version 3 of
*   the License, or (at your option) any later version.
*
*   This program is distributed in the hope that it will be useful, but
*   WITHOUT ANY WARRANTY; without even the implied warranty of
*   MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
*   GNU Affero General Public License for more details.
*
*   You should have received a copy of the GNU Affero General Public
*   License along with this program. If not, see <http://www.gnu.org/licenses/>.
*/

// meta tag robots
osc_add_hook('header', 'bender_nofollow_construct');

bender_add_body_class('login');
osc_current_web_theme_path('header.php');
?>
<div class="form-container form-horizontal form-container-box">
<div class="header">

```

```

        <h1><?php _e('Access to your account', 'bender'); ?></h1>
    </div>
    <div class="resp-wrapper">
        <form action="<?php echo osc_base_url(true); ?>" method="post" >
            <input type="hidden" name="page" value="login" />
            <input type="hidden" name="action" value="login_post" />

            <div class="control-group">
                <label class="control-label" for="email"><?php _e('E-mail', 'bender'); ?></label>
                <div class="controls">
                    <?php UserForm::email_login_text(); ?>
                </div>
            </div>
            <div class="control-group">
                <label class="control-label" for="password"><?php _e('Password', 'bender'); ?></label>
                <div class="controls">
                    <?php UserForm::password_login_text(); ?>
                </div>
            </div>
            <div class="control-group">
                <div class="controls checkbox">
                    <?php UserForm::rememberme_login_checkbox();?> <label for="remember"><?php _e('Remember me', 'bender'); ?></label>
                </div>
                <div class="controls">
                    <button type="submit" class="ui-button ui-button-middle ui-button-main"><?php _e("Log in", 'bender');?></button>
                </div>
            </div>
            <div class="actions">
                <a href="<?php echo osc_register_account_url(); ?>"><?php _e("Register for a free account", 'bender'); ?></a><br /><a href="<?php echo osc_recover_user_password_url(); ?>"><?php _e("Forgot password?", 'bender'); ?></a>
            </div>
        </form>
    </div>
</div>
<?php osc_current_web_theme_path('footer.php') ; ?>

<?php
/*
    *       Oclass       software for creating and publishing online classified
    *                   advertising platforms
    *
    *                   Copyright (C) 2013 OSCLASS
    *
    *       This program is free software: you can redistribute it and/or
    *       modify it under the terms of the GNU Affero General Public License
    *       as published by the Free Software Foundation, either version 3 of
    *       the License, or (at your option) any later version.
    *
    *       This program is distributed in the hope that it will be useful, but
    *       WITHOUT ANY WARRANTY; without even the implied warranty of
    *       MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
    *       GNU Affero General Public License for more details.
    *
    *       You should have received a copy of the GNU Affero General Public
    *       License along with this program. If not, see <http://www.gnu.org/licenses/>.
    */

// meta tag robots
osc_add_hook('header', 'bender_nofollow_construct');

bender_add_body_class('user user-profile');
osc_add_hook('before-main', 'sidebar');
function sidebar(){
    osc_current_web_theme_path('user-sidebar.php');
}
osc_add_filter('meta_title_filter', 'custom_meta_title');
function custom_meta_title($data){
    return _e('Update account', 'bender');
}
osc_current_web_theme_path('header.php') ;
$oosc_user = osc_user();
?>
<h1><?php _e('Update account', 'bender'); ?></h1>
<?php UserForm::location_javascript(); ?>
<div class="form-container form-horizontal">
    <div class="resp-wrapper">
        <ul id="error_list"></ul>
        <form action="<?php echo osc_base_url(true); ?>" method="post">
            <input type="hidden" name="page" value="user" />
            <input type="hidden" name="action" value="profile_post" />
            <div class="control-group">
                <label class="control-label" for="name"><?php _e('Name', 'bender'); ?></label>
                <div class="controls">
                    <?php UserForm::name_text(osc_user()); ?>
                </div>
            </div>
            <div class="control-group">
                <label class="control-label" for="user_type"><?php _e('User type', 'bender'); ?></label>
                <div class="controls">
                    <?php UserForm::is_company_select(osc_user()); ?>
                </div>
            </div>
        </form>
    </div>

```



```

<div class="control-group">
  <label class="control-label" for="phoneMobile"><?php _e('Cell phone', 'bender
  '); ?></label>
  <div class="controls">
    <?php UserForm::mobile_text(osc_user()); ?>
  </div>
</div>
<div class="control-group">
  <label class="control-label" for="phoneLand"><?php _e('Phone', 'bender'); ?></
  label>
  <div class="controls">
    <?php UserForm::phone_land_text(osc_user()); ?>
  </div>
</div>
<div class="control-group">
  <label class="control-label" for="country"><?php _e('Country', 'bender'); ?></
  label>
  <div class="controls">
    <?php UserForm::country_select(osc_get_countries(), osc_user()); ?>
  </div>
</div>
<div class="control-group">
  <label class="control-label" for="region"><?php _e('Region', 'bender'); ?></
  label>
  <div class="controls">
    <?php UserForm::region_select(osc_get_regions(), osc_user()); ?>
  </div>
</div>
<div class="control-group">
  <label class="control-label" for="city"><?php _e('City', 'bender'); ?></label>
  <div class="controls">
    <?php UserForm::city_select(osc_get_cities(), osc_user()); ?>
  </div>
</div>
<div class="control-group">
  <label class="control-label" for="city_area"><?php _e('City area', 'bender');
  ?></label>
  <div class="controls">
    <?php UserForm::city_area_text(osc_user()); ?>
  </div>
</div>
<div class="control-group">
  <label class="control-label"1 for="address"><?php _e('Address', 'bender');
  ?></label>
  <div class="controls">
    <?php UserForm::address_text(osc_user()); ?>
  </div>
</div>
<div class="control-group">
  <label class="control-label" for="webSite"><?php _e('Website', 'bender'); ?></
  label>
  <div class="controls">
    <?php UserForm::website_text(osc_user()); ?>
  </div>
</div>
<div class="control-group">
  <label class="control-label" for="s_info"><?php _e('Description', 'bender');
  ?></label>
  <div class="controls">
    <?php UserForm::info_textarea('s_info', osc_locale_code(), @$osc_user['
    locale'][osc_locale_code()][ 's_info ']); ?>
  </div>
</div>
<div class="control-group">
  <div class="controls">
    <button type="submit" class="ui-button ui-button-middle ui-button-main"><?
    php _e("Update", 'bender'); ?></button>
  </div>
</div>
<div class="control-group">
  <div class="controls">
    <?php osc_run_hook('user_form'); ?>
  </div>
</div>
</form>
</div>
<?php osc_current_web_theme_path('footer.php') ; ?>

<?php
/*
 *      Oclass      software for creating and publishing online classified
 *                  advertising platforms
 *
 *                  Copyright (C) 2013 OSCLASS
 *
 *      This program is free software: you can redistribute it and/or
 *      modify it under the terms of the GNU Affero General Public License
 *      as published by the Free Software Foundation, either version 3 of
 *      the License, or (at your option) any later version.
 *
 *      This program is distributed in the hope that it will be useful, but
 *      WITHOUT ANY WARRANTY; without even the implied warranty of
 *      MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 *      GNU Affero General Public License for more details.
 *
 *      You should have received a copy of the GNU Affero General Public
 *      License along with this program. If not, see <http://www.gnu.org/licenses/>.
 */

```

```

// meta tag robots
osc_add_hook('header', 'bender_follow_construct');

$address = '';
if(osc_user_address()!='') {
    if(osc_user_city_area()!='') {
        $address = osc_user_address().", ".osc_user_city_area();
    } else {
        $address = osc_user_address();
    }
} else {
    $address = osc_user_city_area();
}
$location_array = array();
if(trim(osc_user_city().". ".osc_user_zip())!='') {
    $location_array[] = trim(osc_user_city().". ".osc_user_zip());
}
if(osc_user_region()!='') {
    $location_array[] = osc_user_region();
}
if(osc_user_country()!='') {
    $location_array[] = osc_user_country();
}
$location = implode(" ", $location_array);
unset($location_array);

osc_enqueue_script('jquery-validate');

bender_add_body_class('user-public-profile');
osc_add_hook('after-main', 'sidebar');
function sidebar(){
    osc_current_web_theme_path('user-public-sidebar.php');
}

osc_current_web_theme_path('header.php');
?>
<div id="item-content">
<div class="user-card">

<ul id="user-data">
<li class="name"><?php echo osc_user_name(); ?></li>
<?php if( osc_user_website() != '' ) { ?>
<li class="website"><a href="<?php echo osc_user_website(); ?>"><?php echo
    osc_user_website(); ?></a></li>
<?php } ?>
<?php if( $address != '' ) { ?>
<li class="address"><?php printf(__( '<strong>Address:</strong> %1$s', $address);
    ?></li>
<?php } ?>
<?php if( $location != '' ) { ?>
<li class="location"><?php printf(__( '<strong>Location:</strong> %1$s', $location
    ); ?></li>
<?php } ?>
</ul>
</div>
<?php if( osc_user_info() != '' ) { ?>
<h2><?php _e('User description', 'bender'); ?></h2>
<?php } ?>
<?php echo osc_user_info(); ?>
<?php if( osc_count_items() > 0 ) { ?>
<div class="similar-ads">
<h2><?php _e('Latest listings', 'bender'); ?></h2>
<?php osc_current_web_theme_path('loop.php'); ?>
<div class="paginate"><?php echo osc_pagination_items(); ?></div>
<div class="clear"></div>
</div>
<?php } ?>
</div>
<?php osc_current_web_theme_path('footer.php'); ?>

<?php
/*
 *      Oclass      software for creating and publishing online classified
 *                  advertising platforms
 *
 *                  Copyright (C) 2013 OSCLASS
 *
 *      This program is free software: you can redistribute it and/or
 *      modify it under the terms of the GNU Affero General Public License
 *      as published by the Free Software Foundation, either version 3 of
 *      the License, or (at your option) any later version.
 *
 *      This program is distributed in the hope that it will be useful, but
 *      WITHOUT ANY WARRANTY; without even the implied warranty of
 *      MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 *      GNU Affero General Public License for more details.
 *
 *      You should have received a copy of the GNU Affero General Public
 *      License along with this program. If not, see <http://www.gnu.org/licenses/>.
 */
?>
<div id="sidebar">
<?php if(osc_logged_user_id() != osc_user_id()) { ?>
<?php if(osc_reg_user_can_contact() && osc_is_web_user_logged_in() || !
    osc_reg_user_can_contact() ) { ?>
<div id="contact" class="widget-box form-container form-vertical">

```

```

<h2><?php _e(" Contact", 'bender '); ?></h2>
<ul id="error_list"></ul>
<form action="<?php echo osc_base_url(true); ?>" method="post" name="
    contact_form" id="contact_form">
    <input type="hidden" name="action" value="contact_post" />
    <input type="hidden" name="page" value="user" />
    <input type="hidden" name="id" value="<?php echo osc_user_id();?>" />
    <div class="control-group">
        <label class="control-label" for="yourName"><?php _e('Your name', '
            bender '); ?></label>
        <div class="controls"><?php ContactForm::your_name(); ?></div>
    </div>
    <div class="control-group">
        <label class="control-label" for="yourEmail"><?php _e('Your email
            address', 'bender '); ?></label>
        <div class="controls"><?php ContactForm::your_email(); ?></div>
    </div>
    <div class="control-group">
        <label class="control-label" for="phoneNumber"><?php _e('Phone number
            ', 'bender '); ?> (<?php _e('optional', 'bender '); ?>)</label>
        <div class="controls"><?php ContactForm::your_phone_number(); ?></div>
    </div>

    <div class="control-group">
        <label class="control-label" for="message"><?php _e('Message', 'bender
            '); ?></label>
        <div class="controls textarea"><?php ContactForm::your_message(); ?></
            div>
    </div>

    <div class="control-group">
        <div class="controls">
            <?php osc_run_hook('item_contact_form', osc_item_id()); ?>
            <?php if( osc_recaptcha_public_key() ) { ?>
            <script type="text/javascript">
                var RecaptchaOptions = {
                    theme : 'custom',
                    custom_theme_widget : 'recaptcha_widget'
                };
            </script>
            <style type="text/css"> div#recaptcha_widget, div#recaptcha_image
                > img { width:280px; } </style>
            <div id="recaptcha_widget">
                <div id="recaptcha_image"><img /></div>
                <span class="recaptcha_only_if_image"><?php _e('Enter the
                    words above', 'bender '); ?></span>
                <input type="text" id="recaptcha_response_field" name="
                    recaptcha_response_field" />
                <div><a href="javascript:Recaptcha.showhelp()"><?php _e('Help
                    ', 'bender '); ?></a></div>
            </div>
            <?php } ?>
            <?php osc_show_recaptcha(); ?>
            <button type="submit" class="ui-button ui-button-middle ui-button-
                main"><?php _e(" Send", 'bender '); ?></button>
        </div>
    </div>
    <?php ContactForm::js_validation(); ?>
</form>
</div>
<?php
}
?>
</div><!-- /sidebar -->

<?php
/*
 *      Oclass      software for creating and publishing online classified
 *                  advertising platforms
 *
 *                  Copyright (C) 2013 OSCLASS
 *
 *      This program is free software: you can redistribute it and/or
 *      modify it under the terms of the GNU Affero General Public License
 *      as published by the Free Software Foundation, either version 3 of
 *      the License, or (at your option) any later version.
 *
 *      This program is distributed in the hope that it will be useful, but
 *      WITHOUT ANY WARRANTY; without even the implied warranty of
 *      MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 *      GNU Affero General Public License for more details.
 *
 *      You should have received a copy of the GNU Affero General Public
 *      License along with this program. If not, see <http://www.gnu.org/licenses/>.
 */

// meta tag robots
osc_add_hook('header', 'bender_nofollow_construct');

bender_add_body_class('recover');
osc_current_web_theme_path('header.php');
?>
<div class="form-container form-horizontal form-container-box">
<div class="header">
    <h1><?php _e('Recover your password', 'bender '); ?></h1>
</div>
<div class="resp-wrapper">

```

```

<form action="<?php echo osc_base_url(true); ?>" method="post" >
<input type="hidden" name="page" value="login" />
<input type="hidden" name="action" value="recover_post" />
<div class="control-group">
  <label class="control-label" for="email"><?php _e('E-mail', 'bender'); ?></label>
  <div class="controls">
    <?php UserForm::email_text(); ?>
    <?php osc_show_recaptcha('recover_password'); ?>
  </div>
</div>
<div class="control-group">
  <div class="controls">
    <button type="submit" class="ui-button ui-button-middle ui-button-main"><?
      php _e("Send me a new password", 'bender'); ?></button>
  </div>
</div>
</form>
</div>
<?php osc_current_web_theme_path('footer.php') ; ?>

<?php
/*
 *      Oclass      software for creating and publishing online classified
 *                  advertising platforms
 *
 *                  Copyright (C) 2013 OSCLASS
 *
 *      This program is free software: you can redistribute it and/or
 *      modify it under the terms of the GNU Affero General Public License
 *      as published by the Free Software Foundation, either version 3 of
 *      the License, or (at your option) any later version.
 *
 *      This program is distributed in the hope that it will be useful, but
 *      WITHOUT ANY WARRANTY; without even the implied warranty of
 *      MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 *      GNU Affero General Public License for more details.
 *
 *      You should have received a copy of the GNU Affero General Public
 *      License along with this program. If not, see <http://www.gnu.org/licenses/>.
 */

// meta tag robots
osc_add_hook('header', 'bender_nofollow_construct');

bender_add_body_class('register');
osc_enqueue_script('jquery-validate');
osc_current_web_theme_path('header.php');
?>
<div class="form-container form-horizontal form-container-box">
  <div class="header">
    <h1><?php _e('Register an account for free', 'bender'); ?></h1>
  </div>
  <div class="resp-wrapper">
    <form name="register" action="<?php echo osc_base_url(true); ?>" method="post" >
      <input type="hidden" name="page" value="register" />
      <input type="hidden" name="action" value="register_post" />
      <ul id="error_list"></ul>
      <div class="control-group">
        <label class="control-label" for="name"><?php _e('Name', 'bender'); ?></label>
        <div class="controls">
          <?php UserForm::name_text(); ?>
        </div>
      </div>
      <div class="control-group">
        <label class="control-label" for="email"><?php _e('E-mail', 'bender'); ?></
          label>
        <div class="controls">
          <?php UserForm::email_text(); ?>
          <?php osc_show_recaptcha('register'); ?>
        </div>
      </div>
      <div class="control-group">
        <label class="control-label" for="password"><?php _e('Password', 'bender'); ?>
        <div class="controls">
          <?php UserForm::password_text(); ?>
        </div>
      </div>
      <div class="control-group">
        <label class="control-label" for="password-2"><?php _e('Repeat password', '
          bender'); ?></label>
        <div class="controls">
          <?php UserForm::check_password_text(); ?>
          <p id="password-error" style="display:none;">
            <?php _e("Passwords don't match", 'bender'); ?>
          </p>
        </div>
      </div>
      <?php osc_run_hook('user_register_form'); ?>
      <div class="control-group">
        <div class="controls">
          <button type="submit" class="ui-button ui-button-middle ui-button-main"><?
            php _e("Create", 'bender'); ?></button>
        </div>
      </div>
    </form>
  </div>
</div>
</div>

```

```

<?php UserForm::js_validation(); ?>
<?php osc_current_web_theme_path('footer.php') ; ?>

<?php
/*
 *      Oclass      software for creating and publishing online classified
 *                  advertising platforms
 *
 *                  Copyright (C) 2013 OSCLASS
 *
 *      This program is free software: you can redistribute it and/or
 *      modify it under the terms of the GNU Affero General Public License
 *      as published by the Free Software Foundation, either version 3 of
 *      the License, or (at your option) any later version.
 *
 *      This program is distributed in the hope that it will be useful, but
 *      WITHOUT ANY WARRANTY; without even the implied warranty of
 *      MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 *      GNU Affero General Public License for more details.
 *
 *      You should have received a copy of the GNU Affero General Public
 *      License along with this program. If not, see <http://www.gnu.org/licenses/>.
 */
?>
<div class="actions">
  <a href="#" data-bclass-toggle="display-filters" class="resp-toogle show-filters-btn"><?php
    _e('Display menu', 'bender'); ?></a>
</div>
<div id="sidebar">
  <?php echo osc_private_user_menu( get_user_menu() ); ?>
</div>
<div id="dialog-delete-account" title="<?php echo osc_esc_html(__( 'Delete account', 'bender' ) )
  ; ?>">
  <?php _e('Are you sure you want to delete your account?', 'bender'); ?>
</div>

<?php

/**
 * Receive and save votes from frontend.
 */

$votedUserId = (Params::getParam("userId") == '') ? null : Params::getParam("userId");
$itemId      = (Params::getParam("itemId") == '') ? null : Params::getParam("itemId");
$iVote       = (Params::getParam("vote") == '') ? null : Params::getParam("vote");

$userId = osc_logged_user_id();
$hash  = '';

// Vote on Users (not implemented)
if( isset($iVote) && is_numeric($iVote) && isset($votedUserId) && is_numeric($votedUserId) )
{
  if( $iVote<=5 && $iVote>=1){
    if( can_vote_user($votedUserId, $userId) ) {
      AR::newInstance()->insertUserVote($votedUserId, $userId, $iVote);
    }
  }
  // return updated voting
  $aux_vote = AR::newInstance()->getUserAvgRating($votedUserId);
  $aux_count = AR::newInstance()->getUserNumberOfVotes($votedUserId);
  $vote['vote'] = $aux_vote['vote'];
  $vote['total'] = $aux_count['total'];
  $vote['userId'] = $votedUserId;

  $vote['can_vote'] = true;
  if(!osc_is_web_user_logged_in() || !can_vote_user($votedUserId, $userId) ) {
    $vote['can_vote'] = false;
  }
}

require 'view_votes_user.php';

// Vote/Ratings on Items

if( isset($iVote) && is_numeric($iVote) && isset($itemId) && is_numeric($itemId) )
{
  if( $iVote<=5 && $iVote>=1){ //if vote's value is 1-5, it is valid
    if( $userId == 0 ) {
      $userId = 'NULL';
      $hash = $_SERVER['HTTP_USER_AGENT'].$_SERVER['REMOTE_ADDR'];
      $hash = sha1($hash);
    } else {
      $hash = null;
    }
  }

  $open = osc_get_preference('ads_rating', 'open');
  $user = osc_get_preference('user', 'voting'); //current logged on user

  if($open == 1) {
    if( can_vote($itemId, $userId, $hash) ) {
      AR::newInstance()->insertItemVote($itemId, $userId, $iVote, $hash);
    }
  }
  } else if($user == 1 && is_null($hash) ) {
  if( can_vote($itemId, $userId, $hash) ) {
    AR::newInstance()->insertItemVote($itemId, $userId, $iVote, $hash);
  }
}
}

```

```

        //updating dev table after a rating is successfully received
//AR::newInstance()->update_dev_table($userId, $itemId);

//AR::newInstance()->.initSlopeOneTable(); //compute the slope ones after receiving a
        rating

    }
// updates votes and stars
// on item page
$item = Item::newInstance()->findByPrimaryKey($itemId);
View::newInstance()->.exportVariableToView('item', $item);
if (osc_is_this_category(osc_get_preference('ads-rating', 'plugin_name'),
    osc_item_category_id())) {
    $aux_vote = AR::newInstance()->getItemAvgRating(osc_item_id()); //???
    $aux_count = AR::newInstance()->getItemNumberOfVotes(osc_item_id());
    $vote['vote'] = $aux_vote['vote'];
    $vote['total'] = $aux_count['total'];

    $vote['can_vote'] = true;
    if(osc_get_preference('user', 'voting') == 1) {
        if(!osc_is_web_user_logged_in()) {
            $vote['can_vote'] = false;
        }
    }
    if(!can_vote(osc_item_id(), osc_logged_user_id(), $hash)){
        $vote['can_vote'] = false;
    }
}

require 'view_votes.php';
}
}
?>

<?php

/**
 * Model database for Voting tables
 *
 * @package OSCClass
 * @subpackage Model
 * @since 3.2
 */
class AR extends DAO
{
    /**
     * It references to self object: AR.
     * It is used as a singleton
     *
     * @access private
     * @since 3.2
     * @var AR
     */
    private static $instance ;

    /**
     * It creates a new ad rating object class ir if it has been created
     * before, it return the previous object
     *
     * @access public
     * @since 3.2
     * @return AR
     */
    public static function newInstance()
    {
        if( !self::$instance instanceof self ) {
            self::$instance = new self ;
        }
        return self::$instance ;
    }

    /**
     * Construct
     */
    function __construct()
    {
        parent::__construct();
    }

    /**
     * Return table name voting item
     * @return string
     */
    public function getTable_Item()
    {
        return DB_TABLE_PREFIX.'t_rated_items';
    }

    /**
     * Return table name dev
     * @return string
     */
    public function getTable_dev()
    {
        return DB_TABLE_PREFIX.'t_dev';
    }
}

/**

```

```

* Return table name voting user
* @return string
*/
// public function getTable_User()
// {
//     return DB_TABLE_PREFIX.'t_voting_user';
// }

/**
* Import sql file
* @param type $file
*/
public function import($file)
{
    $path = osc_plugin_resource($file) ;
    $sql = file_get_contents($path);

    if(! $this->dao->importSQL($sql) ){
        throw new Exception( "Error importSQL::AR<br>". $file ) ;
    }
}

/**
* Remove data and tables related to the plugin.
*/
public function uninstall()
{
    $this->dao->query("DELETE FROM ".DB_TABLE_PREFIX."t_plugin_category WHERE s_plugin_name
        = 'ads_rating' ");
    $this->dao->query("DROP TABLE ". $this->getTable_Item());
    $this->dao->query("DROP TABLE ". $this->getTable_dev());
}

// update
public function _update($table, $values, $where)
{
    $this->dao->from($table) ;
    $this->dao->set($values) ;
    $this->dao->where($where) ;
    return $this->dao->update() ;
}

// item related -----
/**
* Insert Item rating
*
* @param type $itemId
* @param type $userId
* @param type $iVote
* @param type $hash
* @return type
*/
function insertItemVote($itemId, $userId, $iVote, $hash)
{
    $aSet = array(
        'fk_i_item_id' => (int)$itemId,
        'i_value' => (int)$iVote,
        's_hash' => is_null($hash) ? "" : "$hash"
    );

    if($userId != 'NULL' && is_numeric($userId) ) {
        $aSet['fk_i_user_id'] = $userId;
    }

    return $this->dao->insert($this->getTable_Item(), $aSet);
}

/**
* Return an average of ratings given an item id
*
* @param type $id
* @return type
*/
function getItemAvgRating($id) {
    if(is_numeric($id)) {

        $this->dao->select('format(avg(i_value),1) as vote');
        $this->dao->from($this->getTable_Item());
        $this->dao->where('fk_i_item_id', (int)$id );

        $result = $this->dao->get();

        if(! $result ) {
            return array() ;
        }

        return $result->row();
    } else {
        return array('vote' => 0);
    }
}
}

```

```

/**
 * Return the number of votes given an item id
 *
 * @param type $id
 * @return type
 */
function getItemNumberOfVotes($id) {

    if(is_numeric($id)) {

        $this->dao->select('count(*) as total');
        $this->dao->from( $this->getTable_Item());
        $this->dao->where('fk_i_item_id', (int)$id );

        $result = $this->dao->get();

        if(! $result ) {
            return array() ;
        }

        return $result->row();

    } else {

        return array('total' => 0);

    }

}

/**
 * Return rating given an item id and hash
 *
 * @param type $itemId
 * @param type $hash
 * @return type
 */
function getItemIsRated($itemId, $hash, $userId = null)
{

    if( is_numeric($itemId) && ( $userId == null || is_numeric($userId) ) ) {

        $this->dao->select('i_value');
        $this->dao->from( $this->getTable_Item());
        $this->dao->where('fk_i_item_id', (int)$itemId );

        if( $userId == null ) {
            $this->dao->where('fk_i_user_id IS NULL');
        } else {
            $this->dao->where('fk_i_user_id', (int)$userId);
        }

        $this->dao->where('s_hash', (string)$hash );

        $result = $this->dao->get();

        if(! $result) {
            return array() ;
        }

        return $result->row();

    } else {

        return array();

    }

}

/**
 * Return an array of items ordered by avg_votes
 *
 * @param type $category_id
 * @param type $order
 * @param type $num
 * @return type
 */
function getItemRatings($category_id = null, $order = 'desc', $num = 5)
{

    $sql = 'SELECT fk_i_item_id as item_id, format(avg(i_value),1) as avg_vote, count(*) as
num_votes, '.DB_TABLE_PREFIX.'.t_item.fk_i_category_id as category_id ';

    if(!is_null($category_id)) {
        $sql .= ', '.DB_TABLE_PREFIX.'.t_category.fk_i_parent_id as parent_category_id ';
    }

    $sql .= 'FROM '.DB_TABLE_PREFIX.'.t_rated_items ';
    $sql .= 'LEFT JOIN '.DB_TABLE_PREFIX.'.t_item ON '.DB_TABLE_PREFIX.'.t_item.pk_i_id = '.
DB_TABLE_PREFIX.'.t_rated_items.fk_i_item_id ';
    $sql .= 'LEFT JOIN '.DB_TABLE_PREFIX.'.t_category ON '.DB_TABLE_PREFIX.'.t_category.
pk_i_id = '.DB_TABLE_PREFIX.'.t_item.fk_i_category_id ';

    if(!is_null($category_id)) {
        $sql .= 'WHERE '.DB_TABLE_PREFIX.'.t_item.fk_i_category_id = '.$category_id.' ';
        $sql .= 'OR '.DB_TABLE_PREFIX.'.t_category.fk_i_parent_id = '.$category_id.' ';
        $sql .= ' AND ';
    } else {
        $sql .= 'WHERE ';
    }
}

```



```

    }

    $sql .= ' '.DB_TABLE_PREFIX.'t_item.b_active = 1 ';
    $sql .= 'AND '.DB_TABLE_PREFIX.'t_item.b_enabled = 1 ';
    $sql .= 'AND '.DB_TABLE_PREFIX.'t_item.b_spam = 0 ';
    $sql .= 'AND ( '.DB_TABLE_PREFIX.'t_item.b_premium = 1 || '.DB_TABLE_PREFIX.'t_category.
        i_expiration_days = 0 ||DATEDIFF('\'.date('Y-m-d H:i:s')'\','.DB_TABLE_PREFIX.'
        t_item.dt_pub_date) < '.DB_TABLE_PREFIX.'t_category.i_expiration_days) ';
    $sql .= 'AND '.DB_TABLE_PREFIX.'t_category.b_enabled = 1 ';
    $sql .= 'GROUP BY item_id ORDER BY avg_vote '.$order.', num_votes '.$order.' LIMIT 0, '.
        $num;

    $result = $this->dao->query($sql);
    if( !$result ) {
        return array() ;
    }

    return $result->result();
}

/**
 * Get recommended ads per user based on his rated items.
 *
 * @param type $userId
 * @param type $limit
 * @return type
 */
public function getPredictBest($userId, $limit = 5)
{
    $sql = 'SELECT d.i.item_id_1 as item, sum(d.i.sum + d.i.count * r.i.value)/sum(d.i.count)
        as avgrat'
        .' FROM '. DB_TABLE_PREFIX . 't_dev d, ' . DB_TABLE_PREFIX . 't_rated_items r'
        .' WHERE r.fk_i_user_id = ' . $userId
        .' AND d.i.item_id_1 NOT IN'
        .' (SELECT fk_i_item_id from ' . DB_TABLE_PREFIX . 't_rated_items WHERE fk_i_user_id
        = ' . $userId . ') '
        .' AND d.i.item_id_1 NOT IN'
        .' (SELECT pk_i_id FROM ' . DB_TABLE_PREFIX . 't_item WHERE fk_i_user_id = ' .
        $userId . ') '
        .' AND d.i.item_id_2 = r.fk_i_item_id '
        .' GROUP BY d.i.item_id_1 '
        .' ORDER BY avgrat DESC limit '
        .' $limit;

    $result = $this->dao->query($sql);

    if(!$result) {
        return array();
    }

    return $result -> result();
}

//returns id of primary image of an ad
public function getThumbnailId($itemId) {

    $sql = 'SELECT pk_i_id AS thumb_id'
        .' FROM '. DB_TABLE_PREFIX . 't_item_resource '
        .' WHERE fk_i_item_id ='. $itemId
        .' LIMIT 1';

    $result = $this->dao->query($sql);

    if(!$result) {
        return array();
    }

    return $result -> row();
}

//returns id of primary image of an ad
public function getThumbnailExt($itemId, $thumbId) {

    $sql = 'SELECT s_extension AS thumb_ext'
        .' FROM '. DB_TABLE_PREFIX . 't_item_resource '
        .' WHERE fk_i_item_id ='. $itemId
        .' AND pk_i_id ='. $thumbId
        .' LIMIT 1';

    $result = $this->dao->query($sql);

    if(!$result) {
        return array();
    }

    return $result -> row();
}

/**
 * Computes slope ones for every rating pair
 * and updates the t_dev table
 *
 */
function _initSlopeOneTable()
{

```

```

/**
 * Delete all the data of t_dev table
 */

$this->dao->query('TRUNCATE TABLE ' . DB_TABLE_PREFIX . 't_dev');
/**
 * Get distinct item_id
 */

$sql_count = "SELECT COUNT(DISTINCT fk_i_item_id) as total FROM " . DB_TABLE_PREFIX . "
    t_rated_items";
$res = $this->dao->query($sql_count);

$rescount = $res->row();
//print $rescount['total'];
$count = $rescount['total'];

$sql = 'SELECT DISTINCT fk_i_item_id FROM ' . DB_TABLE_PREFIX . 't_rated_items';
$rs = $this->dao->query($sql);

for($i=0; $i<$count; $i++) {
    $rsValue = $rs->row($i);
    $itemId = $rsValue['fk_i_item_id'];

    $slopeOneSql = 'INSERT INTO ' . DB_TABLE_PREFIX . 't_dev ('
        . 'SELECT a.fk_i_item_id as i_item_id_1, b.fk_i_item_id as i_item_id_2, count(*) as
            i_count, sum(a.i_value-b.i_value) as i_sum '
        . 'FROM ' . DB_TABLE_PREFIX . 't_rated_items a, ' . DB_TABLE_PREFIX . 't_rated_items b '
        . 'WHERE a.fk_i_item_id = ' . $itemId . ' AND b.fk_i_item_id != a.fk_i_item_id AND a.
            fk_i_user_id = b.fk_i_user_id '
        . 'GROUP BY a.fk_i_item_id,b.fk_i_item_id)';

    $this->dao->query($slopeOneSql);
}
}

/*
function update_dev_table($userId, $itemId) {
    $sql = "SELECT DISTINCT r.fk_i_item_id, (r2.i_value - r.i_value) as rating_difference";
    $sql .= " FROM " . DB_TABLE_PREFIX . "t_rated_items r, " . DB_TABLE_PREFIX . "
        t_rated_items r2";
    $sql .= " WHERE r.fk_i_user_id = " . $userId . " AND r2.fk_i_item_id = " . $itemId . " AND
        r2.fk_i_user_id = " . $userId;
    $result = $this->dao->query($sql);

    if(! $result ) {
        return array() ;
    }

    //For every one of the user s rating pairs ,
    //update the dev table
    foreach ($result->result() as $row) {

        $other_item_id = $row["fk_i_item_id"];
        $rating_difference = $row["rating_difference"];
        $this->dao->select("i_item_id_1");
        $this->dao->from($this->getTable_dev());
        $this->dao->where("i_item_id_1 = " . $itemId . " AND i_item_id_2 = " . $other_item_id );
        $r = $this->dao->get();

        if ( $r ) {

            $update_array = array('i_count' => 'i_count' + 1, 'i_sum' => 'i_sum'+
                $rating_difference);
            $this->update($this->getTable_dev(), $update_array, array('i_item_id_1' => $itemId, '
                i_item_id_2' => $other_item_id));

            if ( $itemId != $other_item_id ) {

                $this->update($this->getTable_dev(), array('i_count' => 'i_count' + 1, 'i_sum' => '
                    i_sum' - $rating_difference), array('i_item_id_1' => $other_item_id, '
                    i_item_id_2' => $itemId));

            }

        }

    }

    else {

        $aSet = array(

            'i_item_id_1' => $itemId != $other_item_id ? (int)$other_item_id : (int)$itemId,
            'i_item_id_2' => $itemId != $other_item_id ? (int)$itemId : (int)$other_item_id,
            'i_count' => 1,
            'i_sum' => $itemId != $other_item_id ? $rating_difference : - $rating_difference
        );
        $this->dao->insert($this->getTable_dev(), $aSet);

    }

}

}*/

/**
 * Delete table entries related to an item id

```

```

*
* @param type $itemId
* @return type
*/
function deleteItem($itemId)
{
    if(is_numeric($itemId)) {
        return $this->dao->delete($this->getTableItem(), 'fk_i_item_id = '.$itemId);
    }
    return false;
}
}
?>

<?php if (!defined('OC_ADMIN') || OC_ADMIN!==true) exit('Access is not allowed.');
```

```

/*
 * OSClass      software for creating and publishing online classified
 *              advertising platforms
 *
 *              Copyright (C) 2010 OSCLASS
 *
 *              This program is free software: you can redistribute it and/or
 *              modify it under the terms of the GNU Affero General Public License
 *              as published by the Free Software Foundation, either version 3 of
 *              the License, or (at your option) any later version.
 *
 *              This program is distributed in the hope that it will be useful, but
 *              WITHOUT ANY WARRANTY; without even the implied warranty of
 *              MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 *              GNU Affero General Public License for more details.
 *
 *              You should have received a copy of the GNU Affero General Public
 *              License along with this program. If not, see <http://www.gnu.org/licenses/>.
 */

if( Params::getParam('plugin_action') == 'post' ) {

    /**
     * Save Item form
     */
    $type_vote      = Params::getParam('type_vote');
    $enable_item    = Params::getParam('enable_item');
    if($enable_item == 'on') {
        osc_set_preference('item_voting', '1', 'voting', 'BOOLEAN');
        if($type_vote == 'user') {
            osc_set_preference('open', '0', 'voting', 'BOOLEAN');
            osc_set_preference('user', '1', 'voting', 'BOOLEAN');
        } else if($type_vote == 'open') {
            osc_set_preference('open', '1', 'voting', 'BOOLEAN');
            osc_set_preference('user', '0', 'voting', 'BOOLEAN');
        }
    } else {
        osc_set_preference('item_voting', '0', 'voting', 'BOOLEAN');
    }

    /**
     * Save User form
     */
    $enable_user    = Params::getParam('enable_user');
    if($enable_user == 'on') {
        osc_set_preference('user_voting', '1', 'voting', 'BOOLEAN');
    } else {
        osc_set_preference('user_voting', '0', 'voting', 'BOOLEAN');
    }
    echo '<div style="text-align:center; font-size:22px; background-color:#00bb00;"><p>' .
        _('Congratulations. The plugin is now configured', 'voting') . '</p></div>';
    osc_reset_preferences();
}
?>

<div id="settings-form" style="padding-left: 15px; padding-right: 15px;">
<div style="padding: 20px;">
<div style="float: left; width: 100%;">
<b style="font-size: 1.5em;"><?php _e('Items', 'voting');?></b>
<form action="<?php echo osc_admin_base_url(true); ?>?page=plugins&action=
renderplugin&file="<?php echo osc_plugin_folder(__FILE__).'conf.php'; ?>
method="POST">
<input type="hidden" name="plugin_action" value="post" />
<p><label for="enable_item"><?php _e('Enable for items', 'voting') ; ?></label>
<input type="checkbox" name="enable_item" <?php if(osc_get_preference('
item_voting', 'voting')) echo 'checked="checked"';?>></p>
<div style="width: 100%;">
<p>
<input <?php if(!osc_get_preference('item_voting', 'voting')){ echo '
disabled=""'; }?> name="type_vote" value="open" id="open" type="
radio" <?php if(osc_get_preference('open', 'voting')) echo '
checked="checked"';?>/>
<label for="open" ><?php _e('Open voting', 'voting'); ?> (<?php _e('
All can vote on the items', 'voting'); ?>)</label>
</p>
<p>
<input <?php if(!osc_get_preference('item_voting', 'voting')){ echo '
disabled=""'; }?>name="type_vote" value="user" id="user" type="
radio" <?php if(osc_get_preference('user', 'voting')) echo '
checked="checked"';?>/>
<label for="user"><?php _e('Users voting', 'voting'); ?> (<?php _e("
Only registered users can vote", 'voting'); ?>)</label>
</p>
</div>
</div>
</div>

```

```

        </div>
    </div>

    <div style="float: left; width: 100%;">
        <b style="font-size: 1.5em;"><?php _e('Users', 'voting');?></b>
        <p><label for="enable-user"><?php _e('Enable for users', 'voting') ; ?></label>
            <input type="checkbox" name="enable-user" <?php if(osc_get_preference('
                user_voting', 'voting')) echo 'checked="checked";?>/></p>
            <input type="submit" value="<?php _e('Save', 'voting'); ?>" />
        </form>
    </div>
</div style="clear: both;"></div>
</div>
</div>

<?php if (!defined('OC_ADMIN') || OC_ADMIN!==true) exit('Access is not allowed. '); ?>
<style>
    p.code {
        padding: 8px;
        background-color: #F3F3F3;
        border: 1px solid #DDD;
    }
    p.code span{
        display: block;
    }
</style>
<div id="settings_form" style="padding-left: 15px; padding-right: 15px;">
<h1>Help</h1>
<h2><?php _e('Plugin information', 'voting') ; ?></h2>
<p>
    <?php _e('This plugin adds a rating system and allows users to vote among them the
        quality of the item and quality sellers', 'voting') ; ?>.
</p>
<p>
    <ul>
        <li><?php _e('Easy plugin configuration.', 'voting') ; ?></li>
        <li><?php _e('Vote items, can be enabled and disabled. Can be configured what kind
            of users can vote items, registered users only or guest too', 'voting') ;
            ?></li>
        <li><?php _e('Vote users, can be enabled and disabled. Only registered users can
            vote sellers', 'voting') ; ?></li>
        <li><?php _e('Allows to show the best rated items or users at frontend, adding
            some extra code at your template', 'voting') ; ?></li>
    </ul>
</p>
<hr>
<h2><?php _e('Show best rated', 'voting') ;?></h2>
<p><?php _e('You can display the best rated list, where you want', 'voting') ; ?></p>
<p><?php _e('Adding this line of code, you can show the items or users at main web page,
    into sidebar', 'voting') ; ?></p>
<p><b><?php _e('Items', 'voting');?></b></p>
<p class="code">
    <?php echo htmlentities('<?php'); ?> echo_best_rated(NUMBER_OF_ITEMS); ?><br>
</p>
<em><?php _e('Edit main.php (located under root theme folder)', 'voting') ; ?></em>
<p class="code">
    <span style="padding-left: 10px;"><?php echo htmlentities('<div id="sidebar">'); ?></span>
    <span style="padding-left: 30px;"><?php echo htmlentities('<div class="navigation">');
        ?></span>
    <span style="padding-left: 50px;"><?php echo htmlentities('<?php'); ?> echo_best_rated
        (3); ?></span>
    <span style="padding-left: 50px;">...</span>
    <span style="padding-left: 30px;"><?php echo htmlentities('</div>'); ?></span>
    <span style="padding-left: 10px;"><?php echo htmlentities('</div>'); ?></span>
</p>
<p><b><?php _e('Users', 'voting');?></b></p>
<p class="code"><span><?php echo htmlentities('<?php'); ?> echo_users_best_rated(
    NUMBER_OF_USERS); ?></span></p>
<em><?php _e('Edit main.php (located under root theme folder)', 'voting') ; ?></em>
<p class="code">
    <span style="padding-left: 10px;"><?php echo htmlentities('<div id="sidebar">'); ?></span>
    <span style="padding-left: 30px;"><?php echo htmlentities('<div class="navigation">');
        ?></span>
    <span style="padding-left: 50px;"><?php echo htmlentities('<?php'); ?>
        echo_users_best_rated(3); ?></span>
    <span style="padding-left: 50px;">...</span>
    <span style="padding-left: 30px;"><?php echo htmlentities('</div>'); ?></span>
    <span style="padding-left: 10px;"><?php echo htmlentities('</div>'); ?></span>
</p>
<hr/>
<h2><?php _e('Change place of "voting plugin"', 'voting') ; ?></h2>
<p><b><?php _e('Vote section', 'voting');?></b> <?php _e('by default is added below item
    description, but if you need place into another place follow this instructions:', '
    voting') ; ?></p>
<p><b><?php _e('Items', 'voting') ; ?></b></p>
<p><?php _e('You need add this line at end of functions.php (located under root theme
    folder) :', 'voting') ; ?><br></p>
<p class="code"><span><?php echo htmlentities('<?php'); ?> osc_remove_hook('item_detail',
    'voting.item_detail'); ?></span></p>

```

```

<p><?php _e('Now you can call function ', 'voting'); ?> <span class="code">
    voting_item_detail()</span> <?php _e('directly where you want <b>into item page</b>',
    'voting');?>.</p>

<p class="code"><span><?php echo htmlentities('<?php'); ?> voting_item_detail(); ?></span
></p>

<p><b><?php _e('Users ', 'voting') ; ?></b></p>

<p><?php _e('You need add this line at end of functions.php (located under root theme
    folder) :', 'voting') ; ?><br></p>
<p class="code"><span><?php echo htmlentities('<?php'); ?> osc_remove_hook('item_detail',
    'voting_item_detail_user'); ?></span></p>

<p><?php _e('Now you can call function ', 'voting'); ?> <p>voting_item_detail_user()</code>
    <?php _e('directly where you want <b>into item page</b>', 'voting');?>.</p>

<p class="code"><span><?php echo htmlentities('<?php'); ?> voting_item_detail_user(); ?></
span></p>

<hr/>

<h2><?php _e('Show user ratings on User Dashboard ', 'voting') ; ?></h2>
<p><?php _e('You can show user ratings on User Dashboard if you pass as argument the user
    id ', 'voting');?> (user-dashboard.php)</p>
<p class="code"><span><?php echo htmlentities('<?php'); ?> voting_item_detail_user(
    osc_logged_user_id() ); ?></p>

<hr/>

<h2><?php _e('Show user ratings on User public profile ', 'voting') ; ?></h2>
<p><?php _e('You can show user ratings on User public profile if you pass as argument the
    user id ', 'voting');?> (user-public-profile.php)</p>
<p class="code"><span><?php echo htmlentities('<?php'); ?> voting_item_detail_user( osc_user_id
    () ); ?></p>

<hr/>

<p><b><?php _e('If you want modify the templates, you can find them in the plugin folder
    ', 'voting');?>.</b></p>
<p><b>view_votes.php</b> <?php _e('Template rating items ', 'voting') ; ?></p>
<p><b>view_votes_user.php</b> <?php _e('Template rating users ', 'voting') ; ?></p>
<br>
<p><b>set_results.php</b> <?php _e('Template best rated items ', 'voting') ; ?></p>
<p><b>set_results_user.php</b> <?php _e('Template best rated users ', 'voting') ; ?></p>
</div>

<?php
/*
Plugin Name: Ads Rating
Description: Allows the rating of each ads and showing recommended ads for the user.
Version: 1.0.0
Short Name: ads rating
*/

require_once 'AR_Model.php' ;

/* Initialize plugin
-----*/

// This is to add the admin link for this page.
osc_add_hook("init", "_init_plugin");

// Initialize plugin informations and other necessary objects.
function _init_plugin() {

    osc_set_preference('ads_rating', 'Ads Rating', 'plugin_name');

    osc_enqueue_style('main-css', osc_base_url() . 'oc-content/plugins/ads_rating/css/main.css')
    ;
}

/* end initialize plugin */

/* Install Ads Rating plugin.
-----*/

// This is needed in order to be able to activate the plugin
osc_register_plugin(osc_plugin_path(__FILE__), '_install');

// Set plugin preferences
function _install() {

    AR::newInstance()->import('ads_rating/struct.sql');
    osc_set_preference('ads_rating', '1', 'rating', 'BOOLEAN');
    osc_set_preference('ads_rating', '1', 'open', 'BOOLEAN');

}

/* end Install Ads Rating plugin. */

/* Uninstall Ads Rating plugin.
-----*/

```

```

// This is a hack to show a Uninstall link at plugins table (you could also use some other
hook to show a custom option panel)
osc_add_hook(osc_plugin_path(__FILE__)." _uninstall", '_uninstall');

// Delete plugin preferences
function _uninstall() {
    AR::newInstance()->uninstall();
    osc_delete_preference('ads-rating', 'rating');
    osc_delete_preference('ads-rating', 'open');
}

/* end Uninstall Ads Rating plugin. */

/* Admin Configuration
-----*/

// This is a hack to show a Configure link at plugins table (you could also use some other
hook to show a custom option panel)
osc_add_hook(osc_plugin_path(__FILE__)." _configure", '_admin-configuration');

function _admin-configuration() {
    // Standard configuration page for plugin which extend item's attributes
    osc_plugin_configure_view(osc_plugin_path(__FILE__));
}

/* end Admin Configuration */

/* Adding plugin menu
-----*/

// This is to add the admin link for this page.
osc_add_hook("init-admin", "add-menu");

function add-menu() {
    if(osc_version() >= 300) {
        osc_add_admin-menu-page("Ads Rating Options" , osc_admin-render-plugin_url(
            osc_plugin_folder(__FILE__) . 'conf.php'), 'ads-rating-plugin', 'administrator' );
        // osc_add_admin_submenu-page('ads-rating-plugin', --('Settings', 'voting'),
            osc_admin-render-plugin_url(osc_plugin_folder(__FILE__) . 'conf.php'), '
            voting-plugin-settings', 'administrator');
        // osc_add_admin_submenu-page('voting-plugin', --('Help', 'voting'),
            osc_admin-render-plugin_url(osc_plugin_folder(__FILE__) . 'help.php'), '
            voting-plugin-help', 'administrator');
    } else {
        osc_add_hook('admin-menu', 'voting-admin-menu');
    }

    osc_enqueue_style('admin-css', osc_base_url() . 'oc-content/plugins/ads-rating/css/admin.css
');
}

function voting-admin-menu() {
    echo '<h3><a href="#">' . --('Voting options', 'voting') . '</a></h3>
<ul>
    <li><a href="' . osc_admin-render-plugin_url(osc_plugin_folder(__FILE__) . 'conf.php') .
        '">&raquo;' . --('Settings', 'voting') . '</a></li>
    <li><a href="' . osc_admin-render-plugin_url(osc_plugin_folder(__FILE__) . 'help.php') .
        '">&raquo;' . --('Help', 'voting') . '</a></li>
</ul>';
}

/* end Adding plugin menu */

/* Item Detail
-----*/

// Show an item special attributes
osc_add_hook('item-detail', 'ads-item-detail');

/* Show form to rate an item. (itemDetail) */
function ads-item-detail() {
    // Fail-early validation, terminate the function if not allowed in the category.
    if (! osc_is-this-category(osc_get-preference('ads-rating', 'plugin-name'),
        osc_item-category-id()) ) return;

    // Fail early validation, terminate the function if plugin is disabled.
    if (! osc_get-preference('ads-rating', 'rating') == '1') return;

    $avg = AR::newInstance()->getItemAvgRating( osc_item-id() );
    $vote['vote'] = $avg['vote'];
}

```

```

$totalVotes = AR::newInstance()->getItemNumberOfVotes( osc_item_id() );
$vote['total'] = $totalVotes["total"];

$hash = '';

if( osc_logged_user_id() == 0 ) {
    $hash = $_SERVER['HTTP_USER_AGENT'] . $_SERVER['REMOTE_ADDR'];
    $hash = sha1($hash);
}
else {
    $hash = null;
}

$vote['can_vote'] = can_vote( osc_item_id(), osc_logged_user_id(), $hash );

require 'item_detail.php';
//AR::newInstance()->_initSlopeOneTable(); //compute the slope ones after receiving a rating
}

/**
 * Check if user can rate an item
 *
 * @param string $itemId
 * @param string $userId
 * @param string $hash
 * @return bool
 */
function can_vote( $itemId, $userId, $hash ) {
    if ( $userId == 'NULL' ) {
        $result = AR::newInstance()->getItemIsRated($itemId, $hash);
    }
    else {
        $result = AR::newInstance()->getItemIsRated($itemId, $hash, $userId);
    }

    if ( count($result) > 0 ) return false;
    else return true;
}

/* end Item Detail */

/* Delete item
-----*/
osc_add_hook('delete.item', '_item_delete');

// Delete item via itemId
function voting_item_delete($itemId) {
    return ModelVoting::newInstance()->deleteItem($itemId);
}

/* end Delete Item */

/* Output functions
-----*/

function echo_recommended_ads() {
    if ( 1 != osc_get_preference('ads.rating', 'rating') ) return;

    // Shows the recommended ads
    // $results = AR::newInstance()->get_predict_best( osc_logged_user_id(), $item_id );
    // $results =
    AR::newInstance()->_initSlopeOneTable(); //compute the slope ones after receiving a rating
    $results = AR::newInstance()->getPredictBest(osc_logged_user_id());
    //return $results;

    require_once 'show_recommended_listings.php';
}

/*****
 *
 * VOTING ON ITEMS
 *****/

/**
 * Return layout optimized for sidebar at main web page, with the best items voted with a
 * limit
 *
 * @param int $num number of items
 */
function echo_best_rated($num = 5)
{
    if( osc_get_preference('item_voting', 'voting') == 1 ) {

```

```

        $filter = array(
            'order' => 'desc',
            'num_items' => $num
        );
        $results = get_votes($filter);
        if(count($results) > 0) {
            error_log( print_r($results, true) );
            $locale = osc_current_user_locale();
            require 'set_results.php';
        }
    }
}

/**
 * Return an array of item votes with given filters
 * <code>
 * array(
 *     'category_id' => (integer_category_id),
 *     'order' => ('desc','asc'),
 *     'num_items' => (integer)
 * );
 * </code>
 * @param type $array_filters
 * @return array of item votes
 */
function get_votes($array_filters)
{
    $category_id = null;
    $order = 'desc';
    $num = 5;

    if(isset($array_filters['category_id'])){
        $category_id = $array_filters['category_id'];
    }
    if(isset($array_filters['order'])){
        $order = strtolower($array_filters['order']);
        if( !in_array($order, array('desc', 'asc') ) ){
            $order = 'desc';
        }
    }
    if(isset($array_filters['num_items'])){
        $num = (int)$array_filters['num_items'];
    }

    return AR::newInstance()->getItemRatings($category_id, $order, $num);
}

/**
 * Print star img src, how many stars will appear given an average of votes
 *
 * @param type $star
 * @param type $avg_vote
 * @return type
 */
function voting_star($star, $avg_vote)
{
    $path = osc_base_url().'/oc-content/plugins/'. osc_plugin_folder(__FILE__);
    $star_ok = $path.'img/ico_vot_ok.gif';
    $star_no = $path.'img/ico_vot_no.gif';
    $star_md = $path.'img/ico_vot_md.gif';

    if( $avg_vote >= $star ) { //is average of votes greater than number of stars being
        considered
        echo $star_ok;
    } else {
        $aux = 1+($avg_vote - $star);
        if($aux <= 0){
            echo $star_no;
            return true;
        }
        if($aux >=1) {
            echo $star_no;
        } else {
            if($aux <= 0.5){
                echo $star_md;
            }else{
                echo $star_ok;
            }
        }
    }
}

<?php $path = osc_base_url().'/oc-content/plugins/'. osc_plugin_folder(__FILE__); ?>
<div id="wrapper_voting_plugin">
<script type="text/javascript">
$(function(){
    $('aPs').click(function(){
        var params = '';
        var vote = 0;
        if( $(this).hasClass('vote1') ) vote = 1;
        if( $(this).hasClass('vote2') ) vote = 2;
        if( $(this).hasClass('vote3') ) vote = 3;
        if( $(this).hasClass('vote4') ) vote = 4;
        if( $(this).hasClass('vote5') ) vote = 5;

        var itemId = <?php echo osc_item_id(); ?>;
        params = 'itemId='+itemId+'&vote='+vote;

        $.ajax({

```



```

        type: "POST",
        url: '<?php echo osc_base_url(true); ?>?page=ajax&action=custom&ajaxfile=<?php
            echo osc_plugin_folder(__FILE__).'ajax.php'?>&'+params,
        dataType: 'text',
        beforeSend: function () {
            $('#voting-plugin').hide();
            $('#voting-loading').fadeIn('slow');
        },
        success: function (data) {
            $('#voting-loading').fadeOut('slow', function () {
                $('#voting-plugin').html(data).fadeIn('slow');
            });
        }
    });
});
</script>
<style>
    #wrapper_voting_plugin {
        margin-top: 10px;
        padding: 5px;
    }

    .votes_txt_vote {
        display: inline;
        float: left;
    }

    .votes_results {
        display: inline;
        float: left;
    }

    .votes_results img {
        height: auto;
        margin-top: -1px;
        margin-left: -2px;
        vertical-align: middle;
        width: auto;
        border: 0 none;
        margin: 0;
        padding: 0;
        float: left;
    }

    .votes_star .vote1 {
        width: 15px;
        z-index: 14;
    }
    .votes_star .vote2 {
        width: 30px;
        z-index: 13;
    }
    .votes_star .vote3 {
        width: 45px;
        z-index: 12;
    }
    .votes_star .vote4 {
        width: 60px;
        z-index: 11;
    }
    .votes_star .vote5 {
        width: 75px;
        z-index: 10;
    }

    .votes_star a {
        display: block;
        height: 19px;
        position: absolute;
    }

    .votes_star {
        background: url("<?php echo osc_base_url().'oc-content/plugins/'.
            osc_plugin_folder(__FILE__);?>img/ico_vot_vo.gif") repeat scroll 0 0
            transparent;
        display: inline;
        float: left;
        height: 20px;
        margin: 0 4px 0 3px;
        position: relative;
        width: 76px;
    }

    .votes_vote {
        display: inline;
        float: left;
        margin-right: 5px;
    }

    .votes_star a:hover {
        background: url("<?php echo osc_base_url().'oc-content/plugins/'.
            osc_plugin_folder(__FILE__);?>img/ico_vot_ov.gif") repeat-x scroll 0 0
            transparent;
    }

    #voting-plugin {
        position: relative;
    }

```

```

    }
</style>
<span id="voting_loading" style="display:none;"> <?php _e('Loading', 'voting');?></span>
<div id="voting_plugin">
    <?php include('view_votes.php');?>
</div>
<div style="clear:both;"></div>
</div>

<?php $path = osc_base_url().'/oc-content/plugins/'. osc_plugin_folder(__FILE__); ?>
<div id="wrapper_voting_plugin">
<script type="text/javascript">
$(function(){
    $('#aPvu').click(function(){
        var params = '';
        var vote = 0;
        if( $(this).hasClass('vote1') ) vote = 1;
        if( $(this).hasClass('vote2') ) vote = 2;
        if( $(this).hasClass('vote3') ) vote = 3;
        if( $(this).hasClass('vote4') ) vote = 4;
        if( $(this).hasClass('vote5') ) vote = 5;

        var userId = <?php echo osc_item_user_id(); ?>;
        params = 'userId='+userId+'&vote='+vote;

        $.ajax({
            type: "POST",
            url: '<?php echo osc_base_url(true); ?>?page=ajax&action=custom&ajaxfile=<?php
echo osc_plugin_folder(__FILE__).'ajax.php'?>&'+params,
            dataType: 'text',
            beforeSend: function () {
                $('#voting_plugin_user').hide();
                $('#voting_loading_user').fadeIn('slow');
            },
            success: function (data) {
                $('#voting_loading_user').fadeOut('slow', function () {
                    $('#voting_plugin_user').html(data).fadeIn('slow');
                });
            }
        });
    });
});
</script>

<style>
#wrapper_voting_plugin{
    margin-top:10px;
    padding: 5px;
}

.votes_txt_vote {
    display: inline;
    float: left;
}

.votes_results {
    display: inline;
    float: left;
}

.votes_results img {
    height: auto;
    margin-top: -1px;
    margin-left: -2px;
    vertical-align: middle;
    width: auto;
    border: 0 none;
    margin: 0;
    padding: 0;
    float: left;
}

.votes_star .vote1 {
    width: 15px;
    z-index: 14;
}
.votes_star .vote2 {
    width: 30px;
    z-index: 13;
}
.votes_star .vote3 {
    width: 45px;
    z-index: 12;
}
.votes_star .vote4 {
    width: 60px;
    z-index: 11;
}
.votes_star .vote5 {
    width: 75px;
    z-index: 10;
}

.votes_star a {
    display: block;
    height: 19px;
    position: absolute;
}

```

```

        .votes_star {
            background: url("php echo osc_base_url().'/oc-content/plugins/'.
                osc_plugin_folder(__FILE__);?&gt;img/ico_vot_vo.gif") repeat scroll 0 0
                transparent;
            display: inline;
            float: left;
            height: 20px;
            margin: 0 4px 0 3px;
            position: relative;
            width: 76px;
        }

        .votes_vote {
            display: inline;
            float: left;
            margin-right: 5px;
        }

        .votes_star a:hover {
            background: url("<?php echo osc_base_url().'/oc-content/plugins/'.
                osc_plugin_folder(__FILE__);?&gt;img/ico_vot_ov.gif") repeat-x scroll 0 0
                transparent;
        }

        #voting_plugin_user {
            position: relative;
        }
    &lt;/style&gt;
    &lt;span id="voting_loading_user" style="display:none;"&gt;&lt;img src="<?php echo $path; ?&gt;img/
        spinner.gif" style="margin-left:20px;"/&gt; &lt;?php _e('Loading', 'voting');?&gt;&lt;/span&gt;
    &lt;div id="voting_plugin_user"&gt;
        &lt;?php include('view_votes_user.php');?&gt;
    &lt;/div&gt;
    &lt;div style="clear:both;"&gt;&lt;/div&gt;
&lt;/div&gt;

&lt;style&gt;
#wrapper_voting_plugin{
    margin-top:10px;
    padding: 5px;
}

.votes_txt_vote {
    display: inline;
    float: left;
}

.votes_results {
    display: inline;
    float: left;
}

.votes_results img {
    height: auto;
    margin-top: -1px;
    margin-left: -2px;
    vertical-align: middle;
    width: auto;
    border: 0 none;
    margin: 0;
    padding: 0;
    float: left;
}

.votes_star .vote1 {
    width: 15px;
    z-index: 14;
}
.votes_star .vote2 {
    width: 30px;
    z-index: 13;
}
.votes_star .vote3 {
    width: 45px;
    z-index: 12;
}
.votes_star .vote4 {
    width: 60px;
    z-index: 11;
}
.votes_star .vote5 {
    width: 75px;
    z-index: 10;
}

.votes_star a {
    display: block;
    height: 19px;
    position: absolute;
}

.votes_star {
    background: url("<?php echo osc_base_url().'/oc-content/plugins/'.
        osc_plugin_folder(
            __FILE__);?&gt;img/ico_vot_vo.gif") repeat scroll 0 0 transparent;
    display: inline;
    float: left;
    height: 20px;
    margin: 0 4px 0 3px;
</pre

```

```

        position: relative;
        width: 76px;
    }

    .votes_vote {
        display: inline;
        float: left;
        margin-right: 5px;
    }

    .votes_star a: hover {
        background: url("php echo osc_base_url().'/oc-content/plugins/'. osc_plugin_folder(
            _FILE_);?&gt;img/ico_vot_ov.gif") repeat-x scroll 0 0 transparent;
    }

    #voting_plugin {
        position: relative;
    }
&lt;/style&gt;
&lt;div class="box location"&gt;
    &lt;h3&gt;&lt;strong&gt;&lt;?php _e("Best rated", 'voting') ; ?&gt;&lt;/strong&gt;&lt;/h3&gt;
    &lt;ul&gt;
        &lt;?php
            $count = 0;
            View::newInstance()-&gt;_erase('items');
            View::newInstance()-&gt;_erase('item');
            foreach($results as $item_vote):
                $avg_vote = $item_vote['avg_vote'];
                $total     = $item_vote['num_votes'];
                $item      = Item::newInstance()-&gt;findByPrimaryKey($item_vote['item_id']);
                View::newInstance()-&gt;exportVariableToView('item', $item );
            ?&gt;
            &lt;li&gt;
                &lt;?php if($count+1 &lt; count($results)){?&gt;
                &lt;div style="border-bottom:1px black dashed;?&gt;
                &lt;?php } else { ?&gt;
                &lt;div&gt;
                &lt;?php } ?&gt;
                &lt;p style="text-align: center;?&gt;&lt;a href="&lt;?php echo osc_item_url(); ?&gt;"&gt;&lt;?php
                    echo osc_item_title(); ?&gt;&lt;/a&gt;&lt;/p&gt;
                &lt;p style="text-align: center;?&gt;
                    &lt;img title="&lt;?php _e('Without interest', 'voting');?&gt;" src="&lt;?php
                        voting_star(1, $avg_vote); ?&gt;"&gt;
                    &lt;img title="&lt;?php _e('Uninteresting', 'voting');?&gt;" src="&lt;?php voting_star
                        (2, $avg_vote); ?&gt;"&gt;
                    &lt;img title="&lt;?php _e('Interesting', 'voting');?&gt;" src="&lt;?php voting_star
                        (3, $avg_vote); ?&gt;"&gt;
                    &lt;img title="&lt;?php _e('Very interesting', 'voting');?&gt;" src="&lt;?php
                        voting_star(4, $avg_vote); ?&gt;"&gt;
                    &lt;img title="&lt;?php _e('Essential', 'voting');?&gt;" src="&lt;?php voting_star(5,
                        $avg_vote); ?&gt;"&gt;
                    &lt;span style="position: relative; top:-5px; padding-right: 4px; padding-left:
                        4px; margin-bottom: 3px;?&gt;&lt;?php echo $total;?&gt; &lt;?php _e('ratings', '
                        voting');?&gt;&lt;/span&gt;
                &lt;/p&gt;
                &lt;/div&gt;
            &lt;/li&gt;
        &lt;?php
            $count++;
            View::newInstance()-&gt;_erase('item') ;
        endforeach;
    ?&gt;
&lt;/ul&gt;
&lt;/div&gt;
&lt;style&gt;
    #wrapper_voting_plugin{
        margin-top:10px;
        padding: 5px;
    }

    .votes_txt_vote {
        display: inline;
        float: left;
    }

    .votes_results {
        display: inline;
        float: left;
    }

    .votes_results img {
        height: auto;
        margin-top: -1px;
        margin-left: -2px;
        vertical-align: middle;
        width: auto;
        border: 0 none;
        margin: 0;
        padding: 0;
        float: left;
    }

    .votes_star .vote1 {
        width: 15px;
        z-index: 14;
    }
    .votes_star .vote2 {
</pre

```

```

        width: 30px;
        z-index: 13;
    }
    .votes_star .vote3 {
        width: 45px;
        z-index: 12;
    }
    .votes_star .vote4 {
        width: 60px;
        z-index: 11;
    }
    .votes_star .vote5 {
        width: 75px;
        z-index: 10;
    }
    .votes_star a {
        display: block;
        height: 19px;
        position: absolute;
    }
    .votes_star {
        background: url("<?php echo osc_base_url().'/oc-content/plugins/'. osc_plugin_folder(
            __FILE__);?>img/ico_vot_vo.gif") repeat scroll 0 0 transparent;
        display: inline;
        float: left;
        height: 20px;
        margin: 0 4px 0 3px;
        position: relative;
        width: 76px;
    }
    .votes_vote {
        display: inline;
        float: left;
        margin-right: 5px;
    }
    .votes_star a: hover {
        background: url("<?php echo osc_base_url().'/oc-content/plugins/'. osc_plugin_folder(
            __FILE__);?>img/ico_vot_ov.gif") repeat-x scroll 0 0 transparent;
    }
    #voting-plugin {
        position: relative;
    }
</style>
<div class="box location">
<h3><strong><?php _e("Best users voted", 'voting') ; ?></strong></h3>
<ul>
<?php
$count = 0;
foreach($results as $user_vote):
    $avg_vote = $user_vote['avg_vote'];
    $total    = $user_vote['num_votes'];
    $user     = User::newInstance()->findbyPrimaryKey($user_vote['user_id']);
    View::newInstance()->exportVariableToView('user', $user) ;
?>
<li>
<?php if($count+1 < count($results)){?>
<div style="border-bottom:1px black dashed;">
<?php } else { ?>
<div>
<?php } ?>
<p style="text-align: center;"><a href="<?php echo osc_user_public_profile_url(
    ); ?>"><?php echo osc_user_name(); ?></a></p>
<p style="text-align: center;">
<img title="<?php _e('Without interest', 'voting');?>" src="<?php
    voting_star(1, $avg_vote); ?>">
<img title="<?php _e('Uninteresting', 'voting');?>" src="<?php voting_star
    (2, $avg_vote); ?>">
<img title="<?php _e('Interesting', 'voting');?>" src="<?php voting_star
    (3, $avg_vote); ?>">
<img title="<?php _e('Very interesting', 'voting');?>" src="<?php
    voting_star(4, $avg_vote); ?>">
<img title="<?php _e('Essential', 'voting');?>" src="<?php voting_star(5,
    $avg_vote); ?>">
<span style="position: relative; top: -5px; padding-right: 4px; padding-left:
    4px; margin-bottom: 3px;"><?php echo $total;?> <?php _e('votes', '
    voting');?></span>
</p>
</div>
</li>
<?php
    $count++;
    View::newInstance()->erase('user') ;
endforeach;
?>
</ul>
</div>
<style>
#wrapper_voting_plugin{
    margin-top: 10px;
    padding: 5px;
}

```

```

.votes_txt_vote {
    display: inline;
    float: left;
}

.votes_results {
    display: inline;
    float: left;
}

.votes_results img {
    height: auto;
    margin-top: -1px;
    margin-left: -2px;
    vertical-align: middle;
    width: auto;
    border: 0 none;
    margin: 0;
    padding: 0;
    float: left;
}

.votes_star .vote1 {
    width: 15px;
    z-index: 14;
}
.votes_star .vote2 {
    width: 30px;
    z-index: 13;
}
.votes_star .vote3 {
    width: 45px;
    z-index: 12;
}
.votes_star .vote4 {
    width: 60px;
    z-index: 11;
}
.votes_star .vote5 {
    width: 75px;
    z-index: 10;
}

.votes_star a {
    display: block;
    height: 19px;
    position: absolute;
}

.votes_star {
    background: url("php echo osc_base_url().'/oc-content/plugins/'. osc_plugin_folder(
        __FILE__);?&gt;img/ico_vot_vo.gif") repeat scroll 0 0 transparent;
    display: inline;
    float: left;
    height: 20px;
    margin: 0 4px 0 3px;
    position: relative;
    width: 76px;
}

.votes_vote {
    display: inline;
    float: left;
    margin-right: 5px;
}

.votes_star a:hover {
    background: url("<?php echo osc_base_url().'/oc-content/plugins/'. osc_plugin_folder(
        __FILE__);?&gt;img/ico_vot_ov.gif") repeat-x scroll 0 0 transparent;
}

#voting-plugin {
    position: relative;
}
&lt;/style&gt;

&lt;div class="ads-rating"&gt;
&lt;p&gt;&lt;/p&gt;
&lt;?php if(count($results)&gt;0) { ?&gt;
&lt;h2&gt; Recommended listings &lt;/h2&gt;

&lt;?php
$count = 0;

//holds id of ad before importing ids of recommended ads
$ad_id = osc_item_id();

foreach($results as $item_vote){

    $item = Item::newInstance()-&gt;findByPrimaryKey($item_vote['item']);
    View::newInstance()-&gt;exportVariableToView('item', $item );

    $rsrc = View::newInstance()-&gt;_get('item');

    //get item id of ad
    $rsrc_id = (int)osc_field($rsrc,'pk_i_id','');

    //get corresponding id of thumbnail of the ad
</pre

```

```

        $thumbRes = AR::newInstance()->getThumbnailId($src_id);
        $thumb_id = $thumbRes['thumb_id'];

        //get corresponding extension of thumbnail of the ad
        $thumbRes = AR::newInstance()->getThumbnailExt($src_id, $thumb_id);
        $thumb_ext = $thumbRes['thumb_ext'];

        $size = explode('x', osc_thumbnail_dimensions());
    ?>

    <ul class="listing-card-list">
        <li class="listing-card">
            <?php if( osc_images_enabled_at_items() ) { ?>
                <?php if(osc_count_item_resources()) { ?>
                    <a class="listing-thumb" href="<?php echo osc_item_url() ; ?>" title="<?php echo
                        osc_item_title() ; ?>">
                        <!-- " width="<?php echo $size[0]; ?>" height="<?php
                            echo $size[1]; ?>"-->
                            "
                                width="<?php echo $size[0]; ?>"
                                height="<?php echo $size[1]; ?>"
                            </a>
                    <?php } else { ?>
                        <a class="listing-thumb" href="<?php echo osc_item_url() ; ?>" title="<?php echo
                            osc_item_title() ; ?>">" width="<?
                                    php echo $size[0]; ?>" height="<?php echo $size[1]; ?>"></a>
                    <?php } ?>
                <?php } ?>
            <div class="listing-detail">
                <div class="listing-cell">
                    <div class="listing-data">
                        <div class="listing-basicinfo">
                            <a href="<?php echo osc_item_url() ; ?>" class="title" title="<?php echo
                                osc_item_title() ; ?>"><?php echo osc_item_title() ; ?></a>
                            <div class="listing-attributes">
                                <span class="category"><?php echo osc_item_category() ; ?></span> -
                                <span class="location"><?php echo osc_item_city(); ?> (<?php echo
                                    osc_item_region(); ?>)</span> <span class="g-hide"><?php echo
                                        osc_format_date(osc_item_pub_date()); ?>
                                <?php if( osc_price_enabled_at_items() ) { ?><span class="currency-
                                    value"><?php echo osc_format_price(osc_item_price()); ?></span><?
                                        php } ?>
                            </div>
                            <p><?php echo osc_highlight( strip_tags( osc_item_description() ),250) ;
                                ?></p>
                        </div>
                        <?php if($admin){ ?>
                            <span class="admin-options">
                                <a href="<?php echo osc_item_edit_url(); ?>" rel="nofollow"><?php _e('
                                    Edit item', 'bender'); ?></a>
                                <span></span>
                                <a class="delete" onclick="javascript:return confirm('<?php echo
                                    osc_esc_js(_('This action can not be undone. Are you sure you
                                        want to continue?', 'bender'))); ?>'" href="<?php echo
                                        osc_item_delete_url(); ?>" ><?php _e('Delete', 'bender'); ?></a>
                                <?php if(osc_item_is_inactive()) {?>
                                    <span></span>
                                    <a href="<?php echo osc_item_activate_url(); ?>" ><?php _e('Activate',
                                        'bender'); ?></a>
                                <?php } ?>
                            </span>
                        <?php } ?>
                    </div>
                </div>
            </div>
        </li>
    </ul>

    <?php
        $count++;
        //View::newInstance()->_erase('temp_item') ;
    } //end foreach

    //return osc_item to id of ad
    $item = Item::newInstance()->findByPrimaryKey($ad_id);
    View::newInstance()->exportVariableToView('item', $item) ;
}
?>

</div>

CREATE TABLE /*TABLE_PREFIX*/t_rated_items (
    fk_i_item_id INT UNSIGNED NOT NULL,
    fk_i_user_id INT UNSIGNED,
    i_value decimal(14,4) UNSIGNED NOT NULL,
    t_datetimestamp TIMESTAMP NOT NULL,
    s_hash VARCHAR(255) NULL,
    FOREIGN KEY (fk_i_item_id) REFERENCES /*TABLE_PREFIX*/t_item (pk_i_id)
) ENGINE=InnoDB DEFAULT CHARACTER SET 'UTF8' COLLATE 'UTF8_GENERAL_CI';

```

```

CREATE TABLE /*TABLE_PREFIX*/t_dev (
  i_item_id_1 int (10) NOT NULL default '0' ,
  i_item_id_2 int (10) NOT NULL default '0' ,
  i_count int (10) NOT NULL default '0' ,
  i_sum decimal(14,4) NOT NULL DEFAULT '0.0000' ,
  PRIMARY KEY(i_item_id_1 , i_item_id_2)
) ENGINE=InnoDB DEFAULT CHARACTER SET 'UTF8' COLLATE 'UTF8_GENERAL_CI';

<?php $path = osc_base_url().'/oc-content/plugins/'. osc_plugin_folder(__FILE__); ?>
<div class="votes_stars">
  <?php if( $vote['can_vote'] && osc_is_web_user_logged_in()) { ?>
    <div class="votes_vote">
      <div class="votes_txt_vote"><?php _e('Vote', 'ads_rating');?></div>
      <div class="votes_star">
        <span id="">
          <a href="#" rel="nofollow" title="<?php _e('Without interest', '
            ads_rating');?>" class="aPs votel"></a>
          <a href="#" rel="nofollow" title="<?php _e('Uninteresting', '
            ads_rating');?>" class="aPs vote2"></a>
          <a href="#" rel="nofollow" title="<?php _e('Interesting', 'ads_rating
            ');?>" class="aPs vote3"></a>
          <a href="#" rel="nofollow" title="<?php _e('Very interesting', '
            ads_rating');?>" class="aPs vote4"></a>
          <a href="#" rel="nofollow" title="<?php _e('Essential', 'ads_rating')
            ;?>" class="aPs vote5"></a>
        </span>
      </div>
      
    </div>
  <?php } ?>
  <div class="votes_results">
    <span style="float:left; padding-right: 4px;"><?php _e('Result', 'ads_rating')
      ;?> </span>
    <?php
      $avg_vote = $vote['vote'];
    ?>
    <img title="<?php _e('Without interest', 'ads_rating');?>" src="<?php
      voting_star(1, $avg_vote); ?>">
    <img title="<?php _e('Uninteresting', 'ads_rating');?>" src="<?php voting_star
      (2, $avg_vote); ?>">
    <img title="<?php _e('Interesting', 'ads_rating');?>" src="<?php voting_star
      (3, $avg_vote); ?>">
    <img title="<?php _e('Very interesting', 'ads_rating');?>" src="<?php
      voting_star(4, $avg_vote); ?>">
    <img title="<?php _e('Essential', 'ads_rating');?>" src="<?php voting_star(5,
      $avg_vote); ?>">
    <span style="float:left; padding-right: 4px; padding-left: 4px;"><?php echo
      $vote['total'];?> <?php _e('votes', 'ads_rating');?></span>
  </div>
</div>

<?php $path = osc_base_url().'/oc-content/plugins/'. osc_plugin_folder(__FILE__); ?>
<div class="votes_stars">
  <div style="float:left;padding-right: 5px;"><?php $aux = User::newInstance()->
    findByPrimaryKey($vote['userId']); echo $aux['s_name']; ?></div>
  <?php if( $vote['can_vote'] ) { ?>
    <div class="votes_vote">
      <div class="votes_star">
        <span id="">
          <a href="#" rel="nofollow" title="<?php _e('Without interest', 'voting
            ');?>" class="aPvu votel"></a>
          <a href="#" rel="nofollow" title="<?php _e('Uninteresting', 'voting')
            ;?>" class="aPvu vote2"></a>
          <a href="#" rel="nofollow" title="<?php _e('Interesting', 'voting')
            ;?>" class="aPvu vote3"></a>
          <a href="#" rel="nofollow" title="<?php _e('Very interesting', 'voting
            ');?>" class="aPvu vote4"></a>
          <a href="#" rel="nofollow" title="<?php _e('Essential', 'voting');?>"
            class="aPvu vote5"></a>
        </span>
      </div>
      
    </div>
  <?php } ?>
  <div class="votes_results">
    <?php
      $avg_vote = $vote['vote'];
    ?>
    <img title="<?php _e('Without interest', 'voting');?>" src="<?php voting_star
      (1, $avg_vote); ?>">
    <img title="<?php _e('Uninteresting', 'voting');?>" src="<?php voting_star(2,
      $avg_vote); ?>">
    <img title="<?php _e('Interesting', 'voting');?>" src="<?php voting_star(3,
      $avg_vote); ?>">
    <img title="<?php _e('Very interesting', 'voting');?>" src="<?php voting_star
      (4, $avg_vote); ?>">
    <img title="<?php _e('Essential', 'voting');?>" src="<?php voting_star(5,
      $avg_vote); ?>">
    <span style="float:left; padding-right: 4px; padding-left: 4px;"><?php echo
      $vote['total'];?> <?php _e('votes', 'voting');?></span>
  </div>
</div>
<?php

```



```

$item = array(
    'fk_i_user_id' => osc_logged_user_id() ,
    'pk_i_id' => Params::getParam( 'itemid' )
);

// Push the updates
update_tags( $item );

prepareLink();

function prepareLink() {
    $tags = explode( ',', Params::getParam( 'tagger-field' ) );
    foreach( $tags as $tag ) : ?>
        <a href="<?php echo osc_search_url( array( 'tag' => urlencode( $tag ) ) )
            ?>"><small><?php echo $tag ?></small></a><?php echo $tag != end( $tags ) ?
            ', ' : '' ?>
    <?php endforeach;
}

<?php
/*
Plugin Name: Tagger
Description: This plugin adds a tagging and search by tag functions
Version: 1.0
Short Name: tagger
*/

require_once 'TaggerModel.php';

/* Initialize plugin
-----*/

// This is to add the admin link for this page.
osc_add_hook("init", "_tagger_load_scripts");
osc_add_hook("init_admin", "_tagger_load_scripts");

// Initialize plugin informations and other necessary objects.
function _tagger_load_scripts() {
    osc_enqueue_style('tagit', osc_base_url() . 'oc-content/plugins/tagger/css/jquery.tagit.css');
    //osc_enqueue_style('tagit', osc_base_url() . 'oc-content/plugins/tagger/css/jquery.tagit-simple-grey.css');
    osc_register_script('tagit', osc_base_url() . 'oc-content/plugins/tagger/js/tag-it.min.js', 'jquery-ui');
    //osc_enqueue_style('tagit', osc_base_url() . 'oc-content/plugins/tagger/css/jquery-ui-1.10.3.custom.css');
    osc_enqueue_script('tagit');
}

/* end initialize plugin */

/* Install Tagger plugin.
-----*/

// This is needed in order to be able to activate the plugin
osc_register_plugin(osc_plugin_path(__FILE__), '_tagger_install');

// Set plugin preferences
function _tagger_install() {
    TaggerModel::newInstance()->import('tagger/struct.sql');
}

/* end Install Tagger plugin. */

/* Uninstall Tagger plugin.
-----*/

// This is a hack to show a Uninstall link at plugins table (you could also use some other
// hook to show a custom option panel)
osc_add_hook(osc_plugin_path(__FILE__)."_uninstall", '_tagger_uninstall');

// Delete plugin preferences
function _tagger_uninstall() {
    TaggerModel::newInstance()->uninstall();
}

/* end Uninstall Tagger plugin. */

/* Admin Configuration
-----*/

```

```

// This is a hack to show a Configure link at plugins table (you could also use some other
hook to show a custom option panel)
osc_add_hook(osc_plugin_path(__FILE__)." _configure", '_tagger_admin_configuration');

function _tagger_admin_configuration() {

    // Standard configuration page for plugin which extend item's attributes
    osc_plugin_configure_view(osc_plugin_path(__FILE__));

}

/* end Admin Configuration */

/* Item Form
-----*/

// When publishing an item we show an extra form with more attributes
osc_add_hook('item_form', 'tag_form_control');

function tag_form_control( $catId='') {

    // if ( osc_is_admin_user_logged_in() ) return;
    if($catId!="") {

        $plugin_info = osc_plugin_get_info('tagger/index.php');

        // Fail-early validation, terminate the function if not allowed in the category.
        if ( ! osc_is_this_category( $plugin_info[ 'short_name' ], $catId ) ) return;

        // This is used as a default available tags for tag-it plugin.
        $allTags = TaggerModel::newInstance()->getAllTags();

        $allTags = TaggerModel::newInstance()->array_pluck( 'tag', $allTags );

        require_once 'item_edit.php';
    }

}

/* end Item Form */

/* Post form
-----*/

// To add that new information to our custom table
osc_add_hook('posted_item', 'save_tags');

function save_tags( $item ) {

    TaggerModel::newInstance()->saveTags( $item[ 'fk.i.user_id' ], $item[ 'pk.i.id' ], Params::
        getParam( 'tagger-field' ) );

}

/* end Post Form */

/* Edit form
-----*/

// Edit an item special attributes
osc_add_hook('item_detail', 'show_tags_on_view');

function show_tags_on_view( $item ) {

    $plugin_info = osc_plugin_get_info('tagger/index.php');

    if ( ! osc_is_this_category( $plugin_info[ 'short_name' ], osc_item_category_id() ) ) return
        ;

    $tags = TaggerModel::newInstance()->getTags( $item[ 'pk.i.id' ] );

    $allTags = TaggerModel::newInstance()->array_pluck( 'tag', $tags );

    require_once( 'item_detail.php' );

}

/* end Edit Form */

/* Item Details
-----*/

// Edit an item special attributes
osc_add_hook('item_edit', 'show_edit_form_control');

function show_edit_form_control( $cat_id, $item_id ) {

    if ( osc_is_admin_user_logged_in() ) return;

    $tags = TaggerModel::newInstance()->getTags( $item_id );

    $tags = TaggerModel::newInstance()->array_pluck('tag', $tags );

    $allTags = TaggerModel::newInstance()->array_pluck( 'tag', $tags );

    require_once 'item_edit.php';
}

```

```

}

/* end Edit Form */

/* Item Update
-----*/

// Edit an item special attributes
osc_add_hook('edited_item', 'update_tags');

function update_tags( $item ) {

    // Remove all tags assigned to user first
    TaggerModel::newInstance()->removeAllTags( $item[ 'fk.i.user.id' ], $item[ 'pk.i.id' ] );

    // Resave the tags
    TaggerModel::newInstance()->saveTags( $item[ 'fk.i.user.id' ], $item[ 'pk.i.id' ], Params::
        getParam( 'tagger-field' ) );

}

/* Tag Search
-----*/

// When searching, add some conditions
osc_add_hook('search_conditions', 'search_condition');

function search_condition( $params ) {

    // Fail early validation.
    if ( ! array_key_exists( 'tag', $params ) ) {
        return;
    }

    Search::newInstance()->addConditions(sprintf("%st_tagger_tags.tag LIKE '%s'",
        DB.TABLE.PREFIX, $params[ 'tag' ]));
    Search::newInstance()->addConditions(sprintf("%st_tagger_tags.id = %st_tagger_tagged_objects
        .tag_id ", DB.TABLE.PREFIX, DB.TABLE.PREFIX));
    Search::newInstance()->addConditions(sprintf("%st_item.pk.i.id = %st_tagger_tagged_objects.
        object_id ", DB.TABLE.PREFIX, DB.TABLE.PREFIX));
    Search::newInstance()->addTable(sprintf("%st_tagger_tagged_objects", DB.TABLE.PREFIX));
    Search::newInstance()->addTable(sprintf("%st_tagger_tags", DB.TABLE.PREFIX));

}

/* end Tag Search */

/* Tag Search Form
-----*/

osc_add_hook('search_form', 'search_form');

function search_form( $cat_id = NULL )
{
    include_once( 'search_form.php' );
}

/* end Tag Search Form */

<div id="tagger-tag-module" class="tagger-tag-module">
    <h5 class="tagger-head"><?php _e('Tags', 'tagger') ; ?> : </h5>
    <div class="tagger-tags-list">
        <?php if ( empty( $tags ) ) : ?>
            <p class="tagger-no-tag" > <em>No tags available </em></p>
        <?php else : ?>
            <?php foreach( $tags as $tag ) : ?>
                <a href="<?php echo osc_search_url( array( 'tag' => urlencode( $tag[ 'tag' ] ) ) )
                    ?>"><small><?php echo $tag[ 'tag' ] ?></small></a><?php echo $tag != end( $tags )
                    ? ', ' : ' ' ?>
            <?php endforeach; ?>
        <?php endif; ?>
    </div>
    <?php if ( osc_is_web_user_logged_in() ) : ?>
        <a href="javascript:void(0)" id="tagger-tags-modify"> <small>Modify Tags</small>
        > </a>
        <input type="text" class="tagger-tags-input" id="tagger-tags-input" name="
            tagger-field" value="<?php echo implode(', ', $allTags ) ?>" />
        <?php endif; ?>
    </div>

<script>
jQuery(document).ready( function($) {

    $('#tagger-tags-input').tagit({
        placeholderText : 'Input tags here...',
        availableTags : <?php echo json_encode( $allTags ) ?>,
        afterTagAdded : function() {
            // $('tagit-close').hide();
        }
    });

    $('#tagger-tags-modify').click( function() {
        $('#tagger-tag-module').find('tagger-tags-list').css('display', 'none');
        $('#tagger-tag-module').find('tagger-tags-modify').css('display', 'none');
    });
}

```

```

        $('#tagger-tag-module').find('ul.tagit')
                                                    .css( 'display', '
                                                    block' )
                                                    .find( 'input[ type="
                                                    text" ]' )
                                                    .focus();
    });

    $('#tagger-tag-module').find('ul.tagit').focusout(function(){
        var tags = $('#tagger-tags-input').serialize();

        $.ajax({
            type: "POST",
            url: '<?php echo osc_base_url(true); ?>?page=ajax&action=custom&ajaxfile=<?php echo
                osc_plugin_folder(__FILE__) ?>ajax.php?itemid=<?php echo osc_item_id() ?>',
            data: tags,
            success: function(data) {
                $('#tagger-tag-module').find('ul.tagit').css('display', 'none');
                $('#tagger-tag-module').find('#tagger-tags-modify').css('display', '
                    block');
                $('#tagger-tag-module').find('.tagger-tags-list').html( data ).css('
                    display', 'block');
            }
        });
    });
});
</script>

<?php if ( osc_is_web_user_logged_in() && ! osc_is_admin_user_logged_in() ) : ?>
<div class="tags-container">
    <label><?php _e('Tags', 'tagger'); ?>
    &nbsp; &nbsp; &nbsp;
</label>
<?php
    // Prevent clearing of the field if the form is reloaded
    if( Session::newInstance()->.getForm('tagger-field') != '' ) {
        $detail['tagger-field'] = Session::newInstance()->.getForm('tagger-field');
    }
    ?>
    <input id="tags-input" type="text" name="tagger-field" placeholder="Input tags here" value
        ="<?php echo isset($tags) ? implode(', ', $tags) : '' ?>" />
</div>

<?php endif; ?>

<script>
jQuery(document).ready( function($) {

    $('#tags-input').tagit({
        placeholderText : 'Input tags here...',
        availableTags : <?php echo json_encode( $allTags ) ?>
    });

});
</script>

<fieldset>
    <h3><?php _e('Search by tag', 'tagger') ; ?></h3>
    <div class="row one_input">
        <input type="text" id="tagger-search-field" class="input-text ui-autocomplete-input" name
            ="tag" value="<?php echo Params::getParam( 'tag' ) ?>" />
    </div>
</fieldset>

-- Freetag Structure v2.02
--
-- Table structure for table 'freetags'
--
CREATE TABLE IF NOT EXISTS /*TABLE_PREFIX*/t_tagger_tags (
  id INT(10) UNSIGNED NOT NULL AUTO_INCREMENT,
  tag VARCHAR(30) NOT NULL DEFAULT '',
  raw_tag VARCHAR(50) NOT NULL DEFAULT '',
  PRIMARY KEY (id)
) ENGINE=InnoDB DEFAULT CHARACTER SET 'UTF8' COLLATE 'UTF8_GENERAL_CI';

--
-- Table structure for table 'freetagged_objects'
--
CREATE TABLE IF NOT EXISTS /*TABLE_PREFIX*/t_tagger_tagged_objects (
  tag_id int(10) UNSIGNED NOT NULL DEFAULT '0',
  tagger_id int(10) UNSIGNED NOT NULL DEFAULT '0',
  object_id int(10) UNSIGNED NOT NULL DEFAULT '0',
  tagged_on datetime NOT NULL DEFAULT '0000-00-00 00:00:00',
  PRIMARY KEY ('tag_id','tagger_id','object_id'),
  KEY 'tag_id_index' ('tag_id'),
  KEY 'tagger_id_index' ('tagger_id'),
  KEY 'object_id_index' ('object_id')
) ENGINE=InnoDB DEFAULT CHARACTER SET 'UTF8' COLLATE 'UTF8_GENERAL_CI';

<?php
/*
*   OSCClass      software for creating and publishing online classified

```

```

*
*           advertising platforms
*
*           Copyright (C) 2010 OSCLASS
*
*           This program is free software: you can redistribute it and/or
*           modify it under the terms of the GNU Affero General Public License
*           as published by the Free Software Foundation, either version 3 of
*           the License, or (at your option) any later version.
*
*           This program is distributed in the hope that it will be useful, but
*           WITHOUT ANY WARRANTY; without even the implied warranty of
*           MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
*           GNU Affero General Public License for more details.
*
*           You should have received a copy of the GNU Affero General Public
*           License along with this program. If not, see <http://www.gnu.org/licenses/>.
*/

/**
 * Model database for Products tables
 *
 * @package OSClass
 * @subpackage Model
 * @since 3.0
 */

require_once( 'libs/freetag/freetag.class.php' );

class TaggerModel extends DAO
{
    private $model_tables = array( "t_tagger_tags", "t_tagger_tagged_objects" );

    private $freetag;

    /**
     * It references to self object: ModelProducts.
     * It is used as a singleton
     *
     * @access private
     * @since 3.0
     * @var ModelProducts
     */
    private static $instance ;

    /**
     * It creates a new ModelProducts object class ir if it has been created
     * before, it return the previous object
     *
     * @access public
     * @since 3.0
     * @return ModelProducts
     */
    public static function newInstance()
    {
        if( !self::$instance instanceof self ) {
            self::$instance = new self ;
        }
        return self::$instance ;
    }

    /**
     * Construct
     */
    function __construct()
    {
        parent::__construct();

        $freetag_options = array(
            'db_user'      => DB_USER,
            'db_pass'     => DB_PASSWORD,
            'db_host'     => DB_HOST,
            'db_name'     => DB_NAME,
            'table_prefix' => DB_TABLE_PREFIX
        );
        $this->freetag = new freetag( $freetag_options );
    }

    /**
     * Import sql file
     * @param type $file
     */
    public function import($file)
    {
        $path = osc_plugin_resource($file) ;
        $sql = file_get_contents($path);

        if( ! $this->dao->importSQL($sql) ){
            throw new Exception( "Error importSQL::TaggerModel<br>". $file ) ;
        }
    }

    /**
     * Remove data and tables related to the plugin.
     */
    public function uninstall()
    {
        foreach ( $this->model_tables as $stable ) {
            $this->dao->query('DROP TABLE '. DB_TABLE_PREFIX . $stable );
        }
    }
}

```

```

    }
}

public function saveTags( $user_id = NULL , $object_id = NULL, $tags = NULL )
{
    if ( is_null( $tags ) ) {
        return FALSE;
    }

    $tags = explode(' ', $tags);

    foreach ( $tags as $tag ) {
        $tag2 = $this->freetag->normalize_tag($tag);
        $this->freetag->tag_object($user_id, $object_id, $tag2 );
    }

    // return TRUE;
}

/**
 * get_tags_on_object
 *
 * You can use this function to show the tags on an object. Since it supports both user-
 * specific
 * and general modes with the $tagger_id parameter, you can use it twice on a page to make
 * it work
 * similar to upcoming.org and flickr, where the page displays your own tags differently
 * than
 * other users' tags.
 *
 * @param int The unique ID of the object in question.
 * @param int The offset of tags to return.
 * @param int The size of the tagset to return. Use a zero size to get all tags.
 * @param int The unique ID of the person who tagged the object, if user-level tags only are
 * preferred.
 *
 * @return array Returns a PHP array with object elements ordered by object ID. Each element
 * is an associative
 * array with the following elements:
 * - 'tag' => Normalized-form tag
 * - 'raw_tag' => The raw-form tag
 * - 'tagger_id' => The unique ID of the person who tagged the object with this tag.
 */
public function getTags( $object_id )
{
    $tags = $this->freetag->get_tags_on_object( $object_id, 0, 0, NULL );

    return $tags;
}

public function getAllTags()
{
    $this->dao->select( 'tag' );
    $this->dao->from( sprintf("%st_tagger_tags", DB_TABLE_PREFIX ) );

    $result = $this->dao->get();

    if( ! $result ) {
        return array();
    }

    return $result->result();
}

public function removeAllTags( $user_id , $object_id )
{
    return $this->freetag->delete_all_object_tags_for_user( $user_id, $object_id );
}

function array_pluck($toPluck, $arr) {
    $ret = array();

    foreach($arr as $item) {
        if (isset($item[$toPluck])) {
            $ret[] = $item[$toPluck];
        }
    }
    return $ret;
}

/**
function array_pluck ($toPluck, $arr) {
    return array_map(function ($item) use ($toPluck) {
        return $item[$toPluck];
    }, $arr);
}
*/
}

```

## XII. Acknowledgement

I cannot thank the Lord enough for the blessings He has given me. Finishing this special project was only possible because of the strength He has given me. I have received an undying amount of support from the people I've been with, and thoughts and prayers from everyone, and I am truly grateful for that.

I want to thank my parents for being the most patient, and supportive parents anyone could ask for. I want to thank them for always giving me their unconditional love, and prayers, and for always leading me to the right direction.

I want to thank my closest friends from college, Arvi, Carmel, Christine, Eunice, and Kyra, for always being dependable to keep my spirits up, and for always believing in me. And I'd like to thank them for giving me the most fun college experience that I'll always keep in memory. Thank you Tin, for the unimaginable companionship you offered me during the times we were left with each other during our "overstay". And I will surely miss your old house and the times it kept me from missing my own home.

I would also like to thank two of my closest buds, Ashley, and Hazel for being there for me in good times and in bad. Thank you for the company you have shared with me which kept me sane. And thank you for having my back, and for giving me some of the pushes I needed to get myself up and working.

I also would not miss thanking my adviser who adopted me, Ms Avegail Carpio, for being a good adviser, and for being a good person who gave me the chance to finish this project which almost seemed impossible. Thank you for the time you have put aside for me, and thank you for putting up with me Ma'am.

Lastly, I would like to thank Matthew for making me smile as I finished my defense.