

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

MOBI-MAPP: A MOBILE APPLICATION FOR QUICK
ALERT AND MONITORING OF DISASTERS

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

Submitted by:

Neil Patrick U. Ballicud

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 “Mobi-MAPP: A Mobile Application for Quick Alert and Monitoring of Disasters” prepared and submitted by Neil Patrick U. Ballicud in partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science has been examined and is recommended for acceptance.

Ma. Sheila A. Magboo, M.Sc.
Adviser

EXAMINERS:

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

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

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

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

Abstract

Several calamities have already been observed here in the Philippines. With its geographic location, it is not surprising that the country has been considered as one of the most disaster-prone in the world. In times of these disasters, that come unpredictably, alertness and preparedness of the people is important as these could prevent possible damages and save lives.

With the advent of modern technological tools and gadgets such as smartphones, with its advanced computing ability and connectivity, it is not impossible to create a mobile early warning system and quick monitoring for these kinds of disaster.

In this project, we propose a system called Mobi-MAPP, a mobile application for quick alert and monitoring of disasters. Here, Mobi-MAPP users can send their reports on a particular disaster situation through the application. Also, data from different concerned local and national agencies are gathered. These data will then be quickly reflected on the Mobi-MAPPs interactive map which will display different disaster information across the country. Mobi-MAPP also provides an alert system wherein users are notified of nearby disaster. Mobi-MAPP also provides a means in which national disaster risk reduction and management units can share disaster information and activate disaster response to affected areas.

Keywords: Disaster Risk Reduction, Mobile Application, Web Services, GIS

Contents

Acceptance Sheet	i
Abstract	ii
List of Figures	iv
I. Introduction	1
A. Background of the Study	1
B. Statement of the Problem	2
C. Objectives of the Study	3
D. Significance of the Project	5
E. Scope and Limitations	6
F. Assumptions	6
II. Review of Related Literature	8
III. Theoretical Framework	12
A. Disaster	12
B. Disaster Risk Reduction	12
C. National Disaster Risk Reduction and Management Council (NDR- RMC)	13
D. Smartphone and the Android Platform	15
E. Geographic Information System (GIS)	15
F. Social Media	16
G. Web Services	16
H. Google Play Services	17
IV. Design and Implementation	18
A. The Mobi-MAPP Architecture	18
B. Entity-Relationship Diagram (ERD)	20
C. Use-Case Diagram	29

V.	Architecture	34
A.	Technical Architecture	34
VI.	Results	36
A.	The Mobile Application	36
B.	The NDRRMC Section	44
C.	Mobi-MAPP Administrators	50
VII.	Discussions	55
VIII.	Conclusions	58
IX.	Recommendations	59
X.	Bibliography	61
XI.	Appendix	64
A.	Source Code	64
XII.	Acknowledgement	183
XIII.	Bibliography	185

List of Figures

1	Mobi-Mapp home Activity Screen	36
2	Icons used to represent each type of disaster	37
3	Info Window	37
4	Knowing your current location	38
5	Search Places	38
6	Mobi-Mapp Login Activity Screen	39
7	Mobi-Mapp Register Activity Screen	40
8	Activate Account	41
9	Activation Email	41
10	Disaster Report Form	42
11	Disaster Autocomplete Location	42
12	Disaster Spinner for type	43
13	Optional fields	43
14	Signin Page	44
15	NDRRMC reports	45
16	CALABARZON RDRRMC reports	45
17	Validation of reports	46
18	Active disasters	46
19	View to Update or End Disaster	47
20	Additon/Update of Disaster	47
21	Disaster Map	48
22	Recent/Past Disasters	48
23	Communication between NDRRMC Units	49
24	Communication between NDRRMC Units	49
25	Assign Personnel	50
26	See Operations	50
27	Register NDRRMC unit officer	51
28	See all officers in the unit	51

29	See all registered units	52
30	Register Unit	52
31	Register Unit	53
32	See all mobile users	53
33	View mobile users and ban/unban	54

I. Introduction

A. Background of the Study

Over the years, several calamities have been observed in the Philippines, whose geographic location makes the country prone to disasters such as earthquakes, volcanic eruptions, typhoons which result to floods, etc. In fact, the World Disaster Report 2012 ranked the Philippines as the third most disaster-prone country in the world, next to pacific countries, Tonga and Vanuatu [1]. It is very evident with the recent Habagat floods, tectonic movements across Philippine plates, Mt. Mayon eruptions, which have caused several damages to properties and lives.

In the event of these disasters, preventive measures like early warning and monitoring is necessary. One way by which this can be done is by taking advantage of the increase in smartphone usage [2] in the country. Aside from being portable, smartphones have advance computing ability, which allows creating useful applications such as those that can be related to disaster management. Mobile phones can be used for vigilance and reporting incidence of disasters or damages by sending related pictures and details. Also, locations can be tracked through GPS (Global Positioning System) that could be embedded to the phone or made available by the Network Service Provider [3]. With these, it is also possible to create an interactive map that allows locating disaster situation and depicting its corresponding information.

Filipinos, being considered as worlds top social networking users [4], make this social media as means for interaction and can also serve as a source of significant up-to-date data and communication about disaster events.

Government efforts on these situations are also very much considered. The National Disaster Risk Reduction and Management Council (NDRRMC), that is

in-charge of disaster management in the country, has a large scope running from the national scale to regional (RDRRMC) to provincial (PDRRMC) down to the smallest local government units, Municipal and City DRRMC to Barangay Development Councils.

DOST PAGASA has also introduced Project Noah [5], a web and mobile application that enables users to assess nationwide hazards such as rains, typhoons, conditions of dams, etc. for prevention and mitigation with regards to disaster situations.

With the still unpredictable forces of nature, innovations on disaster risk reduction also push through. And with the continuous advancement of technology, creating modern schemes to provide citizens awareness on disaster situations can be a significant move.

B. Statement of the Problem

During the occurrence of disasters, people are often unaware of the situation or there is a lack of early warning alert that could prompt them for an immediate response. Also, data about these incidents which can be acquired from concerned groups and agencies are not consolidated into one place which people can look into.

It is difficult for the people to know the locations of the present disaster. Many citizens are only dependent in what media tell or news from local disaster management bulletin may not reach them right away. Early warning alerts can bring people awareness and preparedness for these serious situations that can cause major damage to their lives.

Sources of disaster information may come from different places. People are informed through media (either from television or radio broadcasts), advisories from

local and national agencies or from across the internet (e.g. feeds from government or news sites). However, with the data coming from all these mediums, people may find it difficult to search where information needed could be obtained.

C. Objectives of the Study

General Objective:

The main goal of the system is to provide the application users an interface in which they can send disaster reports, search and locate disaster information through a user-friendly map and to be notified in case of nearby disaster situations. The system also provides interface for the National, Regional, Provincial, and Local DRRMCs (City and Municipal) for information relaying and dispatching of teams that will respond to a disaster event and a moderator, for filtering incoming mobile reports and social media feeds for encoding needed data to the application.

Specific Objective:

This system operates with the following functionalities:

1. Allows the mobile users to
 - (a) Register to enable uploading disaster information
 - (b) Log-in their accounts
 - (c) Send reports through their mobile phones
 - (d) Receive notifications
 - i. Acknowledge of account acceptance
 - ii. Deactivation of account
 - iii. Alert for nearby disaster
 - iv. Nearest route to safe shelter

- (e) Check the applications disaster map for nearby situations or quick information
 - i. Input the desired location
 - ii. See hazard dots located on the map
 - iii. Click those dots to see hazard information
- (f) Deactivate account
- (g) Logout account.

2. Allows LDRRMC (Local Disaster Risk Reduction and Management Council)

- (a) Log-in through their account
- (b) Receive disaster reports
- (c) Validate disaster reports
- (d) Add/View/Edit/Delete disaster data and nearest shelter to main server
- (e) Check disaster map
- (f) Activate Operations Teams (DRRMC Heads only)
- (g) Send report to mother NDRRMC unit
- (h) Log-out their accounts

3. Allows PDRRMC (Provincial Disaster Risk Reduction and Management Council), RDRRMC (Regional Disaster Risk Reduction and Management Council), NDRRMC (National Disaster Risk Reduction and Management Council) to

- (a) Log-in through their account
- (b) View disaster reports
- (c) View disaster data and nearest shelter
- (d) Check disaster map
- (e) Activate Operations Teams (DRRMC Heads only)

- (f) Send and receive report to and from mother and child NDRRMC units
 - (g) Log-out their accounts
4. Allows system to access social media feeds related to disaster situations
5. Allows system administrators to:
- (a) NDRRMC unit administrators
 - i. Log-in with his/her account
 - ii. Perform user management for their respective unit personnel
 - iii. Log-out his/her account
 - (b) Overall administrator
 - i. Log-in with his/her account
 - ii. Perform user management for DRRMC administrators, moderators and mobile reporter
 - iii. Ban mobile reporters w/ bad reputation
 - iv. Unban mobile reporters.
 - v. Log-out his/her account

D. Significance of the Project

This project lets the application users to be aware of disaster situations across the country by providing them a tool in which they can monitor disaster situations. Mobi-MAPP will allow them to conveniently search for a location they wish to check for possible disasters.

The system will also provide helpful alerts for those whose current location resides near an occurring calamity. Mobi-MAPP will also provide a means for disaster management units of the government for quick response to affected areas.

The system will also allow vigilance of mobile application users in times of disasters. Users may send in their reports through the mobile application which can help for disseminating this information to the possible users also of the application. Here, multiple sources of data are utilized so that it can provide beneficial information that could trigger their awareness in times of these kinds of situations.

E. Scope and Limitations

1. This system only includes natural disasters such as earthquake, flood, landslide, volcanic eruptions, fire, and tsunami.
2. Other forms of disasters like rebellion, and other man-made disasters are not included.
3. Mobile users should have smartphones with Android only as their operating system.
4. Only mobile users who are registered and signed in and not banned, can send reports.
5. Mobile users dont need to sign in to view disaster map.
6. NDRRMC units must register and login first to be able to use Mobi-MAPP.
7. Only LDRRMC (Local NDRRMC units) can validate report, add, update, and end disasters.
8. System will be available to the Philippines only.
9. Only mobile users who downloaded and installed Mobi-MAPP and are connected to internet can use the application.

F. Assumptions

This system works with the following assumptions:

1. NDRRMC units are always available to receive and validate reports

2. Agencies agreed to also share information into the application

II. Review of Related Literature

In the course of disasters, happening unpredictably, alertness and awareness of the people are essential in order to prepare and reduce risk and possible damages these events can bring. Different techniques and technological tools have been developed in addressing these needs in disaster management and some of which have found to be useful in efficient and timely manner of handling these situations.

Geographic Information System (GIS) has played a significant role in terms of disaster management. GIS is widely used because of its ability of capture data in geographic format and display information in an interactive map. [6] Developing GIS to study and evaluate different disaster-prone areas in China becomes an essential part in disaster risk assessments. In Xiamen City, researchers used GIS to create a map that will show the vulnerability of the city to natural disasters. Vulnerability of the city was calculated through certain geographic data factors such as natural environment, population, economy, urban morphology etc. [7] In Guangzhou, one of the most flood-prone cities in China, GIS was used to express spatial relationships that could determine the risk or vulnerability of certain areas and districts in Guangzhou, a flood hazard evaluation map was developed through the help of a GIS. [8]

Here in the Philippines, GIS is also used to monitor different disaster-prone areas. HEC RAS model [9] presents how the system could provide a near-real time condition of the Marikina River. Near-real time water levels are monitored and near-real time flood extent map is shown in the Project NOAH application.

In Web-Based GIS System for Real-Time Field Data Collection Using a Personal Mobile Phone[10], it has showed how mobile phones have been a great use for real-time field data collection. Field data collection is an essential step for researches in conducting studies on particular place. Traditional pen-and-paper

based methods consume a lot of time in this practice with comparison to modern mobile phones that can capture images, locate themselves through GPS, and send important messages through SMS/MMS. These messages can also be obtained and convert to useful data that can be stored in a database and reflect those into the GIS map. With these people like researchers can easily record field data and locate them with the use of the digital map.

Crowdsourcing, in times of disasters, also plays a big role in sharing information about these events and in reporting details about damages on affected areas. The crowd, with the use of their mobile phones can easily capture important events and share them to the public with the use of internet [11]. In *Harnessing the Crowdsourcing Power of Social Media for Disaster Relief*, advantages of crowdsourcing especially on mobile reports and through social media includes timely collection of requests and status reports, collection of data from different sources such as emails, tweets and microblogs, and inclusion of geo-tags in every reports sent [12]. For instance, in Alabama, USA, where harmful tornadoes frequently hit the state, mined crowd-sourced information posted on a public photo sharing site Flickr was found to be useful to help responders and to locate damages. [13] Damages must be identified as early as possible so that possible damages can also be prevented. Damage Tracker [14] used crowd sourcing to report damages brought by disasters, particularly tornadoes, through a mobile application which can capture and upload geo-tagged photos of the said damages. It also used cloud storage and a web application that stores uploaded photos and displays in an interactive map, in which users can view in details the current happening or the damages itself.

In Bangladesh, whose geographic location is also prone to natural disasters, researchers have proposed a location-based early warning and evacuation system for mobile phones [3]. Here, Locations of users are known through GPS and network location of their mobile phones that are sent to the main server. Disaster

and evacuation data are updated in the main server by weather offices and other local authorities. Data that are stored in the server are then used for informing citizens who will be mainly affected by the disaster.

The said system used Ray-casting algorithm to be able to identify affected citizens covered by the disaster. If a user is identified as a probable victim of the disaster, the user will get automatic notifications. Notifications will provide visual and audio warnings. A map and a shortest path to the nearest evacuation center are also provided using the OpenStreetMap (OSM) [15] and CloudMade [16] technology, respectively. The system used the Android Cloud to Device Messaging for disseminating automatic notifications. When a user is unable to go to the safe shelter, it is submitted to the server and the system will now contact rescue teams.

The success of the said system is that it could help normal citizens as well as tourists, whom maybe are unfamiliar with some places, and also, can help blind people, with the help of audio notifications. Also, it also helps the authority to identify people in need of rescuing. The main challenge and limitation of the system is the lack of map details of Bangladesh provided in the OSM.

Here in the Philippines, DOSTs Project NOAH, a system integrating its current science and research development projects to provide hazard maps that could let people be aware of nationwide hazards, particularly on weather situation, dams and basins state, tide levels in coastal areas, that could cause disastrous events like landslide, flash floods, etc. Flood reports are also reflected on the map showing details about flood incidents. Flood maps for a particular range of years of some areas are also present. Users can opt to check different weather status in some areas covered by Doppler radars and weather stations. [5]

In the special problem Mobile Disaster Communication System, it showed how

government agencies from office of the President down to Local Government Units (LGUs) can rapidly communicate, through mobile phones especially through Short Message Services (SMS), in times of disaster [17]. Also, in the special problem Mobile Disaster Reporting System, it showed how mobile phones can report incidents of disaster and can quickly request emergency response. These requests are sent to the web application for validation and for quick dispatching of rescuers [18].

In this project, different tools and techniques have seen to be useful in times of disaster situations. Data from concerned authorities can be consolidated into data storage. Geographic Information System (GIS) can help analyze and display spatial data, such as disaster information, into functional maps. Also, with the rapid growth of number of smartphone users, first-hand disaster information, such as damages, can be reported. Integrating tools, like GIS and other mapping technologies, into these mobile phones could also inform and alert mobile users of disaster situation around.

III. Theoretical Framework

A. Disaster

Disaster is defined as something (such as a flood, tornado, fire, plane crash, etc.) that happens suddenly and causes much suffering or loss to many people [19]. It is a sudden, calamitous event that seriously disrupts the functioning of a community or society and causes human, material, and economic or environmental losses that exceed the community's or society's ability to cope using its own resources. Though often caused by nature, disasters can have human origins [20]

In the context of this project, the system classifies disaster as earthquakes, floods, tornadoes, tsunamis, landslides, volcanic eruptions. The system only included natural forms of disasters.

B. Disaster Risk Reduction

Disaster risk reduction is the concept and practice of reducing disaster risks through systematic efforts to analyze and reduce the causal factors of disasters. Reducing exposure to hazards, lessening vulnerability of people and property, wise management of land and the environment, and improving preparedness and early warning for adverse events are all examples of disaster risk reduction. [21]

In this project, it aims to provide disaster risk reduction with the use of the system. Here, the system uses techniques such as gathering disaster data from people and informing citizens about occurring disasters. With these, the project lets users to be aware of disasters happening, and in turn reduce risks.

C. National Disaster Risk Reduction and Management Council (NDRRMC)

NDRRMC, formerly known as the NDCC (National Disaster Coordinating Council), is the primary agency of the government under the Department of National Defense that mainly handles the protection and welfare of citizens with regards to disasters.[22]

Republic Act (R.A.) 10121 deals with strengthening the Philippines disaster risk reduction providing a framework and institutionalizing a disaster risk reduction and management plan. Its scope includes risk assessment and early warning, knowledge building and awareness raising, preparedness for early response and quick recovery, and all aspects of disaster risk reduction and management. [23]

NDRRMC mainly consider four major aspects in disaster management:

1. Prevention and Mitigation, that focuses on reducing vulnerabilities and exposure to potential impacts
2. Preparedness, to establish citizens anticipation and easy recovery from possible negative impacts
3. Response, that provides life preservation and handling of basic needs of affected population and
4. Rehabilitation and recovery, which restore and improve facilities and ensuring living and livelihood to affected community.

Structure of the NDRRMC

Sections of the RA 10121 states NDRRMC, replacing the old NDCC, with the Secretary of National Defense as the chairperson of the council. He, as the chairperson, may call upon other entities of the government and non-government sectors for assistance in DRRM. Its vice-chairpersons include the Depart of Science

and Technology (DOST) Secretary, Department of Interior and Local Government (DILG) Secretary, Department of Social Welfare and Development (DSWD) Secretary and the Nation Economic Development Authority (NEDA) General, as vice-chairs for the 4 major aspects respectively stated above. NDRRMC members also include heads of different agencies of the government.

On the Regional Level, the RDRRMC (Regional DRRM Council), headed by regional directors of the Office of Civil Defense and composed of regional directors and executives of other government agencies, shall coordinate, integrate, supervise and evaluate activities of Local DRRM units.

On the local level, Local DRRM Office (LDRRMO) include Provincial DRRMC, under provincial goes City and Municipal DRRMCs. Under City and Municipal DRRMCs include Barangay Development Council (BDC). Like the higher levels, LDRRMO supervises, evaluates and coordinates with lower level councils responsible for the direction and development of DRRM programs within territorial jurisdiction. LDRRMO, headed by LGU chief executive, is composed of three staff responsible for: 1) Administration and training, 2) Research and planning, 3) Operations and Warning.

Government agencies and LGUs may mobilize individuals and organized volunteers to participate in disaster risk reduction and management programs and activities. Roster of accredited volunteers are recorded and maintained through LDRRMO. Accreditation is done at the city or municipal level, in which, LGUs take full responsibility for the welfare and protection in the mobilization of volunteers.

Coordination during emergencies

The following shall take the lead in managing emergency disaster situations:

1. The BDC, if a barangay is affected
2. The city of municipal DRRMC, if two or more barangays are affected
3. The Provincial DRRMC, if two or more cities or municipalities are affected
4. The Regional DRRMC, if two or more provinces are affected
5. The NDRRMC, if two or more regions are affected

The NDRRMC and intermediary LDRRMCs shall always act as support for the LGUs as the primary responders. Other sectors shall work with coordination with NDRRMC according to its policies.

D. Smartphone and the Android Platform

Smartphone is defined as a category of mobile device that provides advanced capabilities beyond a typical mobile phone. Smartphones run complete operating system (OS) software that provides a standardized interface and platform for application developers.[24] Smartphones, in compared to other phones, have advanced computing ability and connectivity.

Android is an operating system for smartphones created by Google that is based on Linux. Android is open-source and has a large community of developers extending various functionalities, written primarily in modified Java programming language.

E. Geographic Information System (GIS)

GIS is a system that stores, retrieves, analyze, and manage geographic data. It is a special-purpose information system wherein geolocation and spatial data are the primary means. Information retrieved or analyzed usually comes from digital maps. It is widely used in spatial analysis and statistics that help in decision-making in particular field. GIS integrates technologies such as creating statistical

models, analysis of aerial and satellite images, drafting of maps. [25]

In this project, GIS is used to integrate disaster data to the disaster map, retrieve geographic location of the users and to establish paths from user locations and nearest safe shelter or evacuation center.

F. Social Media

Social media is defined as means among people and users to interact to each other within the web. Social media includes Facebook, Twitter, and others. Social media has not only played a role in peoples social interaction in their daily lives but also can be used in disaster management.

Social media such as Twitter is proven to help in times of disaster events. Twitter provides users to broadcast short messages that can provide reports with structural elements such as time, location, tags or categories which can be helpful in times of disaster situations. [11]

In this project, we include Twitter as a possible source of disaster data which will then be forwarded to the main users of the system.

G. Web Services

A web service is generally defined as communication between two applications within the World Wide Web. Providers of these web services are also called application service provider in which these providers share data to other applications. [26]

H. Google Play Services

This provides android applications to add attractive and useful features to them.

These features include Maps, Directions, Locations, etc. [27]

1. Google Maps API

Google Maps is a web mapping service application offered by Google that provides digital and interactive maps. Users of the application are able to search and locate places, view information with those places, zoom in and zoom out of the map, and many more. [28]

Google Maps API lets developers integrate their application with the Google Maps, augment different data to these maps, create markers to indicate special points and allow their user to explore the map with built functions such as zoom, rotate, tilt, search, etc. [29]

2. Google Locations Services API

This API is a powerful tool in handling location of android users. This also includes automatic handling of user location and their accuracy. [30]

3. Google Cloud Messaging API

GCM is a service provided by Google that allows an application to send data from the applications server to an Android device. Also, GCM provides sending of messages from an applications server to a target Android device, by which GCM also handles queuing and delivery of these messages to specified targets. [31]

IV. Design and Implementation

Mobi-MAPP has 3 major group of users: 1) The Mobile App User, 2) The NDR-RMC officers and 3) The NDRRMC Administrators.

The mobile application users are the major end-users of the system. They check the map for disaster situations across Philippine map and are alerted in case of a disaster. Also, mobile application user can send mobile reports through the application into the system.

NDRRMC officers are the ones responsible for validating disaster reports. Also, they are the ones that feeds the server of the current disaster situations in the country. They also handle monitoring of these events by sending relief and rescue operation teams to respond their affected areas. They can also communicate to other units for information sharing and cooperation.

Only LDRRMC units can validate disaster reports. Also, they are only the ones who can add new disaster data and end an active disaster.

Lastly, the administrators, theyre the ones who check availability of the system. They are also in charge of user management, creating, updating, and deleting of the accounts of the users. The overall administrator of the system, on the other hand, manages disaster units, register them and its respective administrator. He is also in charge in performing user management of mobile app users.

A. The Mobi-MAPP Architecture

Here, we take a look on how the system works in general. Figure 1 shows actual setup of the system.

Figure 1 (a) shows the mobile reporting section. Here, the mobile application users send reports of disaster events with the use of the mobile application (via SMS or the internet). For instance, a Mobi-MAPP user found that there is a waist-high flood in a certain place, he can take a photo of the incident, fill-up disaster data fields, and send to the following section.

Figure 1 (b) shows the DRRM (Disaster Risk Reduction and Management) section. This section accounts to how NDRRMC units respond to disaster situations. They input disaster data to the main server according to the situations happening in their respective areas. They also receive reports for validation from the moderator to check if a report is consistent. Here, data from web services available are also fed to the main server.

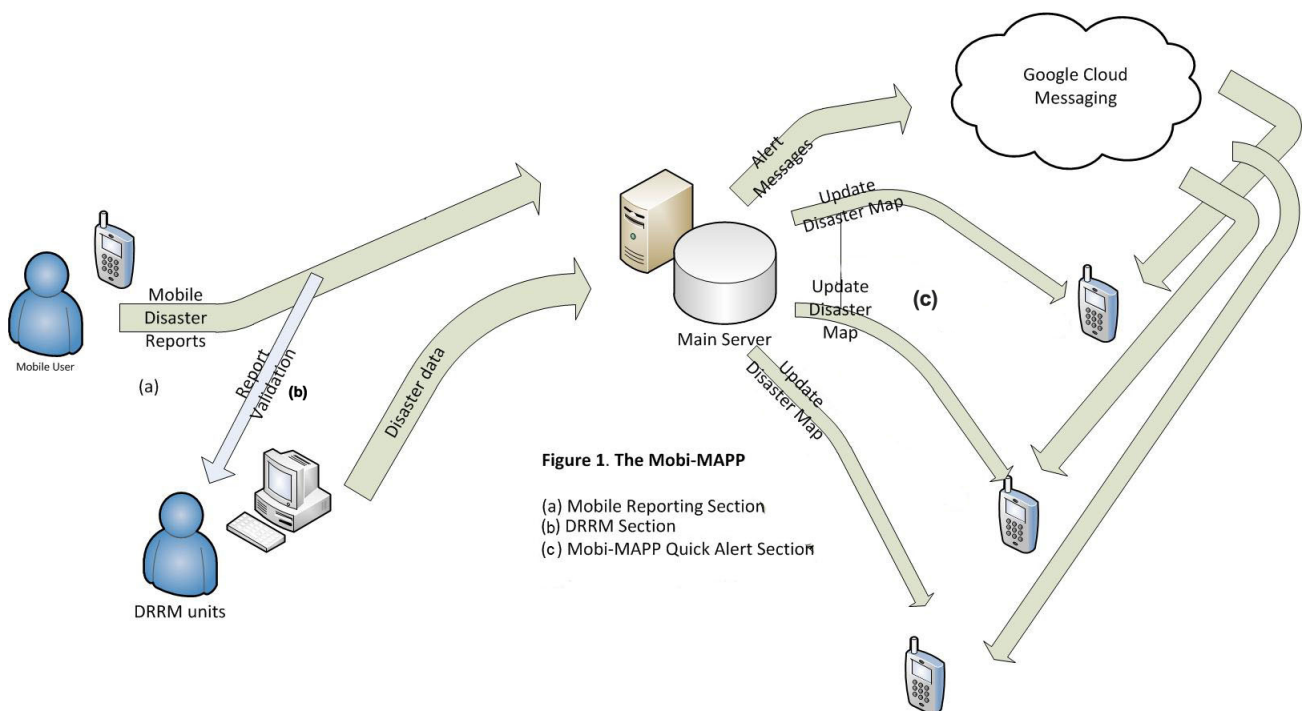


Figure 1 (c) shows the Mobi-MAPP Quick Alert section. Here, data from the data server are analyzed and send directly to the mobile application users. These processed data will now be reflected on the interactive map of the system. Users can search for a location, zoom in and zoom out, and clicks disaster balloons/i-cons that will show them information about the disaster. Also, geolocations of

the mobile application users are regularly sent in the system and analyzed. This section also comes in when a mobile user is detected to be near a particular disaster. Locations of the users are being tracked within a threshold time and these locations are being sent to the main server. When this location is detected to be near a particular disaster, it will send an alert message to the mobile.

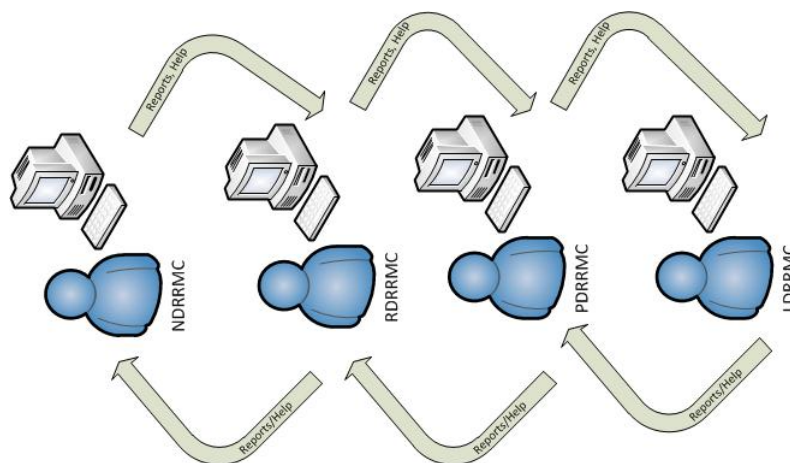


Figure 2. DRRM Section

The DRRMC Section

DRRMC section accounts to the process by which disaster information is communicated among NDRRMC units. Figure 2 shows structure of the Mobi-MAPP communication system for the NDRRMC units. NDRRMC unit may report or send disaster information, ask for help to their mother unit. Likewise, NDRRMC unit may report assign help or task to their child unit.

B. Entity-Redlationship Diagram (ERD)

Data are stored in the main server and stored in a relational database management system. The following diagram shows the entity-relationship model in the system.

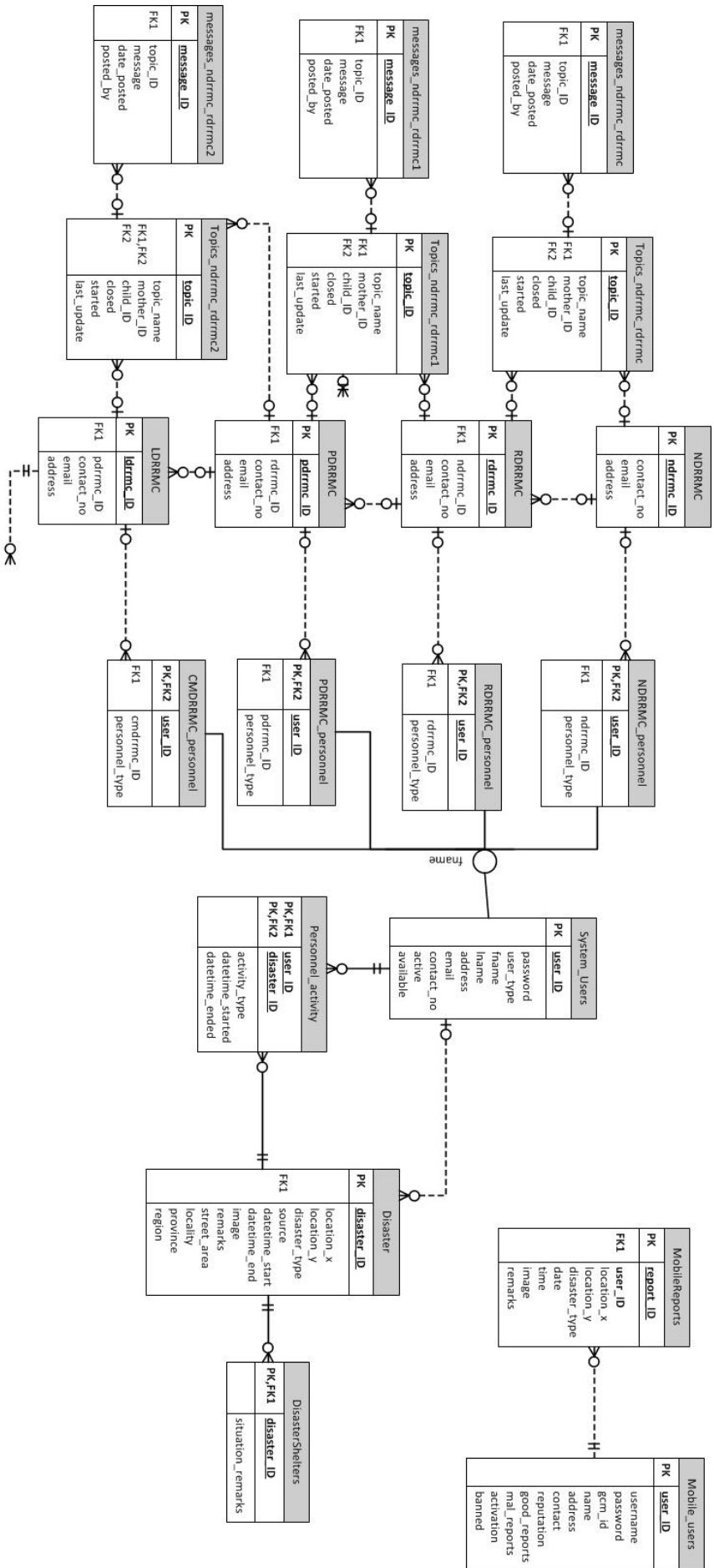


Figure 3. ERD of Mobi-Mapp

Figure 3 shows the ERD for the Mobi-MAPP. As said earlier, the system has four general types of users: the mobile application users, the DRRMC personnel (NDRRMC, RDRRMC, PDRRMC, LDRRMC) and the administrators. Here, the `user_type` determines the user type of the system. Mobile reporter sends many report and a report is only accounted to a single reporter.

NDRRMC handles different RDRRMCs for the different regions in the country. Likewise, A RDRRMC handles PDRRMCs for each province in the region. Also, PDRRMC has different LDRRMCs. Each agency handles an administrator and personnel that may include rescuers, volunteers, officers, encoders, etc that is specified in the `personnel_type` field.

A certain user may be involved in different disaster situation. Likewise, a disaster is handled by different personnel. These activities are recorded with date and time started and ended.

A disaster can have many forms each with different attributes. A disaster can be a flood, landslide, earthquake, tsunami, fire or volcanic eruption. Also, shelters are initially located before a disaster. In a disaster event, a disaster can have many safe shelters.

Topics and Messages Tables refer to the communication between NDRRMC units.

Data Dictionary

Table 1. System_Users

Field	Data Type	Description	KEY
user_ID	INT AUTO_INCREMENT	unique ID for a user	PRIMARY
fname	varchar(50)	user firstname	
lname	varchar(50)	user lastname	
password	varchar(32)	password of user for login	
user_type	enum(ndrrmc,pdrmc, rdrmc,lrrmc,ov_admin)	type of user	
address	varchar(100)	address of the user	
email	varchar(50)	email address of the user	
contact_no.	varchar(20)	contact number of the user	
available	enum(‘yes;no’)	if user is available	

Table 2. NDRRMC

Field	Data Type	Description	KEY
ndrrmc_ID	int	unique ID of NDRRMC	Primary
contact_no	varchar(20)	contact no of the agency	
email	varchar(30)	email of the agency	
address	varchar(70)	address of the agency	
registered	tinyint(1)	if agency is registered	

Table 3. RDRRMC

Field	Data Type	Description	KEY
rdrmmc_ID	int	unique ID of RDRRMC	PRIMARY
ndrrmc_ID	int	ID of mother NDRRMC	FOREIGN
contact_no	varchar(20)	contact no of the agency	
email	varchar(30)	email of the agency	
address	varchar(70)	address of the agency	
region	varchar(70)	region of the agency	
registered	tinyint(1)	if agency is registered	

Table 4. PDRRMC

Field	Data Type	Description	KEY
pdrmmc_ID	int	unique ID of PDRRMC	PRIMARY
rdrmmc_ID	int	ID of mother RDRRMC	FOREIGN
contact_no	varchar(20)	contact no of the agency	
email	varchar(30)	email of the agency	
address	varchar(70)	address of the agency	
region	varchar(70)	region of the agency	
province	varchar(70)	province of the agency	
registered	tinyint(1)	if agency is registered	

Table 5. LDRRMC

Field	Data Type	Description	KEY
ldrrmc_ID	int AUTO_INCREMENT	unique ID of LDRRMC	PRIMARY
pdrmmc_ID	int	ID of mother PDRRMC	FOREIGN
contact_no	varchar(20)	contact no of the agency	
email	varchar(30)	email of the agency	
address	varchar(70)	address of the agency	
region	varchar(70)	region of the agency	
province	varchar(70)	province of the agency	
locality	varchar(70)	locality of the agency	
registered	tinyint(1)	if agency is registered	

Table 6. NDRRMC_personnel

Field	Data Type	Description	KEY
user_ID	varchar(20)	unique ID for a user	PRIMARY
ndrrmc_ID	int	ID of the NDRRMC	Foreign
personnel_type	enum(admin,rescuer,officer,other)	personnel type	

Table 7. RDRRMC_personnel

Field	Data Type	Description	KEY
user_ID	varchar(20)	unique ID for a user	PRIMARY
rdrmmc_ID	int	ID of the RDRRMC	Foreign
personnel_type	enum(admin,rescuer,officer,other)	personnel type	

Table 8. PDRRMC_personnel

Field	Data Type	Description	KEY
user_ID	varchar(20)	unique ID for a user	PRIMARY
pdrmmc_ID	int	ID of the PDRRMC	Foreign
personnel_type	enum(admin,rescuer,officer,other)	personnel type	

Table 9. LDRRMC_personnel

Field	Data Type	Description	KEY
user_ID	varchar(20)	unique ID for a user	PRIMARY
ldrrmc_ID	int	ID of the LDRRMC	Foreign
personnel_type	enum(admin,rescuer,officer,other)	personnel type	

Table 10. Peronnel_activity

Field	Data Type	Description	KEY
user_ID	varchar(20)	ID for a user	PRIMARY
disaster_ID	int	ID for the disaster	PRIMARY
activity_type	enum(relief,rescue,other)	type of activity	
datetime_started	datetime	start datetime of activity	
datetime_ended	datetime	end datetime of activity	

Table 11. MobileReports

Field	Data Type	Description	KEY
report_ID	int	auto-incremental id for a report	PRIMARY
user_ID	varchar(20)	unique ID for a user	FOREIGN
location_x	double	x-coordinate location of the report	
location_y	double	y-coordinate location of the report	
disaster_type	enum(flood,earthquake,landslide, tsunami,fire,volc_activity)	type of disaster	
date	date	date of the report	
time	time	time of the report	
img_url	varchar(127)	image url of the disaster	
remarks	text	details and comment about the report	

Table 12. Mobile_Users

Field	Data Type	Description	KEY
user_ID	INT AUTO.INCREMENT	unique ID for a user	PRIMARY
username	varchar(50)	account username	
name	varchar(50)	user fullname	
password	varchar(32)	password of user for login	
address	varchar(100)	address of the user	
email	varchar(50)	email address of the user	
contact_no.	varchar(20)	contact number of the user	
banned	enum(ýes,ńo)	if user is banned	
good_reports	int	valid reports sent	
bad_reports	int	bad reports sent	
reputation	int	user's reputation in the system	

Table 13. Disaster

Field	Data Type	Description	KEY
disaster_ID	int	auto-increment; unique identifier for disaster	PRIMARY
location_x	double	location X of a disaster	
location_y	double	location Y of a disaster	
disaster_type	enum(flood,earthquake,fire,landslide,tsunami,volc_activity)	type of the disaster	
source	varchar(20)	source of a disaster	FOREIGN
datetime_start	datetime	start datetime of a disaster	
datetime_end	datetime	end datetime of a disaster	
image_url	varchar(127)	image url of a disaster	
remarks	text	other description of a disaster	
street_area	varchar(30)	street or area of a disaster	
locality	varchar(30)	city or municipality of a disaster	
province	varchar(30)	province of a disaster	
region	varchar(30)	region of a disaster	

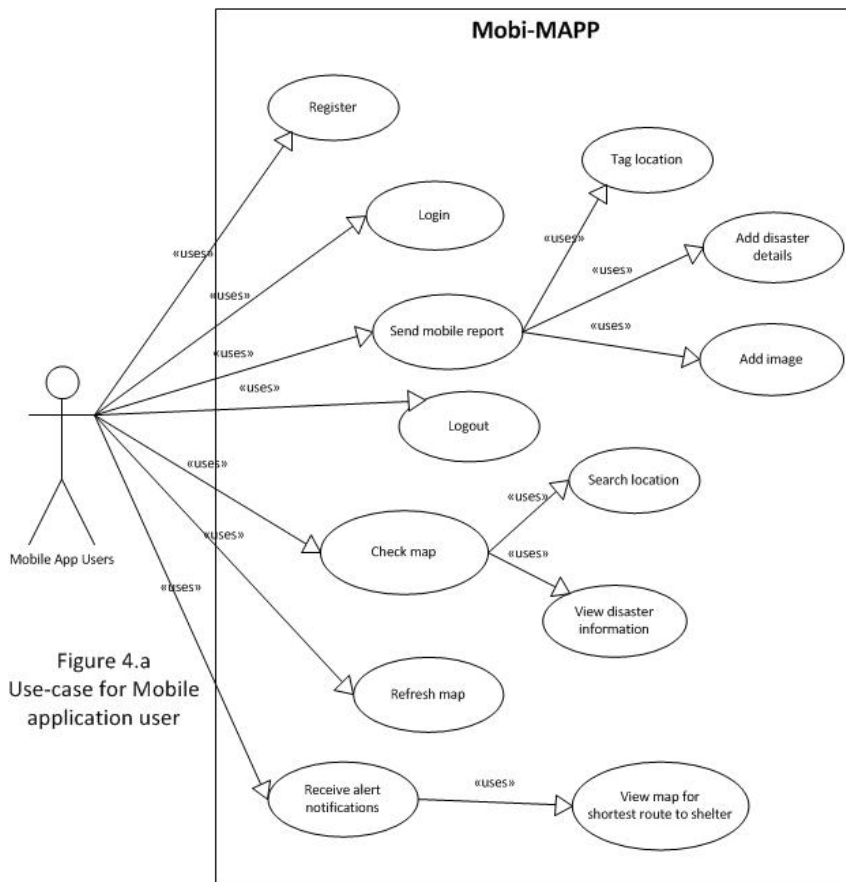
Table 14. DisasterShelters

Field	Data Type	Description	KEY
shelter_ID	int	auto-increment; unique ID for a shelter	PRIMARY
disaster_ID	int	ID of the disaster	FOREIGN
street_area	varchar(30)	Varchar(30); street or area of ashelter	
city_municipality	varchar(30)	Varchar(30); city or municipality of a sheleter	
province	varchar(30)	Varchar(30); province of a shelter	
coord_x	double	Double; location X of a shelter	
coord_y	double	Double; location Y of a shelter	

C. Use-Case Diagram

Use-case for mobile application user

Mobile application user can send mobile reports. Here, the user tags a particular location and write down details about it and add image before sending these data. Mobile users also check the disaster map. He can search a particular location and view information about a certain disaster. User can also refresh map to ensure data are being displayed properly. See Figure 4.a



Use case for LDRRMC officers

Figure 4.c shows the use-case for the LDRRMC officers. He/she logs in. They validate reports. They manage disaster data in which they add disaster, update disasters, and end disasters. Also, they report or ask help from their mother DR-RMCs in case of disaster situations in their area. The head of the DRRMC can activate rescue teams. The officer logouts.

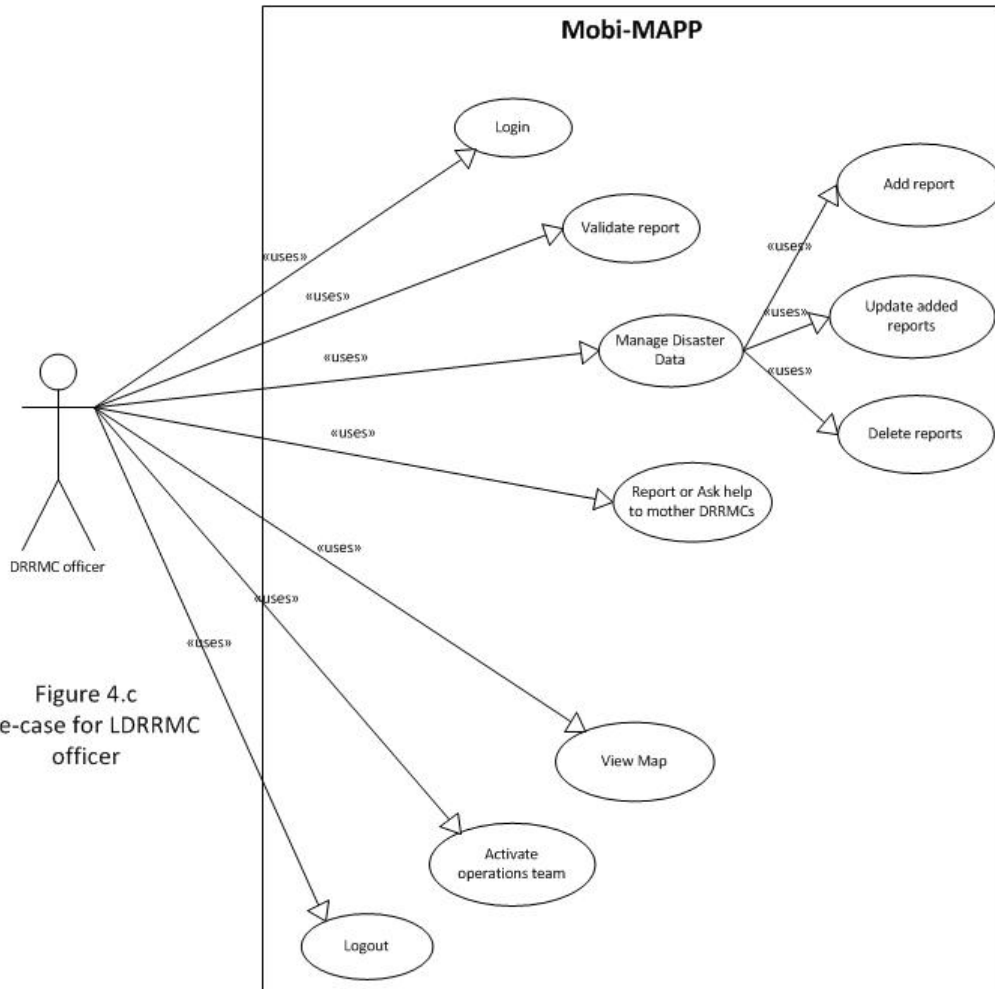


Figure 4.c
Use-case for LDRRMC officer

Use case for N/R/PDRRMC officers

Figure 4.c shows the use-case for the national, regional and provincial NDR-RMC officers. He/she logs in. They receive reports sent to them by child DRRMC units. They can view reports and disasters and check the disaster map. The head of the DRRMC can activate rescue teams. The officer logouts.

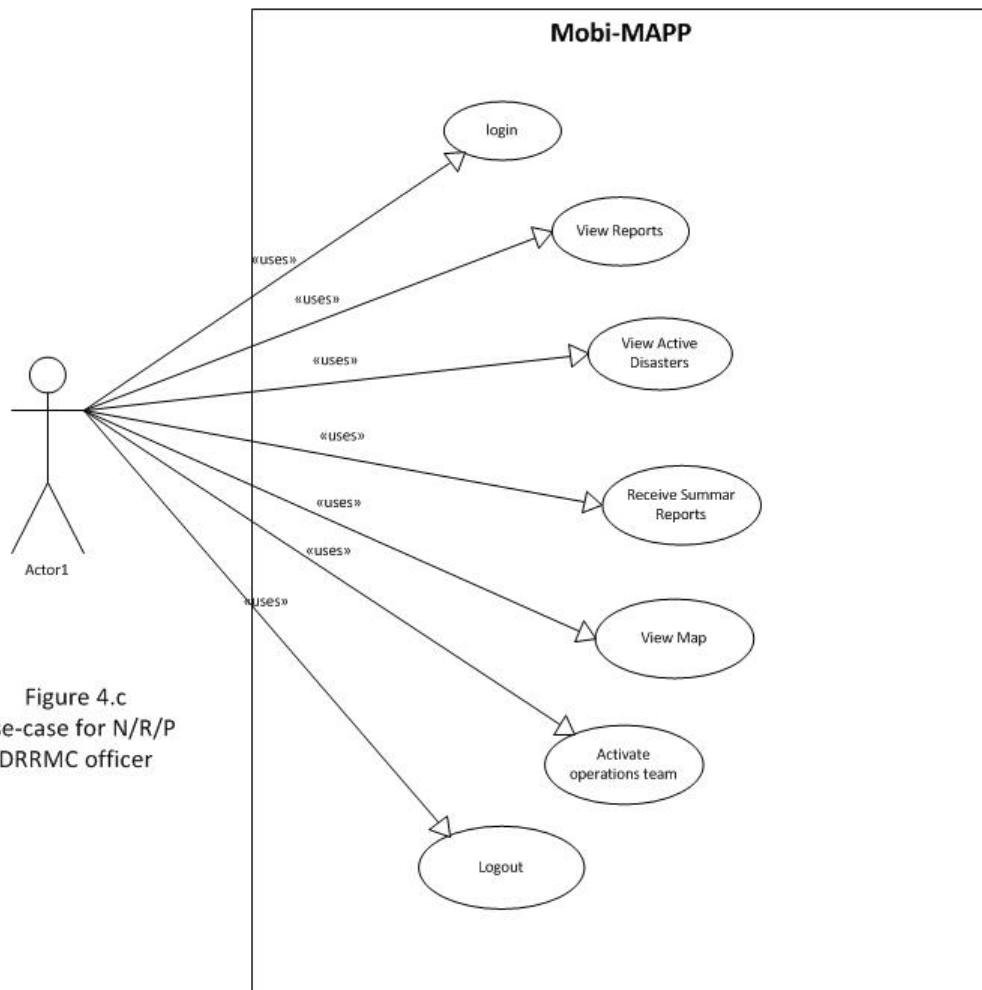


Figure 4.c
Use-case for N/R/P
DRRMC officer

Use case for administrators

1. DRRMC administrators

Figure 4.d shows the use case for the DRRMC administrators. The administrator logs in, view accounts for the personnel in the council he is involved, manage them which includes adding, updating and deleting accounts. Administrator logouts his/her account.

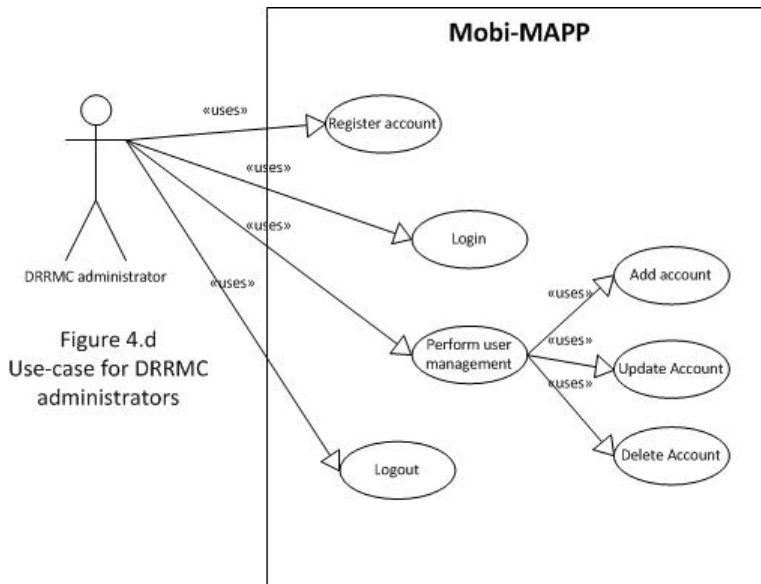


Figure 4.d
Use-case for DRRMC administrators

2. Overall administrators

Figure 4.e shows the use case for overall administrator. The admin logs in and views accounts of users particularly DRRMC administrators, mobile reporters. He/she manages user accounts in which they add, edit and delete accounts. He/she can also ban and unban mobile reporters whom added abuse reports.

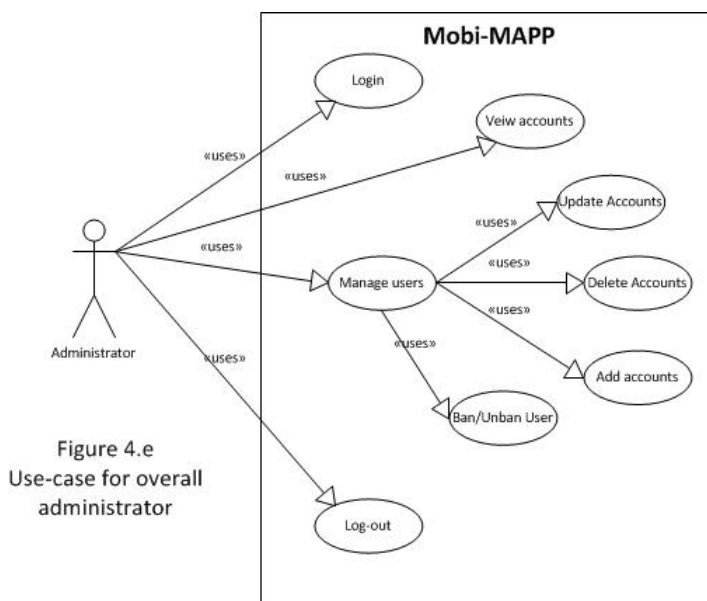
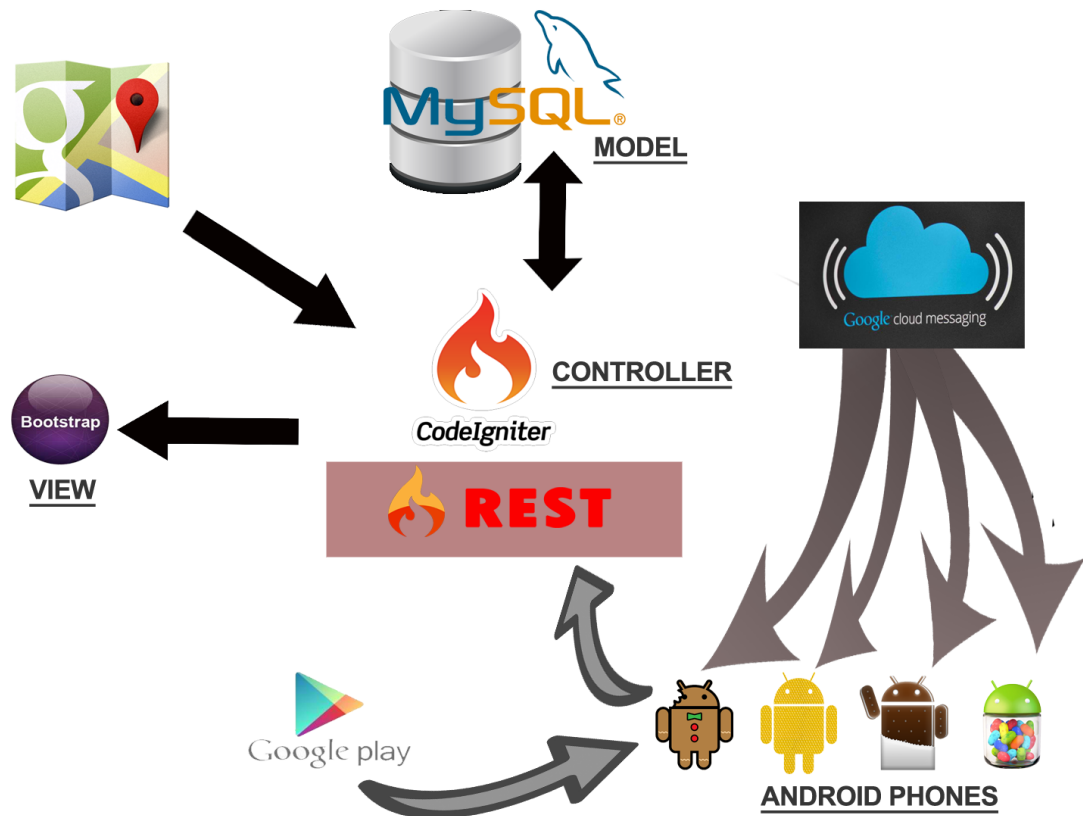


Figure 4.e
Use-case for overall administrator

V. Architecture

A. Technical Architecture

The Mobi-MAPP



System Requirements

The following are required to run the system server:

1. PHP version 5.1.6 or newer.
2. MySQL 5 and above
3. Must have proper API Keys from Google

For the client side:

1. For the web-based system

(a) Javascript-enabled browsers

(b) Has been tested to work on Chrome, Firefox, Internet Explorer 8-11

2. For the mobile client

(a) Android 2.3.3 (GingerBread) and above

(b) Has been tested to work on screens from ldpi-xxhdpi

VI. Results

A. The Mobile Application

Mobi-MAPP is an integrated tool which combines mobile reporting, disaster location, alert system, management and validation provided by the National Disaster Risk Reduction and Management Council (NDRRMC).

The main functionality of the system runs on how it will enable users to view and know basic details of disasters across the Philippines. Figure 1 shows the main activity screen of the Mobi-MAPP android application. Here, disasters are represented by disaster icons (figure 2) attached to their respective geolocation on the map (base map provided by Google Maps).

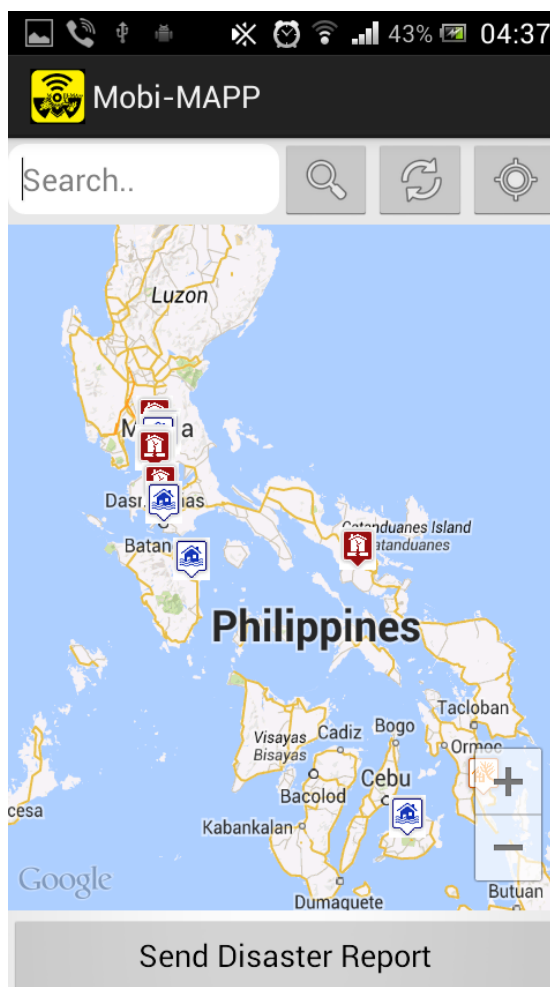


Figure 1: Mobi-Mapp home Activity Screen



Figure 2: Icons used to represent each type of disaster

When these icons are clicked on the map, an info window will pop-up showing the icons respective disaster information. (See figure 3)

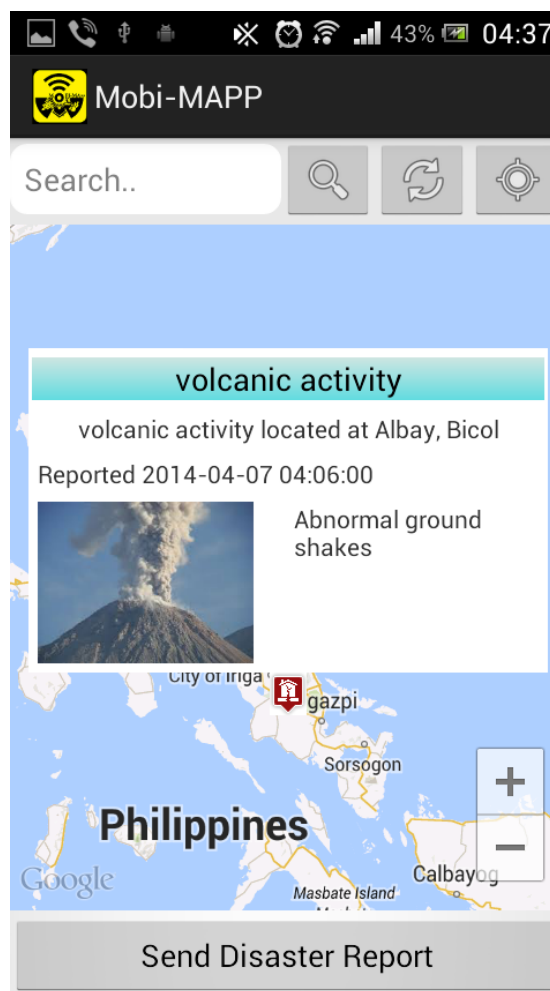


Figure 3: Info Window

In the Main Activity Screen, user can also enter location at the autocomplete search field for his/her desired location (figure 5). With this, the user can check if a disaster is near the location he/she entered. Also, provided is the My Location (figure 4) button, where user can see where he/she is located at the very

moment. Beside the button, is the refresh action button, where the application refreshes the map and gather data again from the database.

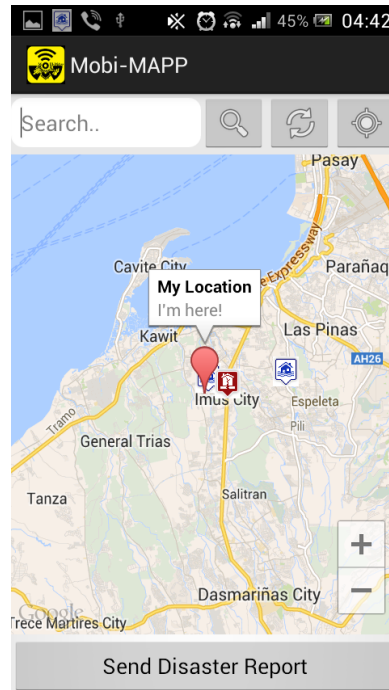


Figure 4: Knowing your current location

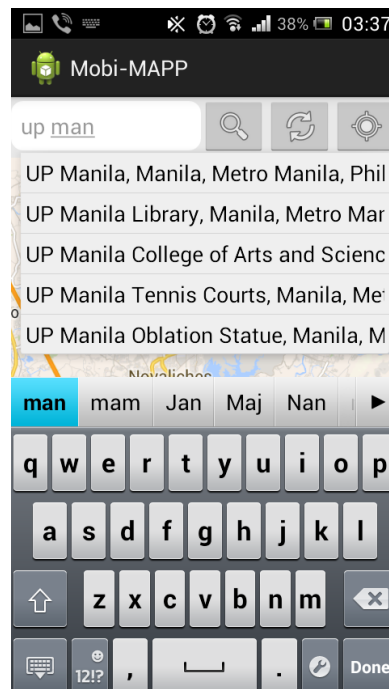


Figure 5: Search Places

Google Maps provides the other functionalities such as zooming, drag to navigate, see routes, etc. The Send Disaster Report button, when clicked, changes the

activity screen from the main home activity to the activity where he/she can report disaster situations. But before the user can send disaster report, he/she must login.first in the system. Figure 6 below shows the sign in activity screen for the user to be able to send report and also to receive incoming alerts (push notifications)

If the user doesnt have account yet, he/she must register. In creating account, the user just have to click register button and he/she well be directed to the register activity screen (Figure 7).

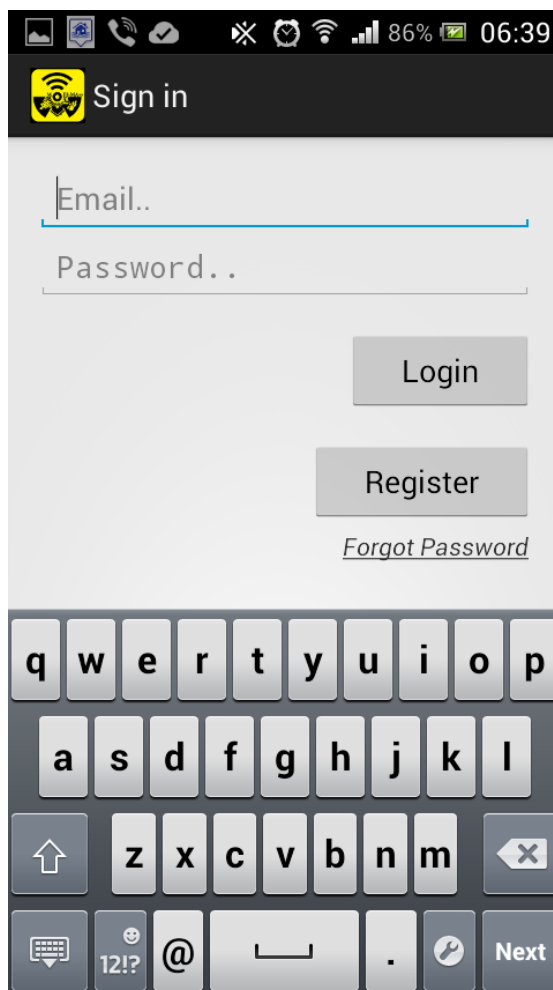


Figure 6: Mobi-Mapp Login Activity Screen

After registration, an email will be sent to the user to verify and activate his account. The email contains the link of the activation of the MobMAPP account. With this, Mobi-MAPP can assure that the user is a real person and really owns

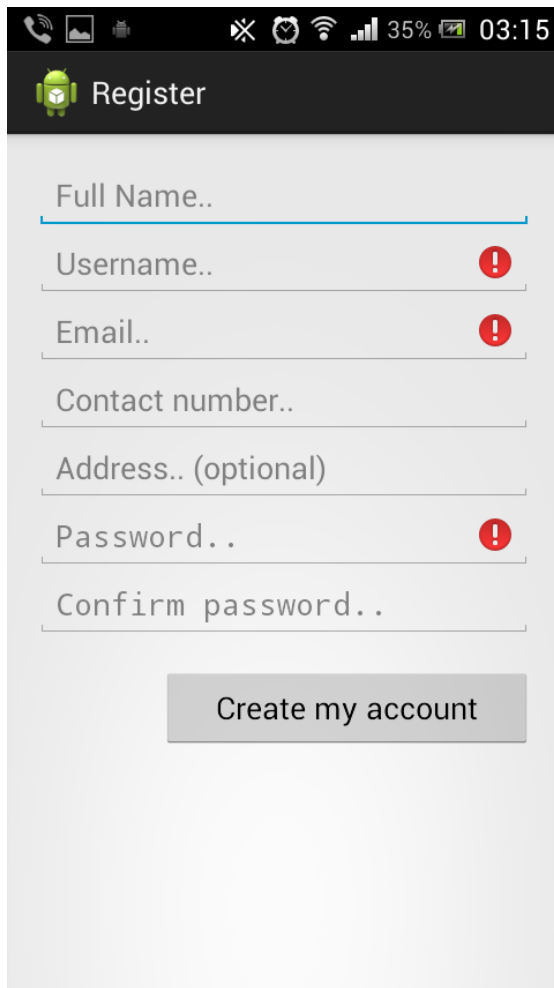


Figure 7: Mobi-Mapp Register Activity Screen

the email account. See figure 8 and figure 9.

After activation of the account, the user can now send mobile reports and also, receive push notifications or alerts for any disaster.

In sending mobile report (figure 10), user can enter report location on a place autocomplete text field provided (figure 11), selects a disaster type on a spinner (figure 12), attach photo and enter some remarks (figure 13)

After sending in the report, it will now be forwarded to the validation section, where NDRRMC personnel will check if the report sent by the user is invalid. The next part shows how NDRRMC will manage these reports and how to handle different disaster data.

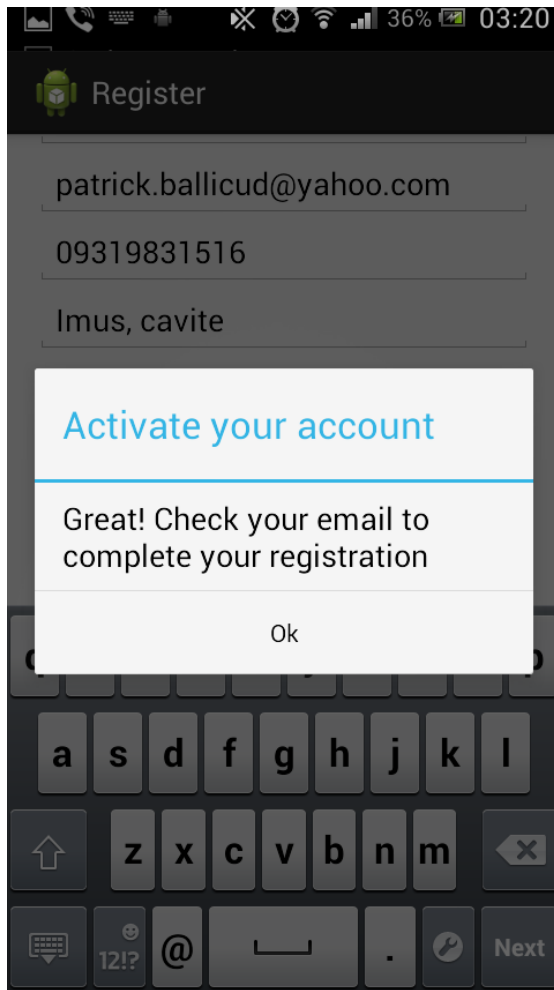


Figure 8: Activate Account

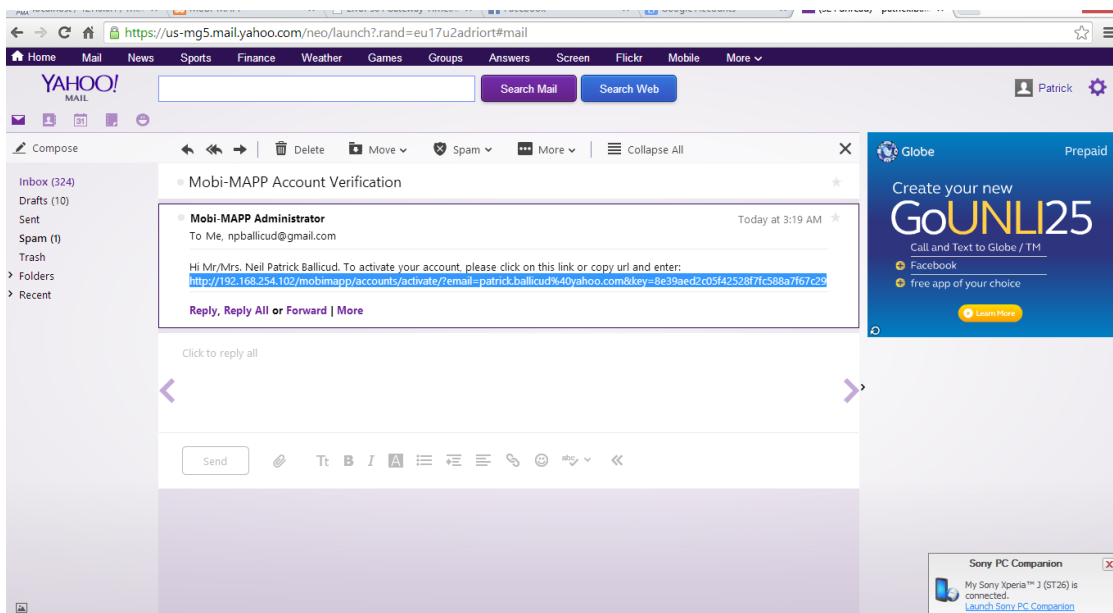


Figure 9: Activation Email

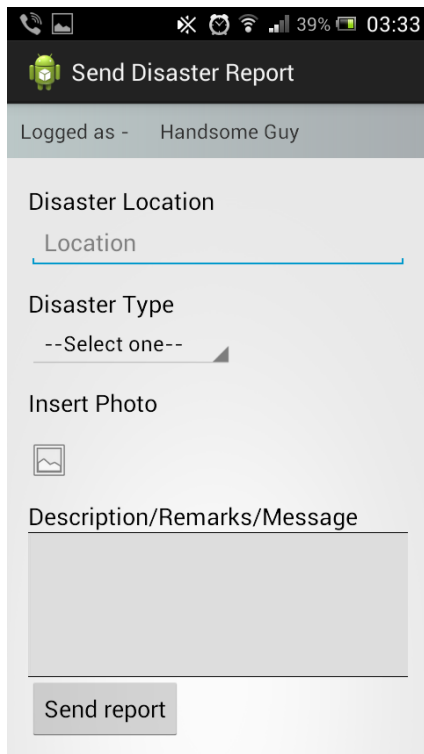


Figure 10: Disaster Report Form

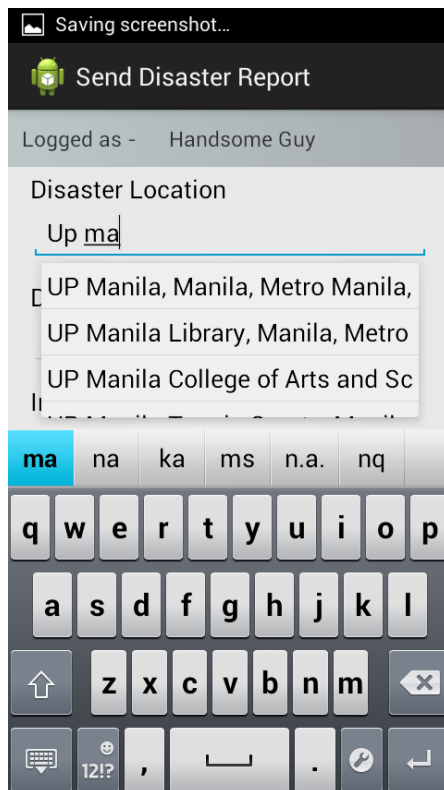


Figure 11: Disaster Autocomplete Location

a

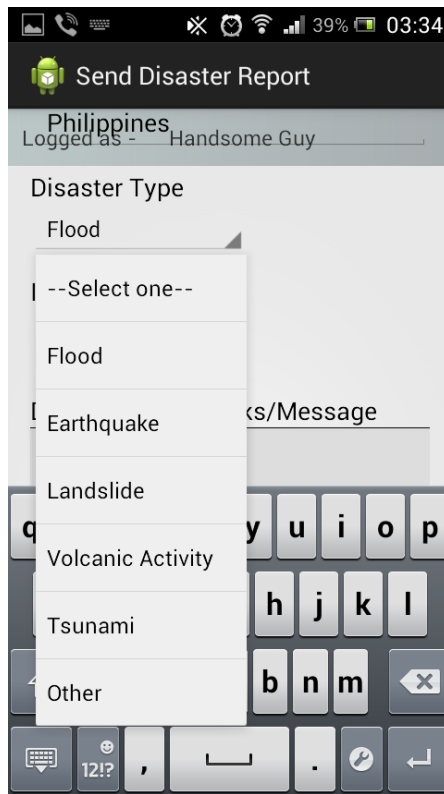


Figure 12: Disaster Spinner for type

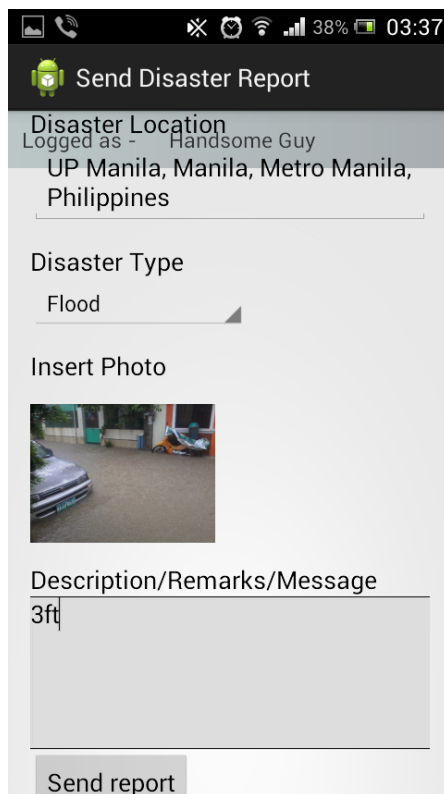


Figure 13: Optional fields

B. The NDRRMC Section

Consistency of data is required before mobile users see the map, check the markers and view the information. The National Disaster Risk Reduction and Management Council (NDRRMC), the governments main agency are the ones in-charge of data input.

There are two ways to insert a disaster record into the main database: 1) Via validation of reports sent in by Mobi-MAPP mobile users and 2) Direct inputting of disaster data. To start with, the NDRRMC officer signs in with his/her account in the Mobi-MAPP. Figure 14 shows the sign in page.

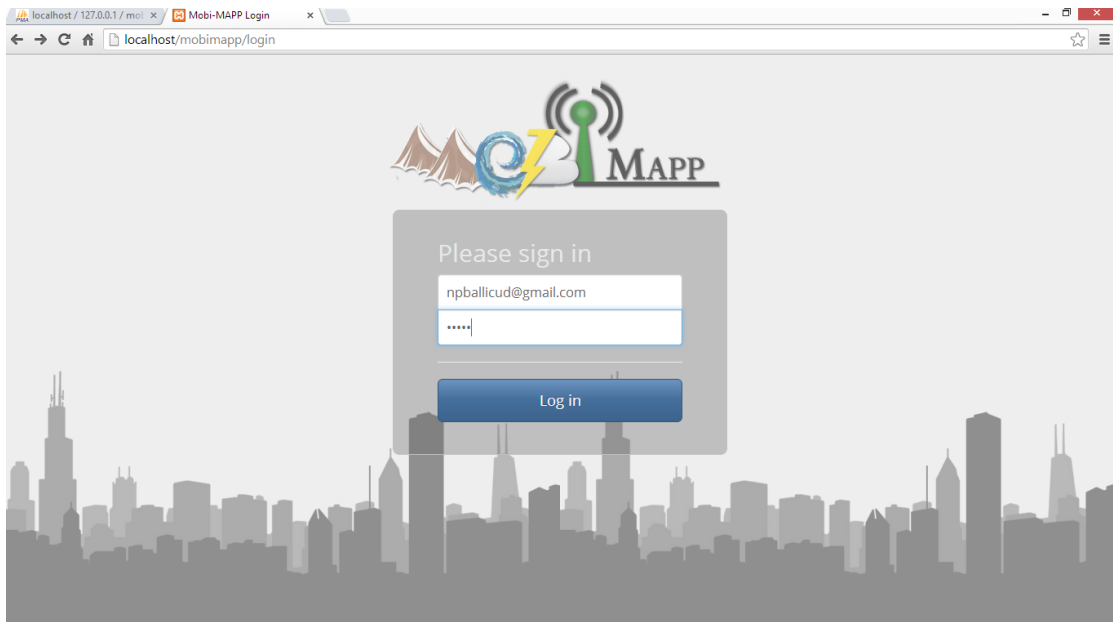


Figure 14: Signin Page

After successful login, the user will be redirected to the reports section where he sees reports from mobile users. Note that a DRRMC unit only sees report with location within its scope. Ex. NDRRMC unit sees all reports across the Philippines (See figure 15) but CALABARZON RDRRMC only sees those with location within Region IV-A (figure 16)

On the list of reports, officer can view full details (e.g. if he wants to see image) by clicking View button (Figure 17). With these, LDRRMC units validate report by clicking the following buttons.

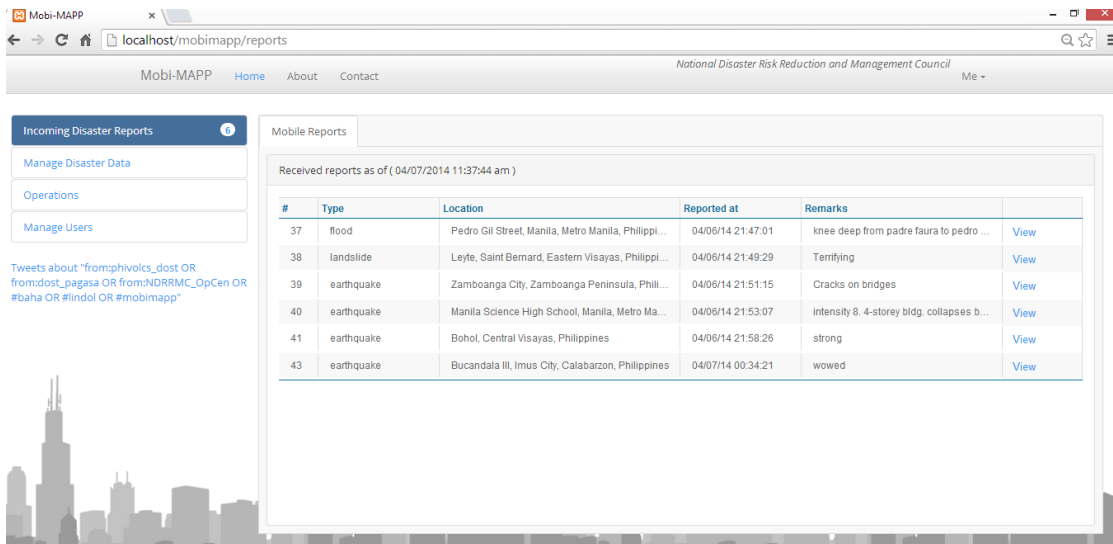


Figure 15: NDRRMC reports

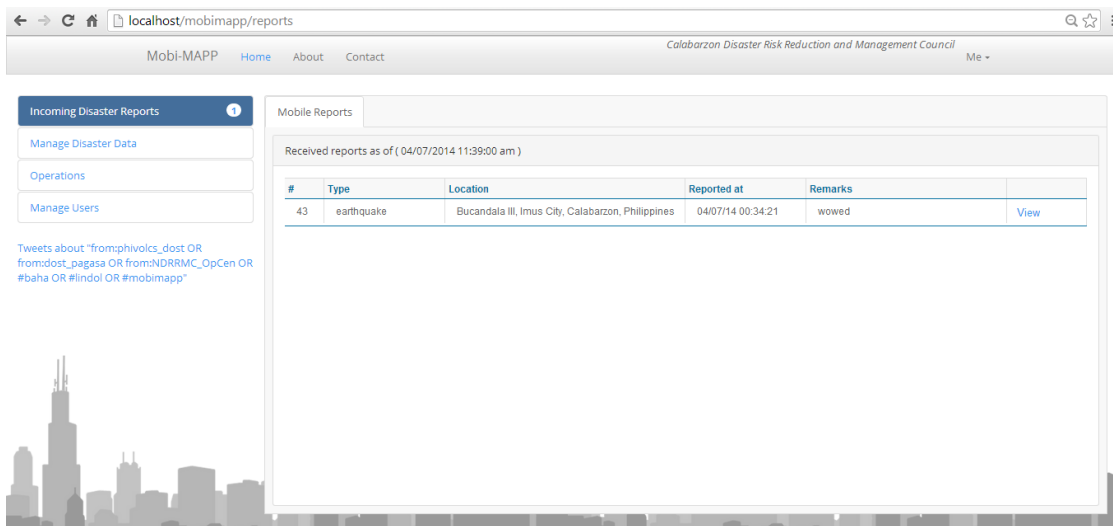


Figure 16: CALABARZON RDRRMC reports

When a report is validated, it will directly be displayed on the current or active disasters tab (shown on Figure 18) and to view one, just click again view link.

As said earlier, disasters can be added to the database not only by validation of reports, but also on direct inputting, this can be done on the Add new tab on the manage disaster data section (Figure 20). When a disaster must be updated, same user interface is also used.

Safe shelters can also be added in adding or updating of a disaster.

Note that only LDRRMC units can add disaster data.

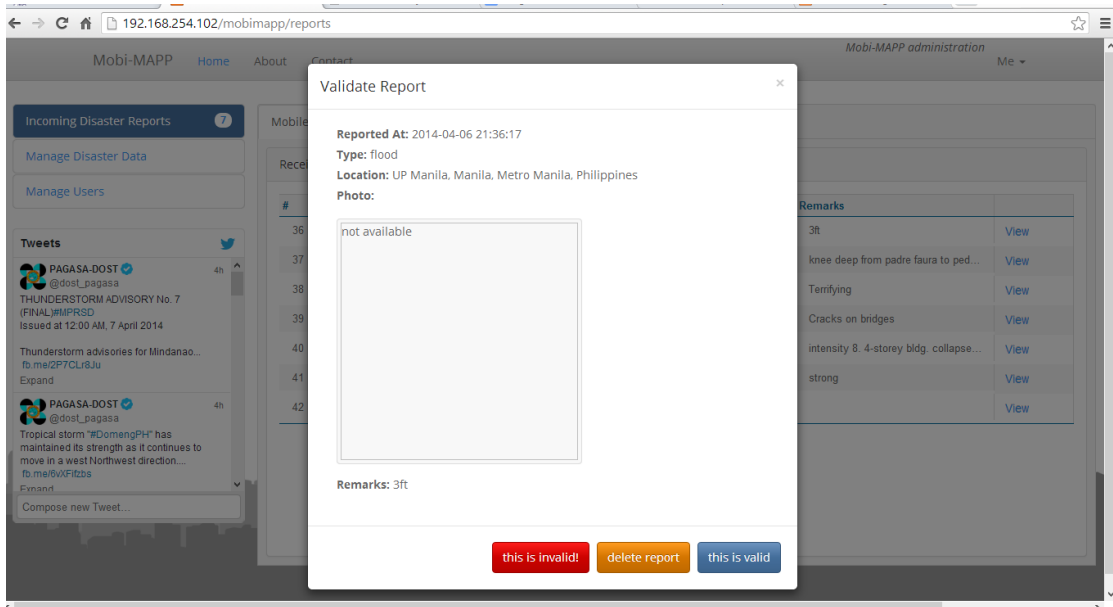


Figure 17: Validation of reports

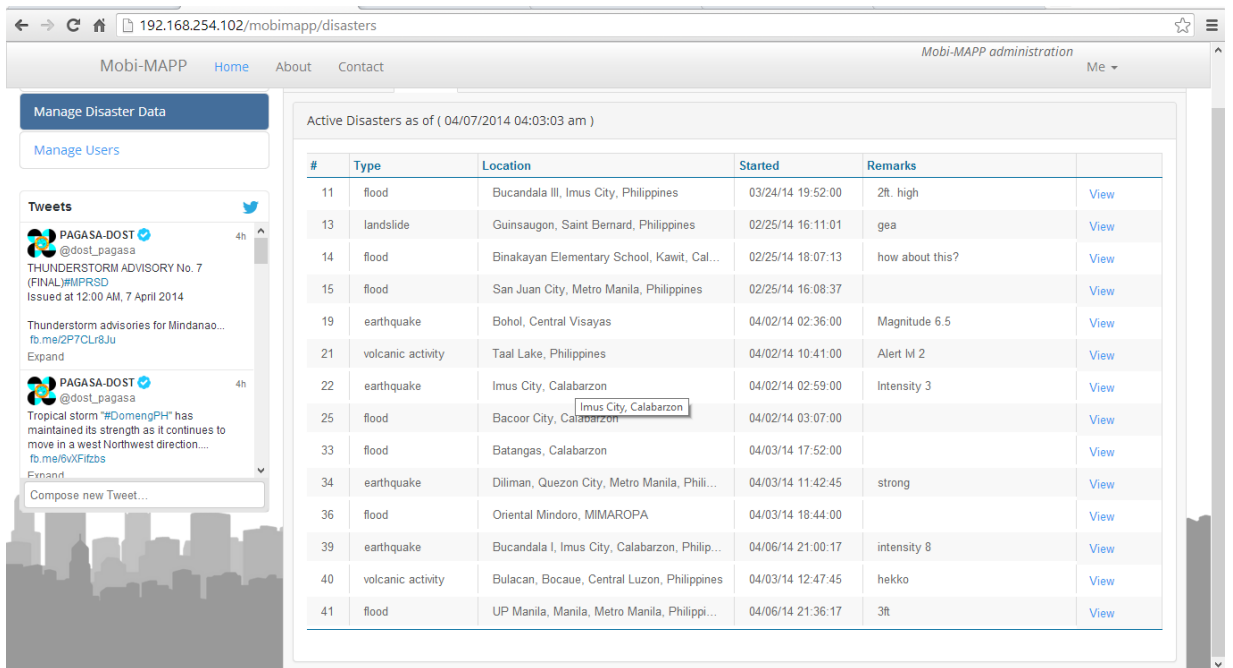


Figure 18: Active disasters

After successful addition of new disaster, officer can check the map by just clicking into the disaster map tab. Just like the mobile application, the map shown in the tab contains markers indicating disasters. Clicking those markers give you information on its respective disaster.

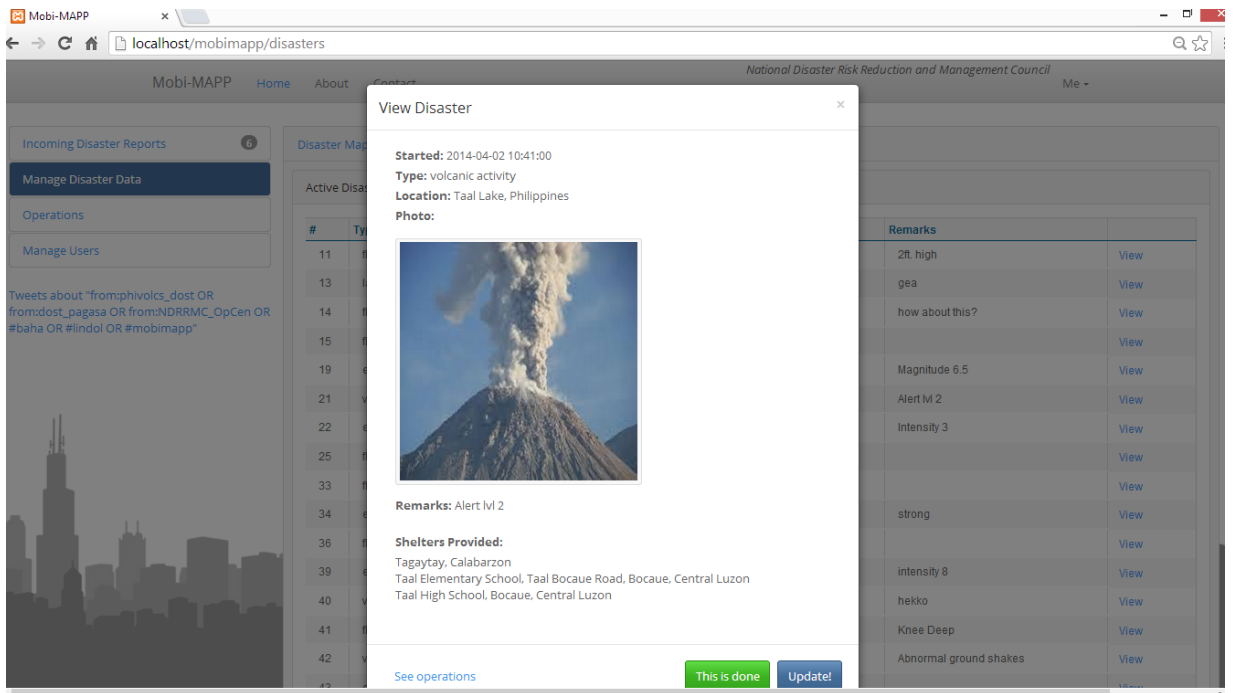


Figure 19: View to Update or End Disaster

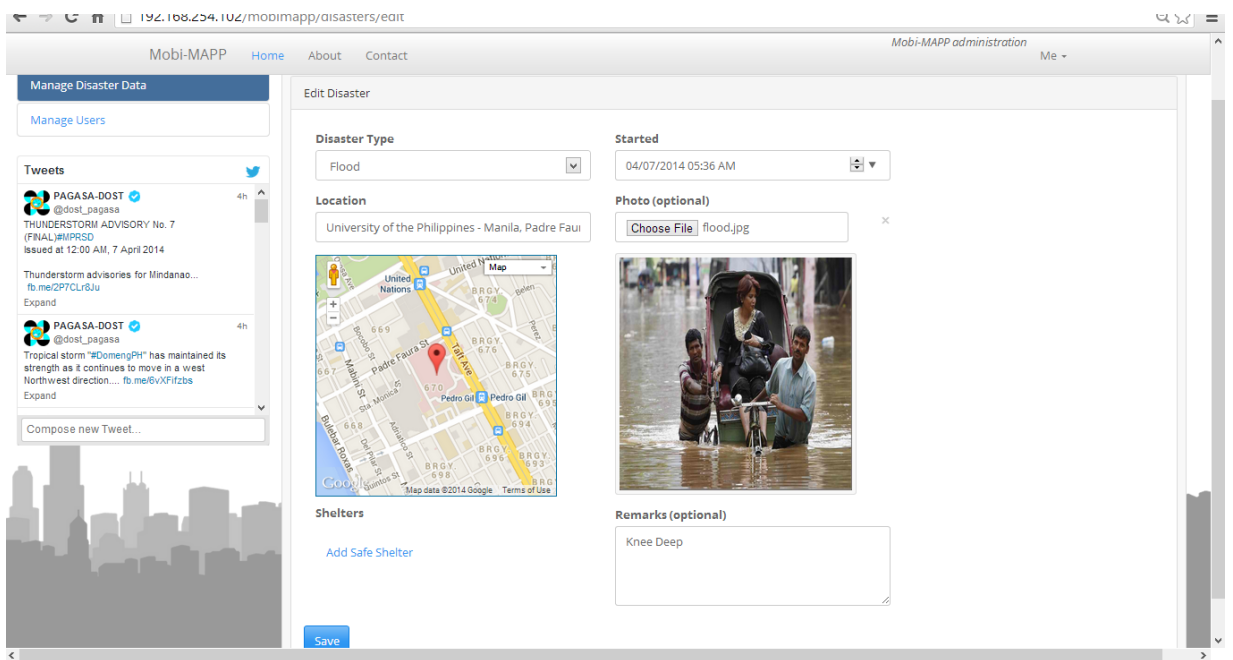


Figure 20: Additon/Update of Disaster

Basic map operations provided by Google Maps like zoom, navigate, satellite and map view, etc. can also be used. (See figure 21)

If the officer want to view past disasters, he/she just clicks the recent tab. These past disasters are sorted according to its recentness. (See figure 22)

Another functionality that Mobi-MAPP offers is that NDRRMC unit person-

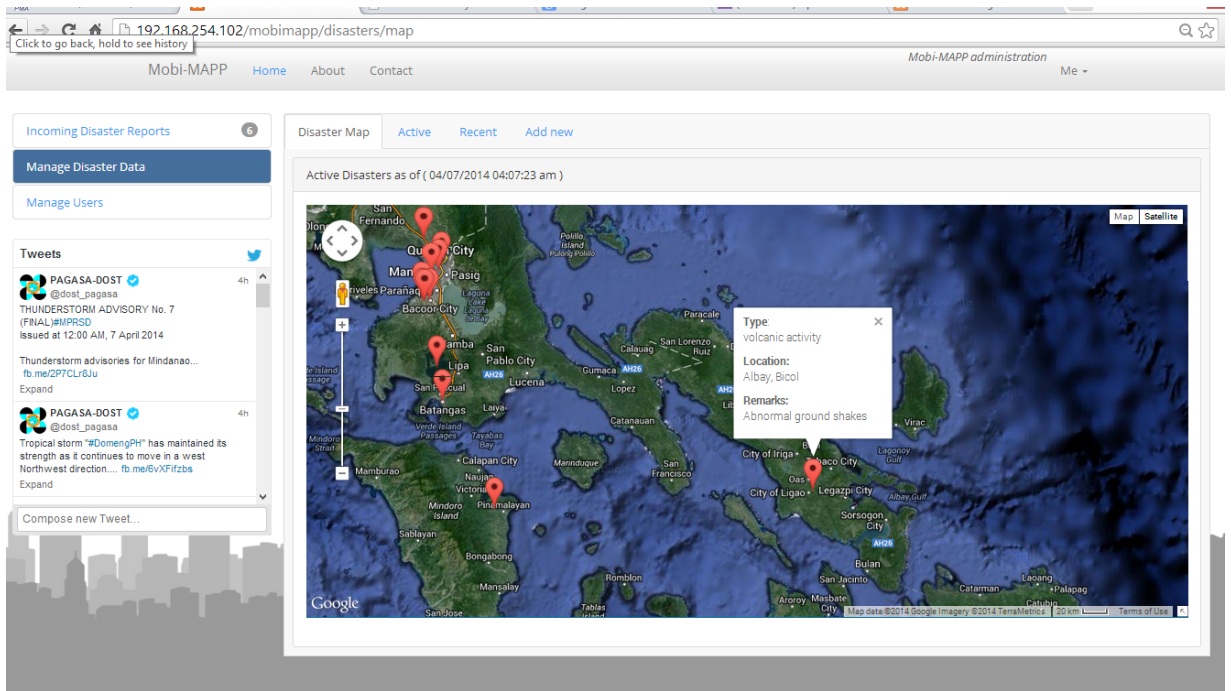


Figure 21: Disaster Map

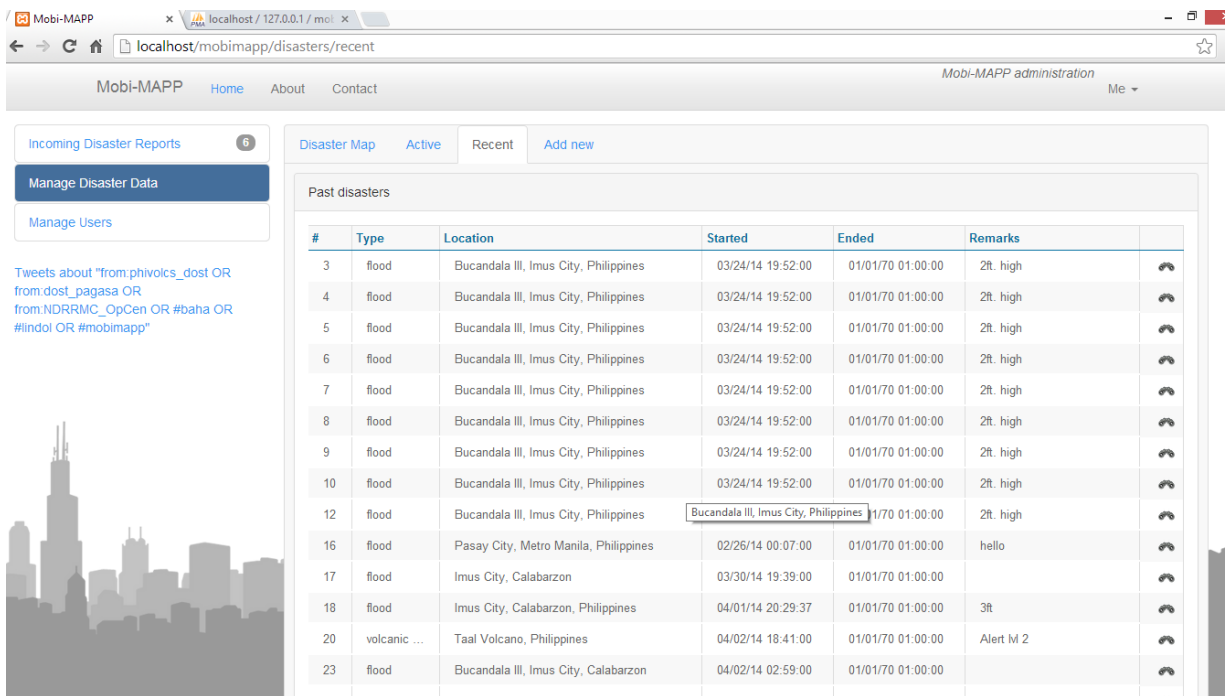


Figure 22: Recent/Past Disasters

nel can communicate with its mother and children units (e.g. RDRRMC can communicate with NDRRMC and its child PDRRMCs). See figure 23 and 24 for sample communication between NDRRMC and CALABARZON RDRRMC

Aside from communication, Mobi-MAPP also offers assignment of officers to

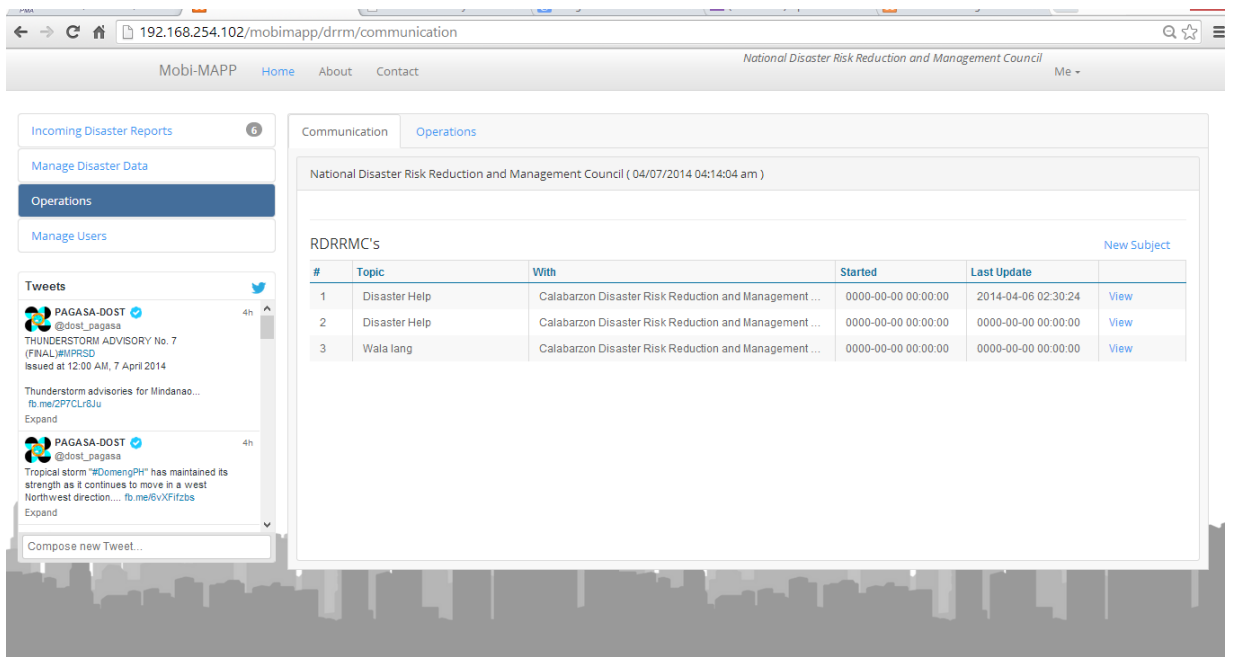


Figure 23: Communication between NDRRMC Units

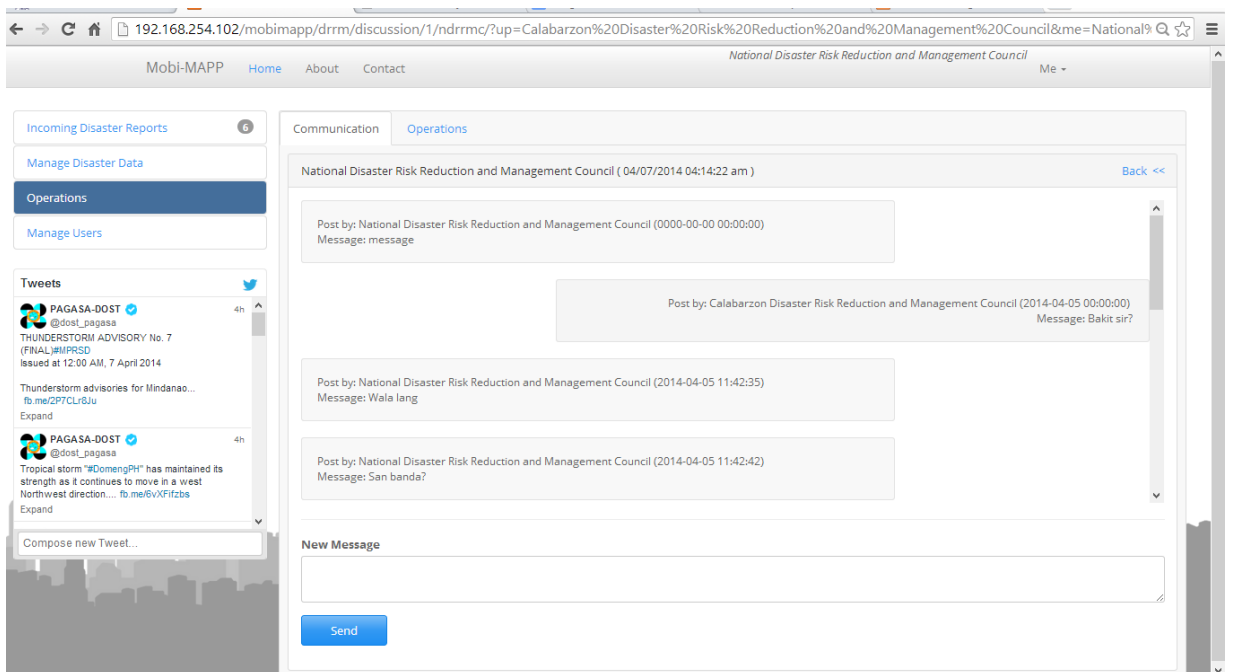


Figure 24: Communication between NDRRMC Units

a specific disaster. Note that only the unit head can assign officers to a specific disaster situation. Also, an officer can only be assigned to one disaster, otherwise he/she will be marked unavailable.

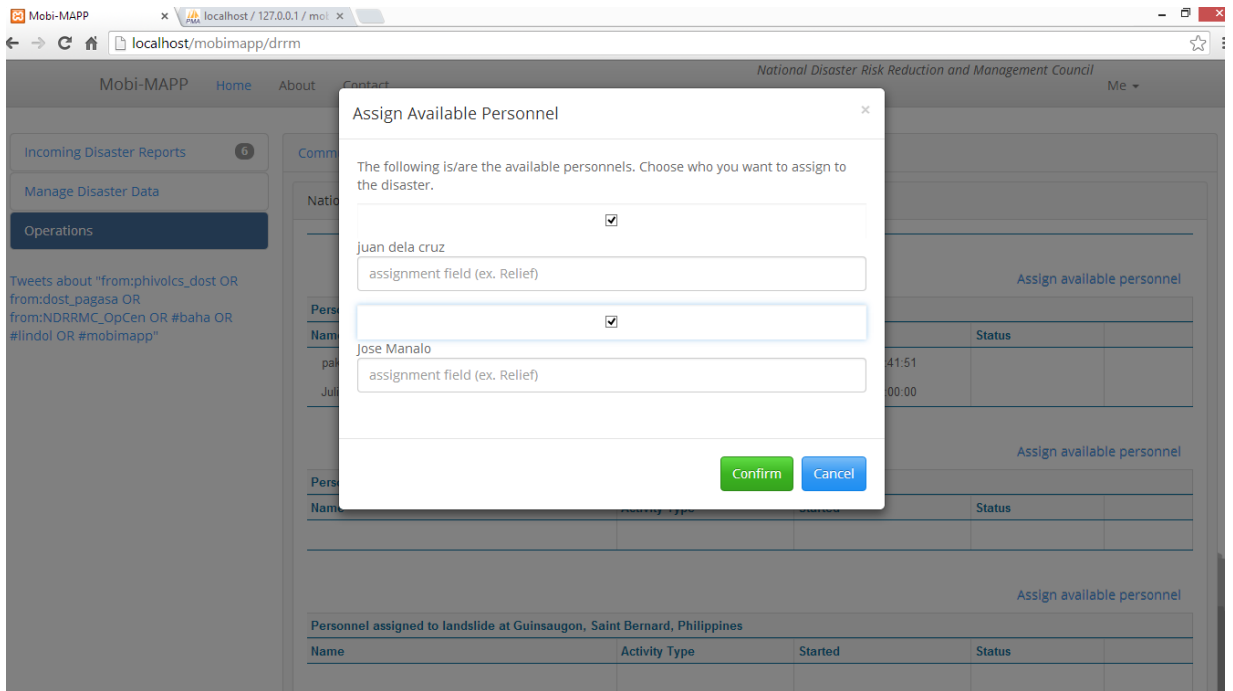


Figure 25: Assign Personnel

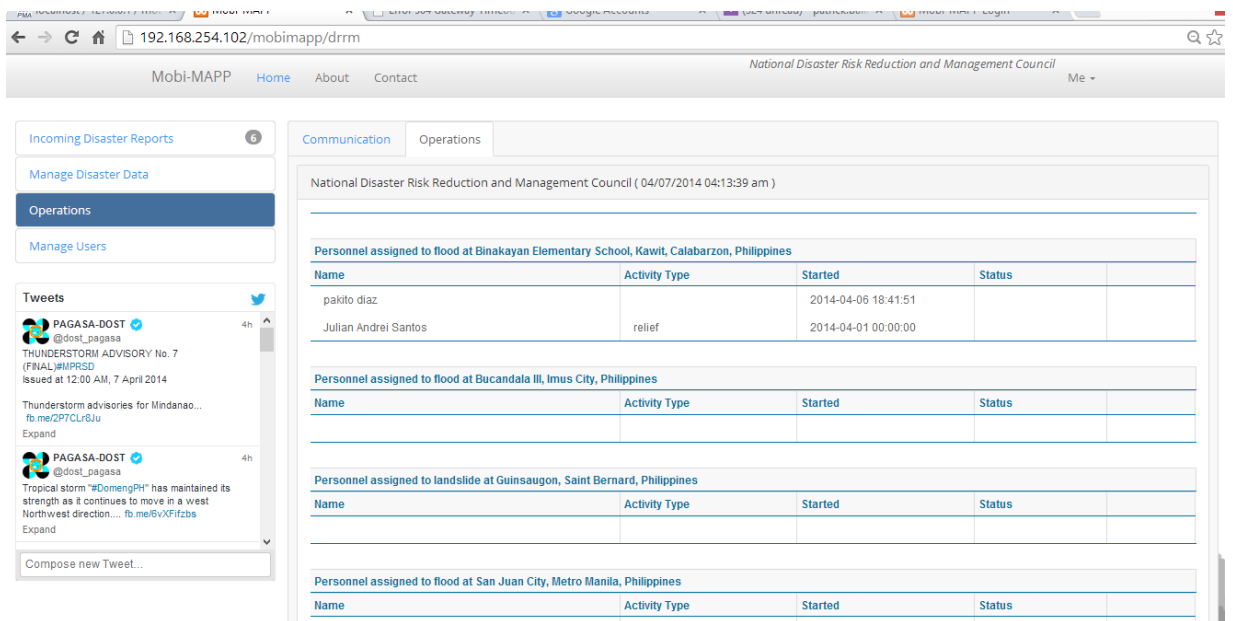


Figure 26: See Operations

C. Mobi-MAPP Administrators

1. NDRRMC unit administrators

NDRRMC unit administrators are the ones responsible for managing users of their respective unit. For example, Imus LDRRMC unit administrator can only manage those users within Imus LDRRMC. See figure 27 and 28

for addition and viewing of accounts.

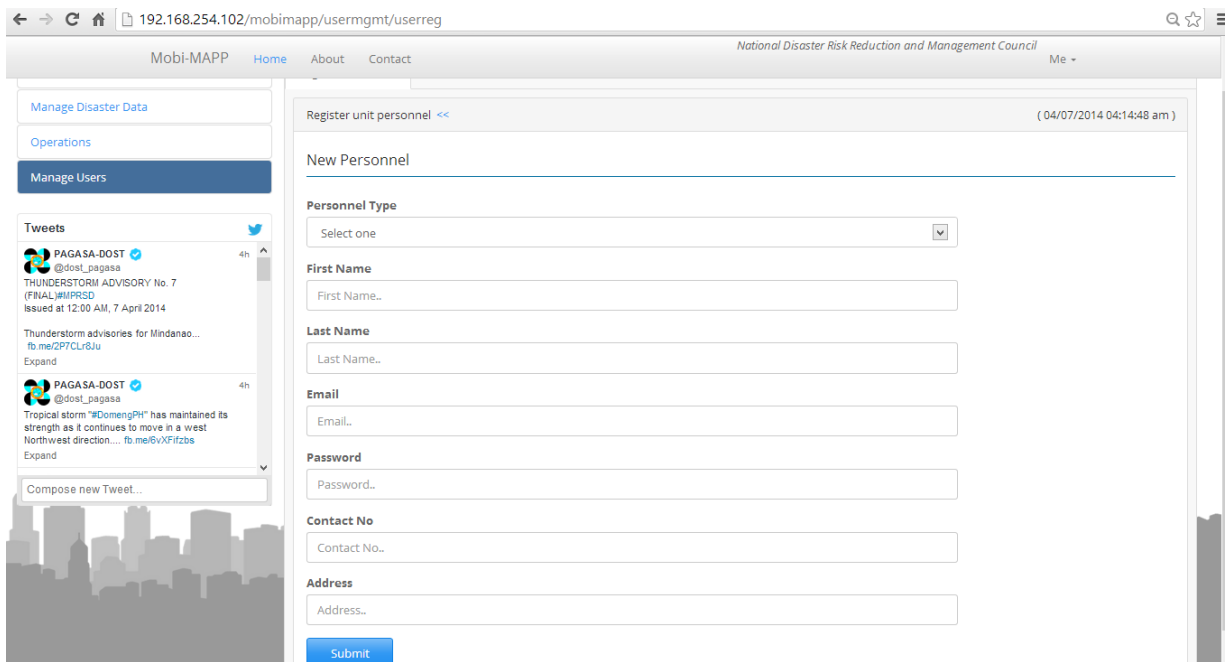


Figure 27: Register NDRRMC unit officer

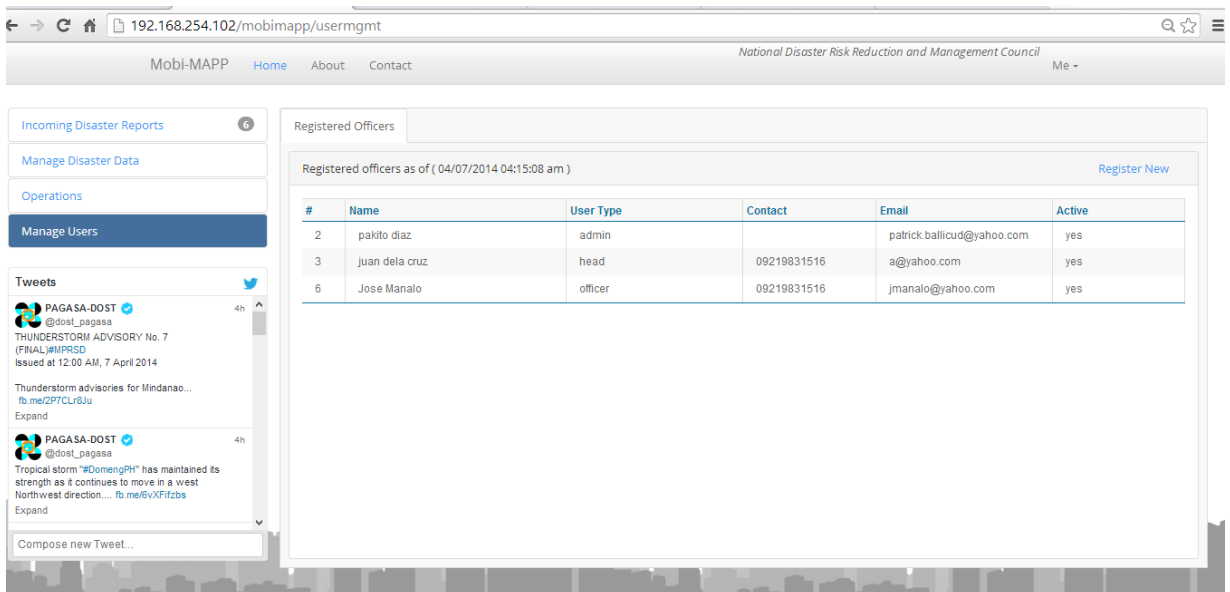


Figure 28: See all officers in the unit

2. Overall administrator

Overall administrators are the one responsible in managing NDRRMC unit and its administrators (Figure 29), registering NDRRMC units (Figure 30 and 31), and see status of mobile users.

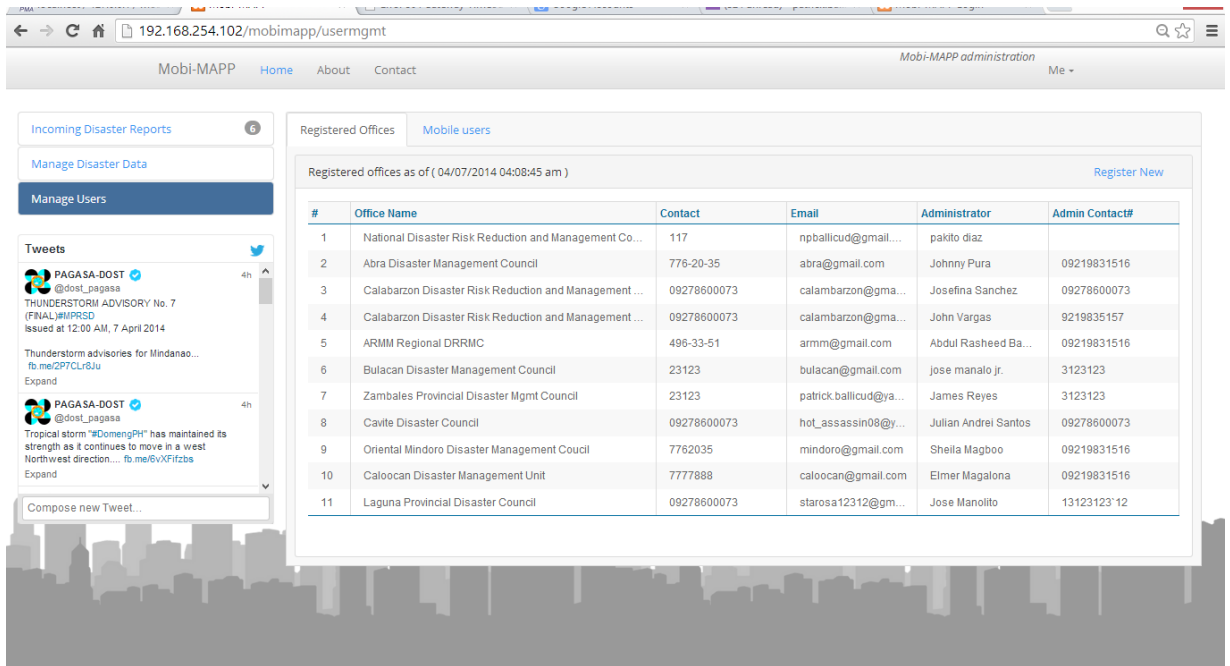


Figure 29: See all registered units

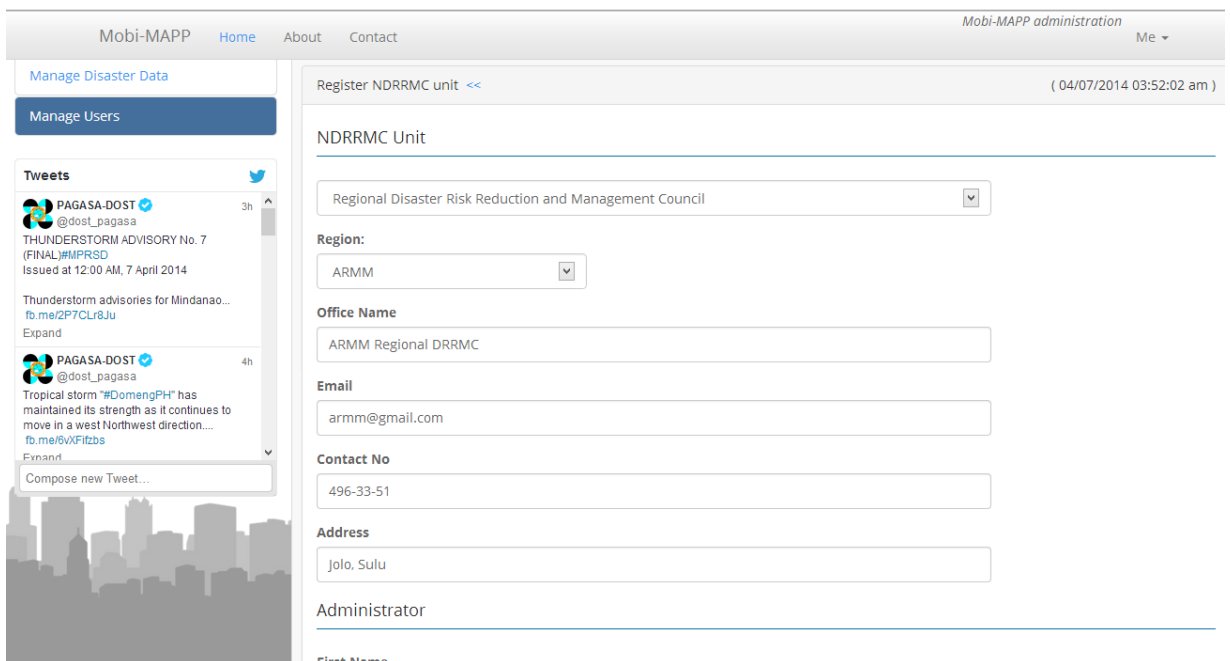


Figure 30: Register Unit

He/She can also see mobile users with bad reputation and ban them from sending reports (Figure 32 and 33)

Mobi-MAPP Home About Contact Mobi-MAPP administration Me ▾

Administrator

First Name

Last Name

Email

Password

Contact No

Address




Figure 31: Register Unit

192.168.254.102/mobimapp/usermgmt/mobile Mobi-MAPP administration Me ▾

Mobi-MAPP Home About Contact

Incoming Disaster Reports 6 | Registered Offices | Mobile users

Manage Disaster Data

Manage Users

Tweets

PAGASA-DOST @dost_pagasa 4h
THUNDERSTORM ADVISORY No. 7 (FINAL)#MFRSD
Issued at 12:00 AM, 7 April 2014
Thunderstorm advisories for Mindanao... fb.me/2P7CLr6Ju
Expand

PAGASA-DOST @dost_pagasa 4h
Tropical storm "DomengPH" has maintained its strength as it continues to move in a west Northwest direction.... fb.me/6vXFfzbs
Expand

Compose new Tweet...

MobiMAPP-mobile users as of (04/07/2014 04:11:01 am)

#	Name	Contact	Email	Reputation	Invalid Reports	Banned	
1	Neil Patrick Ballicud	09278600073	npballicud1@gmail.com	10	-1	no	View
7	asdasdfasdf	123123	patrick.ballicud_1@yahoo...	0	0	no	View
8	asdasdf	123123	patrick.ballicud1@yahoo.com	0	0	no	View
9	Puyo Ballicud	123123	npballicud1@gmail.com	0	0	no	View
10	aaaaa	q123	npballicud2@gmail.com	0	0	no	View
11	patrick ballicud	09278600073	patrick.ballicud1@yahoo.com	0	0	yes	View
12	patrick ballicud jr.	09278600073	patrick.ballicud2@yahoo.com	0	0	no	View
13	puyo jr	09219831516	patrick.ballicud@gmail.com	0	0	no	View
14	puyo gee jr	09219831516	patrick.ballicud3@yahoo.com	0	0	no	View
15	lance ballicud	7762035	patrick.ballicud3@yahoo.com	0	0	no	View
16	qweqweqweqwe	123123	npballicud123123d@gmail.c...	0	0	no	View
17	patrick ballicud jr	123123123	npbnpballicud@gmail.com	0	0	no	View
18	patrick ballicud	09278600073	2312312npballicud@gmail...	0	0	no	View
19	lance pogi ballicud	123123123	npballicud1@gmail.com	3	0	no	View
20	Allan Josue Ballicud	09219831516	hot_assassin13@yahoo.com	0	0	no	View
21	pat ballicuf	1334	hot_assassin@yahoo.com	20	0	no	View
22	Patrick Ballicud	09278600073	npballicud5@gmail.com	23	0	no	View




Figure 32: See all mobile users

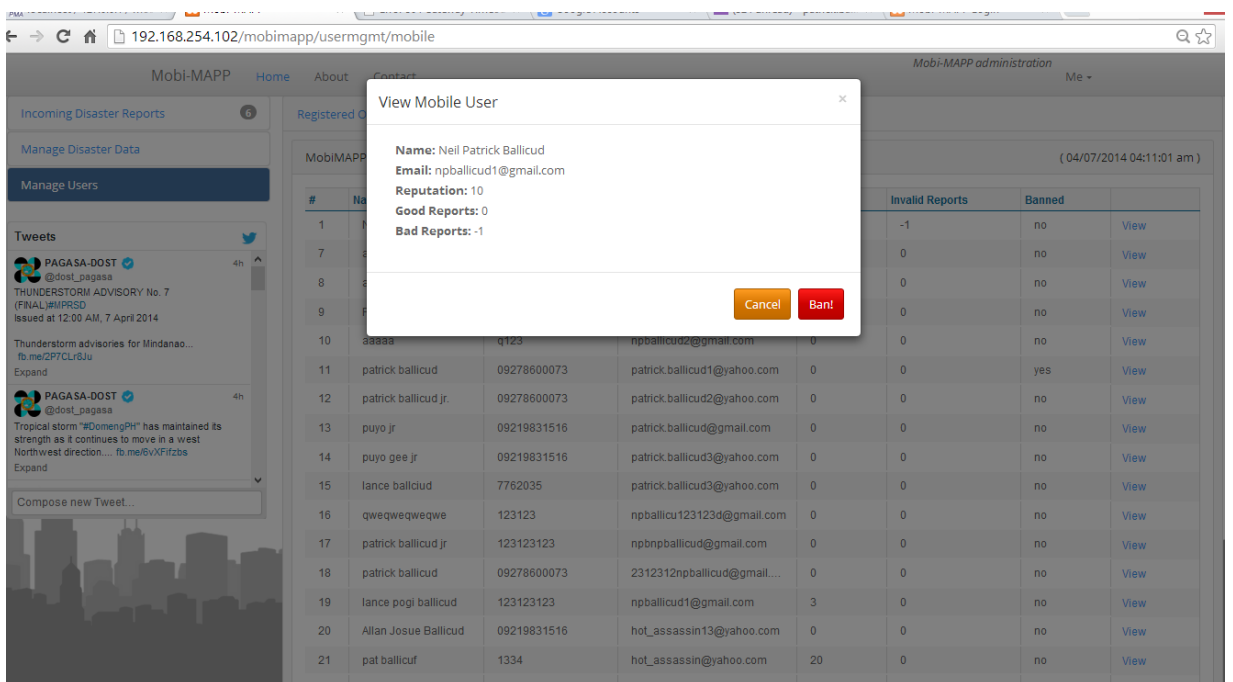


Figure 33: View mobile users and ban/unban

VII. Discussions

This current working version of Mobi-MAPP has functionalities that were able to fulfill the primary objectives of the project. Its primary objectives include, first, to enable mobile users to report disaster situation and in turn, view all disasters currently happening across the country and notify them for a new disaster. Results about show how user can manage to report with the user-friendly interface to guide them and input with the correct format. With this, errors on parsing certain types of reports (like in SMS, MMS report) can be reduced. Note that only registered user, via account registration and email verification, can send disaster report. With this, the system can avoid bots and lessen bogus or invalid reports. Aside from sending of reports, results above show how mobile users can view all active disasters currently happening across the country through a user-friendly map that shows different markers to specific area corresponding to the disaster situation on that particular location. User can search locations and know where he/she is currently located. Another good feature Mobi-MAPP possesses is that it enables user to be notified in case of disaster. Using the Androids Push Notification system, Mobi-MAPP can easily disseminate and alert all registered users of any disasters.

With the following functionalities, we can say that Mobi-MAPP can indeed give mobile user awareness and quick alert in times of disaster. With this, Mobi-MAPP can help in the improvement of community awareness and alert system in the country. Another main objective of the Mobi-MAPP is to give concerned agency, primarily the NDRRMC, an interface to validate disaster reports coming from mobile senders or even in social media feeds. Since NDRRMC is the countrys primary agency in terms of disaster situations, they have the authority to validate reports and to come up with fast disaster data. Also, local NDRRMC units have the authority to add and update disaster data and in turn, mark disaster situations if it already ended. After insertion or modification of disaster data, it will

then be displayed to a map provided which reflects to how mobile users see this disasters on their mobile map.

Additional features include communication among NDRRMC units, dispatching and assignments of officers to certain disaster response operation, and user management among NDRRMC units and mobile users.

With all the functionalities stated above, mobile users can assure consistency of what displays in their screen and what notifies them. Also, with these, we can say that data needed to be checked by certain user in a certain disaster can be collected and consolidated by Mobi-MAPP from validated mobile reports, some social media feeds and from authorized disaster data given by the NDRRMC.

With the development of these functionalities comes with different issues. First was concern with the NDRRMC units where each may have different structure and dynamics in responding to certain types of disasters, including validation of reports and managing disaster data accounted to certain location assign to particular agency unit, as this affects code implementation and database structure of the Mobi-MAPP system.

Another challenge in the development of the system is the integration of different technology used as Mobi-MAPP consists of web-based system for disaster data management done by NDRRMC units and Android-based system for mobile clients in reporting and receiving disaster data. Bridges to these, like Web Services, were critical as Mobi-MAPP Android-based client connects to the main data server and a lot of the functionality of the system depends on it, technologies such as places and geocoding API. Moreover, the system was also limited to certain versions of Android. Since smartphones programming is one of the new and modern trends in software development, new different builds and versions rise and

stability of Mobi-MAPP mobile part is being affected. Mobi-MAPP was tested with minimum version Gingerbread to KitKat.

VIII. Conclusions

The Mobi-MAPP System, (A Mobile Application for Quick Alert and Monitoring of Disasters), caters to the needs of both the DRRMC and the ordinary people for immediate information about disasters

For the DRRMC, it gives them the ability to communicate systemically with each unit through a built in messaging function with upload capability. It allows for an organized action about a disaster situation through assignment of personnel-in-charge. Also, the Mobi-MAPP becomes a venue to consolidate trusted information from PAG-ASA, DOST, PHIVOLCS and other concerned agencies. And for the most part, it becomes an additional and instantaneous source of disaster info because of the reports coming from people who witness the disasters first-hand.

Meanwhile, for the mobile users, it provides them a chance to contribute to the well-being of other people because of the little effort it takes to send a report thru the mobile application. In return, they can also receive early warning through the immediate notification in their mobile devices once a disaster report has been validated by the DRRMC.

Both web and mobile application has a user-friendly interface that can be easily navigated.

IX. Recommendations

The Mobi-MAPP project has adequately addressed the problems it aimed to solve such as quick alert system for disasters and more extensive information dissemination. It can improve the working dynamics of the DRRMC, which is one of the primary users of the system. Also, it draws in the involvement of the ordinary masses through the incorporation of today's trends such as smartphones and social networking sites. There are still, however, some ways to improve the system and make it more useful to its users.

One is the addition of report auto-generation for the use of the Disaster Risk Reduction and Management Council. The application already provides an easily accessible and organized source of disaster data. Adding this extra feature would further ease their job because they would not have to do the reports manually anymore. Also, this would create a standard form of reports for the DRRMC.

Another is the identification of possible shelter houses and the shortest route to them. This will be especially useful during evacuation situations where there is still a high danger for disaster aftermaths during transfer to the evacuation sites. The current Mobi-MAPP system already allows for the assignment of shelter houses by the DRRMC based on proximity so this could already be a basis for the shortest route identification feature.

As mentioned before, modern technologies and trends have been incorporated with the Mobi-MAPP, hence making it more user-friendly to today's generation. That is, relative to the older means of acquiring disaster information wherein we just wait for the TV programs, newspapers, radio and other forms of media to bring it to the people. With the features of the Mobi-MAPP, the people themselves can be a source of information to others. This aspect of the system can still be made more efficient if other forms of social media other than twitter were

used (i.e. Facebook, Google+). As done in the existing Twitter feeds feature, we can take advantage of the hash tag trend in order to accumulate information from ordinary posts from social networking sites. Also, reporting thru SMS will improve the system even further because reports can be sent even without a working internet connection something which will prove to be very useful in times of calamity.

X. Bibliography

- [1] “World disaster 2012: Focus on forced migration and displacement,” October 2012. International Federation of Red Cross and Red Crescent Societies.
- [2] D. West and E. Valentini, “How mobile devices are transforming disaster relief and public safety,” *Center for Technology Innovation at Brookings*, 2013.
- [3] K. M. Rahman, T. Alam, and M. Chowdhury, “Location based early disaster warning and evacuation system on mobile phones using openstreetmap,” *IEEE Conference on Open Systems ICOS*, 2012.
- [4] J. Russel, “Philippines named social networking capital of the world,” May 2011.
- [5] “Project noah.” Available from <http://noah.dost.gov.ph/>.
- [6] A. Carrara and F. Guzzetti, “Geographical information systems in assessing natural hazards,” in *Gis and Natural Hazards: An overview from a Gis Perspective*, pp. 21–34, Kluwer Academic Publishers, 1995.
- [7] J. Ye, G. Lin, M. Zhang, Z. Lin, and J. Chen, “Gis-based study of natural disaster vulnerability for xiamen city,” *18th International Conference on Geoinformatics*, 2010.
- [8] A. Pan, M. Yang, and B. Chen, “Flood hazard evaluation and gis in guangzhou,” *International Conference on Multimedia Technology (ICMT)*, 2010.
- [9] J. Santillan, E. Paringit, R. Ramos, R. Mendoza, N. Espanola, and J. Alconis, “Near real-time flood extent monitoring in marikina river, philippines: Model parameterization using remotely-sensed data and field measurements,” in *Proceedings of the 33rd Asian Conference on Remote Sensing (ACRS)*, 2012.

- [10] K. K. Lwin and Y. Muruyama, "Web-based gis system for real-time field data collection using a personal mobile phone," *Journal of Geographic Information System*, 2011.
- [11] J. Ortmann, M. Limbu, D. Wang, and T. Kauppinen, "Crowdsourcing linked open data for disaster management," in *Proceedings of Terra Cognita 2011, The 10th International Semantic Web Conference*, 2011.
- [12] H. Gao, "Harnessing the crowdsourcing power of social media for disaster relief," *IEEE Intelligent Systems*, 2011.
- [13] R. Fontugne, K. Cho, Y. Won, and K. Fukuda, "Disasters seen through flickr cameras,," in *Proceedings of Special Workshop on Internet and Disasters (SWID)*, ACM, 2011.
- [14] C. Hodapp, M. Robbins, J. Gray, and A. Graettinger, "Damage tracker - a cloud and mobile system for collecting damage information after natural disasters," in *Proceedings of the 51st ACM Southeast Conference*, 2013.
- [15] "Openstreetmap." Available from <http://wiki.openstreetmap.org/>.
- [16] "Cloudmade." Available from <http://cloudmade.com/technologies/hybrid>.
- [17] B. Gamboa, "Mobile disaster communication system." Undergraduate Special Problem, Dept. Phys. and Math., UP Manila, 2013.
- [18] J. Alimanza, "Mobile disaster reporting system." Undergraduate Special Problem, Dept. Phys. and Math., UP Manila, 2011.
- [19] "Definition of disaster." Retrieved from Merriam-Webster (<http://www.merriamwebster.com/dictionary/disaster>).
- [20] "Definition of disaster." Retrieved from International Federation of Red Cross (<http://ifrc.org>).

- [21] “Definition of disaster risk reduction.” Retrieved from The United Nations Office of Disaster Risk Reduction (UNISDR).
- [22] “National disaster risk reduction and management framework,” 2012. Framework from NDRRMC.
- [23] “Republic act 10121 an act strengthening the ndrrmf and institutionalizing the national disaster risk reduction and management plan, appropriating funds therefor and for other purposes.” from Philippine Consitution passed 2009.
- [24] “Definition of smartphone.” Available from PhoneScoop(<http://www.phonescoop.com/glossary/term.php?gid=131>).
- [25] “Geographic information systems as an integrating technology: Context, concepts, and definitions.” Retrieved from <http://www.colorado.edu/geography/gcraft/notes/intro/intro.html>.
- [26] “Definition of web services.” Available from <http://searchsoa.techtarget.com/definition/Web-services>.
- [27] “Google play services.” Available from <http://developer.android.com/google/play-services/index.html>.
- [28] “Google maps.” Available from <https://developers.google.com/maps/>].
- [29] “Google maps api.” Available from <https://developers.google.com/maps/>].
- [30] “Google locations api.” Available from <https://developer.android.com/google/play-services/location.html>.
- [31] “Google cloud messaging.” Available from <http://developer.android.com/google/gcm/index.html>.

XI. Appendix

A. Source Code

The following codes refer to the Strings and Styles used for the UI of the Android Application

strings_activity_disaster_report.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
  <string name="title_active_disaster_report">Report Disaster </string>
  <string name="prompt_disaster_location">Location </string>
  <string name="text_disaster_location">Disaster Location </string>
  <string name="text_disaster_type">Disaster Type </string>
  <string name="text_remarks">Description/Remarks/Message </string>
  <string name="text_insert_photo">Insert Photo </string>

  <string-array name="array_disaster_type">
    <item>--Select one--</item>
    <item>Flood</item>
    <item>Earthquake</item>
    <item>Landslide</item>
    <item>Volcanic Activity </item>
    <item>Tsunami</item>
    <item>Fire</item>
  </string-array>
  <string name="title_activity_disaster_report">Send Disaster Report </string>
  <string name="action_send_report">Send report </string>
  <string name="progress_sending_report">Sending your disaster report... </string>
  <string name="error_enter_location">Please enter location </string>
</resources>
```

strings_activity_login.xml

```
<resources>
  <string name="title_activity_login">Sign in </string>

  <!-- Strings related to login -->
  <string name="prompt_email">Email </string>
  <string name="prompt_password">Password </string>
  <string name="action_sign_in_register"><b>Sign in</b> or register </string>
  <string name="action_sign_in_short">Sign in </string>
  <string name="action_forgot_password">Recover lost password </string>
  <string name="login_progress_signing_in">Signing in&#8230;</string>
  <string name="error_invalid_email">This email address is invalid </string>
  <string name="error_invalid_password">This password is too short </string>
  <string name="error_incorrect_password">This password is incorrect </string>
  <string name="error_field_required">This field is required </string>
  <string name="text_forgot_password"><u><i>Forgot Password</i></u></string>
  <string name="process_forgot_password">Sending request.. </string>
</resources>
```

strings_activity_register.xml

```
<resources>
  <string name="title_activity_register">Register </string>

  <!-- Strings related to login -->
  <string name="prompt_email">Email.. </string>
  <string name="prompt_password">Password.. </string>
  <string name="action_forgot_password">Recover lost password </string>
  <string name="register_progress_creating_account">Creating your account.. </string>
  <string name="error_invalid_email">This email address is invalid </string>
  <string name="error_invalid_password">This password is too short </string>
  <string name="error_incorrect_password">This password is incorrect </string>
  <string name="error_field_required">This field is required </string>
  <string name="prompt_fullname">Full Name.. </string>
  <string name="prompt_confirm_password">Confirm password.. </string>
  <string name="error_password_do_not_match">password does not match </string>
  <string name="action_create_account">Create my account </string>
  <string name="error_email_already_exists">Email already exists </string>
  <string name="prompt_user_address">Address.. (optional) </string>
  <string name="prompt_contact_no">Contact number.. </string>
  <string name="prompt_username">Username.. </string>
</resources>
```

strings.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
  <string name="app_name">Mobi-MAPP </string>
  <string name="action_settings">Settings </string>
  <string name="hello_world">Hello world! </string>
  <string name="send_report">Send Disaster Report </string>
  <string name="logged_as">Logged as - </string>
  <string name="log_name">Anonymous </string>
  <string name="refresh">Refresh.. </string>
</resources>
```

```

<string name="my_location">My location</string>
<string name="title_activity_disaster_report">Send Disaster Report</string>
<string name="title_activity_mobi_login">Mobi.Login</string>
<string name="login_username">Username: </string>
<string name="login_password">Password: </string>
<string name="login_register">Register</string>
<string name="login_go">Login</string>
<string name="login_forgot"><u>forgot acct?</u></string>
<string name="search">Search..</string>
<string name="logout">Logout</string>
<string name="about">About</string>
<string name="deactivate">Deactivate</string>
<string name="server_registering">Trying (attempt %1$d/%2$d) to register device on Demo Server.</string>
<string name="server_registered">From Demo Server: successfully added device!</string>
<string name="server_unregister">From Demo Server: successfully removed device!</string>
<string name="server_register_error">Could not register device on Demo Server after %1$d attempts.</string>
<string name="server_unregister_error">Could not unregister device on Demo Server (%1$s).</string>
<string name="gcm_registered">From GCM: device successfully registered!</string>
<string name="gcm_unregister">From GCM: device successfully unregistered!</string>
<string name="gcm_message">From GCM: you got message!</string>
<string name="gcm_error">From GCM: error (%1$s).</string>
<string name="gcm_recoverable_error">From GCM: recoverable error (%1$s).</string>
<string name="gcm_deleted">From GCM: server deleted %1$d pending messages!</string>

</resources>

styles.xml
-----
<?xml version="1.0" encoding="utf-8"?>
<resources>

    <!--
        Base application theme, dependent on API level. This theme is replaced
        by AppBaseTheme from res/values-vXX/styles.xml on newer devices.
    -->
    <style name="AppBaseTheme" parent="android:Theme.Light">
        <!--
            Theme customizations available in newer API levels can go in
            res/values-vXX/styles.xml, while customizations related to
            backward-compatibility can go here.
        -->

        </style>

    <!-- Application theme. -->
    <style name="AppTheme" parent="AppBaseTheme">
        <!-- All customizations that are NOT specific to a particular API-level can go here. -->
    </style>

    <style name="LoginFormContainer">
        <item name="android:layout_width">match_parent</item>
        <item name="android:layout_height">wrap_content</item>
        <item name="android:padding">16dp</item>
    </style>
</resources>

```

The following codes refer to the Layouts for the GUI of the Android Application

```

activity_main.xml
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:gravity="center">

    <fragment
        android:id="@+id/map"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_marginTop="50dp"
        android:layout_marginBottom="50dp"
        android:name="com.google.android.gms.maps.SupportMapFragment"/>

    <Button
        android:id="@+id/send_report_but"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:layout_alignParentLeft="true"
        android:layout_alignParentRight="true"
        android:text="@string/send_report"
        android:onClick="loginUser" />

    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_alignParentRight="true"
        android:layout_alignParentTop="true" >

        <AutoCompleteTextView
            android:id="@+id/autocomp_search_ET"
            android:layout_width="0dp"

```



```

        android:layout_weight="0.49"
        android:layout_height="wrap_content"
        android:padding="8dip"
        android:singleLine="true"
        android:textColor="#999999"
        android:layout_gravity="center"
        android:background="@drawable/rounded_edittext"
        android:dropDownWidth="fill_parent"
        android:hint="@string/search" />

<ProgressBar
    android:id="@+id/address_progress"
    android:layout_width="0dp"
    android:layout_height="0.17"
    android:layout_weight="wrap_content"
    android:gravity="center"
    android:indeterminate="true"
    android:visibility="gone" />

<ImageButton
    android:id="@+id/search_but"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_weight="0.17"
    android:src="@drawable/ic_menu_search" />

<ImageButton
    android:id="@+id/refresh_but"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_weight="0.17"
    android:src="@drawable/ic_menu_refresh"
    android:contentDescription="@string/refresh" />

<ImageButton
    android:id="@+id/my_loc_but"
    android:layout_width="0dp"
    android:layout_weight="0.17"
    android:layout_height="wrap_content"
    android:src="@drawable/ic_menu_mylocation"
    android:contentDescription="@string/my_location" />

</LinearLayout>

</RelativeLayout>
activity_disaster_report.xml
<merge xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    tools:context=".DisasterReport" >

<LinearLayout
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:background="@drawable/grayish_bg">

<TextView
    android:layout_width="0dp"
    android:layout_weight="0.35"
    android:layout_height="wrap_content"
    android:padding="10dp"
    android:textSize="15sp"
    android:text="@string/logged_as"
    android:gravity="left"
/>

<TextView
    android:id="@+id/usernameView"
    android:layout_width="0dp"
    android:layout_weight="0.68"
    android:layout_height="wrap_content"
    android:padding="7dp"
    android:textSize="15sp"
/>

</LinearLayout>

<!-- Login progress -->

<LinearLayout
    android:id="@+id/send_report_status"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:gravity="center_horizontal"
    android:orientation="vertical"
    android:visibility="gone" >

<ProgressBar
    style="?android:attr/progressBarStyleLarge"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="8dp" />

<TextView
    android:id="@+id/send_report_status_message"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="16dp"

```

```

        android:fontFamily="sans-serif-light"
        android:text="@string/progress_sending_report"
        android:textAppearance="?android:attr/textAppearanceMedium" />
</LinearLayout>

<!-- Login form -->

<ScrollView
    android:id="@+id/send_report_form"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_gravity="left" >

    <LinearLayout
        style="@style/LoginFormContainer"
        android:orientation="vertical" >

        <TextView
            android:id="@+id/text_disaster_location"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginTop="45dp"
            android:text="@string/text_disaster_location"
            android:textAppearance="?android:attr/textAppearanceMedium" />

        <AutoCompleteTextView
            android:id="@+id/disaster_location"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:ems="10"
            android:hint="@string/prompt_disaster_location" >

        </AutoCompleteTextView>

        <TextView
            android:id="@+id/text_disaster_type"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginTop="14dp"
            android:text="@string/text_disaster_type"
            android:textAppearance="?android:attr/textAppearanceMedium" />

        <Spinner
            android:id="@+id/disaster_type_spinner"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"/>

        <TextView
            android:id="@+id/text_add_photo"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginTop="14dp"
            android:text="@string/text_insert_photo"
            android:textAppearance="?android:attr/textAppearanceMedium" />

        <ImageView
            android:id="@+id/disaster_photo_imageview"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginTop="15dp"
            android:scaleType="fitStart"
            android:adjustViewBounds="true"
            android:src="@android:drawable/ic_menu_gallery"
            android:onClick="onPhotoClick" />

        <TextView
            android:id="@+id/text_disaster_remarks"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginTop="14dp"
            android:text="@string/text_remarks"
            android:textAppearance="?android:attr/textAppearanceMedium" />

        <EditText
            android:id="@+id/disaster_remarks"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:background="@drawable/textlines"
            android:gravity="top"
            android:singleLine="false"
            android:lines="5"
            />

        <Button
            android:id="@+id/button_send_report"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/action_send_report"
            />

    </LinearLayout>
</ScrollView>

</merge>

activity_login.xml
-----
<merge xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    tools:context=".LoginActivity" >

```

```

<!-- Login progress -->
<LinearLayout
    android:id="@+id/login_status"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:gravity="center_horizontal"
    android:orientation="vertical"
    android:visibility="gone" >

    <ProgressBar
        style="?android:attr/progressBarStyleLarge"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginBottom="8dp" />

    <TextView
        android:id="@+id/login_status_message"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginBottom="16dp"
        android:fontFamily="sans-serif-light"
        android:text="@string/login_progress_signing_in"
        android:textAppearance="?android:attr/textAppearanceMedium" />
</LinearLayout>

<!-- Login form -->
<ScrollView
    android:id="@+id/login_form"
    android:layout_width="match_parent"
    android:layout_height="match_parent" >

    <LinearLayout
        style="@style/LoginFormContainer"
        android:orientation="vertical" >

        <EditText
            android:id="@+id/email"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:hint="@string/prompt_email"
            android:inputType="textEmailAddress"
            android:maxLines="1"
            android:singleLine="true" />

        <EditText
            android:id="@+id/password"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:hint="@string/prompt_password"
            android:imeActionId="@+id/login"
            android:imeActionLabel="@string/action_sign_in_short"
            android:imeOptions="actionUnspecified"
            android:inputType="textPassword"
            android:maxLines="1"
            android:singleLine="true" />

        <Button
            android:id="@+id/sign_in_button"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="right"
            android:layout_marginTop="16dp"
            android:paddingLeft="32dp"
            android:paddingRight="32dp"
            android:text="@string/login_go" />

        <Button
            android:id="@+id/register_button"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="right"
            android:layout_marginTop="16dp"
            android:paddingLeft="32dp"
            android:paddingRight="32dp"
            android:text="@string/login_register" />

        <TextView
            android:id="@+id/forgot_password_ET"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginTop="40dp"
            android:layout_margin="3dp"
            android:layout_gravity="right"
            android:text="@string/text_forgot_password"/>

    </LinearLayout>
</ScrollView>

</merge>
activity_register.xml
-----
<merge xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    tools:context=".RegisterActivity" >

```

```

<!-- Login progress -->
<LinearLayout
    android:id="@+id/register_status"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:gravity="center_horizontal"
    android:orientation="vertical"
    android:visibility="gone" >

    <ProgressBar
        style="?android:attr/progressBarStyleLarge"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginBottom="8dp" />

    <TextView
        android:id="@+id/register_status_message"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginBottom="16dp"
        android:fontFamily="sans-serif-light"
        android:text="@string/register_progress_creating_account"
        android:textAppearance="?android:attr/textAppearanceMedium" />
</LinearLayout>

<!-- Login form -->
<ScrollView
    android:id="@+id/register_form"
    android:layout_width="match_parent"
    android:layout_height="match_parent" >

    <LinearLayout
        style="@style/LoginFormContainer"
        android:orientation="vertical" >

        <EditText
            android:id="@+id/reg_full_name"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:hint="@string/prompt_fullname"
            android:inputType="text"
            android:maxLines="1"
            android:singleLine="true" />

        <EditText
            android:id="@+id/reg_user_name"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:hint="@string/prompt_username"
            android:inputType="text"
            android:maxLines="1"
            android:singleLine="true" />

        <EditText
            android:id="@+id/reg_email"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:hint="@string/prompt_email"
            android:inputType="textEmailAddress"
            android:maxLines="1"
            android:singleLine="true" />

        <EditText
            android:id="@+id/reg_contact_no"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:hint="@string/prompt_contact_no"
            android:imeActionId="@+id/login"
            android:imeActionLabel="@string/action_sign_in_short"
            android:imeOptions="actionUnspecified"
            android:inputType="text"
            android:maxLines="1"
            android:singleLine="true" />

        <EditText
            android:id="@+id/reg_user_address"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:hint="@string/prompt_user_address"
            android:imeActionId="@+id/login"
            android:imeActionLabel="@string/action_sign_in_short"
            android:imeOptions="actionUnspecified"
            android:inputType="text"
            android:maxLines="1"
            android:singleLine="true" />

        <EditText
            android:id="@+id/reg_password"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:hint="@string/prompt_password"
            android:imeActionId="@+id/login"
            android:imeActionLabel="@string/action_sign_in_short"
            android:imeOptions="actionUnspecified"
            android:inputType="textPassword"

```

```

        android:maxLines="1"
        android:singleLine="true" />

        <EditText
            android:id="@+id/reg_confirm_password"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:hint="@string/prompt_confirm_password"
            android:imeActionId="@+id/login"
            android:imeActionLabel="@string/action_sign_in_short"
            android:imeOptions="actionUnspecified"
            android:inputType="textPassword"
            android:maxLines="1"
            android:singleLine="true" />

        <Button
            android:id="@+id/create_account_button"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="right"
            android:layout_marginTop="16dp"
            android:paddingLeft="32dp"
            android:paddingRight="32dp"
            android:text="@string/action_create_account" />
    </LinearLayout>
</ScrollView>

</merge>

```

```

custom_infowindow.xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:orientation="vertical" >

    <TextView
        android:id="@+id/tv_marker_title"
        android:layout_marginTop="3dp"
        android:layout_marginBottom="3dp"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:gravity="center"
        android:background="@drawable/marker_title"
        android:textAppearance="?android:attr/textAppearanceMedium" />

    <TextView
        android:id="@+id/tv_snippet1"
        android:singleLine="false"
        android:layout_margin="3dp"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:gravity="center"
        android:textAppearance="?android:attr/textAppearanceSmall" />

    <TextView
        android:id="@+id/tv_snippet2"
        android:layout_width="match_parent"
        android:layout_margin="3dp"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceSmall" />

    <LinearLayout
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:orientation="horizontal">

        <ImageView
            android:id="@+id/iv_snippet3"
            android:layout_width="0dp"
            android:layout_weight="0.5"
            android:layout_height="wrap_content"
            android:layout_margin="3dp"
            android:scaleType="fitStart"
            android:maxWidth="125dp"
            android:maxHeight="125dp"
            android:adjustViewBounds="true" />

        <TextView
            android:id="@+id/tv_snippet4"
            android:layout_width="0dp"
            android:layout_weight="0.5"
            android:layout_height="wrap_content"
            android:layout_margin="3dp"
            android:textAppearance="?android:attr/textAppearanceSmall"/>

    </LinearLayout>
</LinearLayout>

```

```

list_places.xml
<?xml version="1.0" encoding="utf-8"?>
<TextView xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:padding="4dp"

```

```
android:textAppearance="?android:attr/textAppearanceMedium" />
```

The following codes refer to the controller and utility classes of the Android Application

MainActivity.java

```
package com.patricksp.mobimapp;

import java.io.IOException;
import java.io.StringReader;
import java.net.MalformedURLException;
import java.net.URL;
import java.sql.Timestamp;
import java.util.ArrayList;
import java.util.Hashtable;
import java.util.List;
import java.util.Locale;
import java.util.concurrent.atomic.AtomicInteger;

import org.apache.http.NameValuePair;
import org.apache.http.message.BasicNameValuePair;
import org.json.JSONObject;

import com.google.android.gms.gcm.GoogleCloudMessaging;
import com.google.android.gms.maps.CameraUpdate;
import com.google.android.gms.maps.CameraUpdateFactory;
import com.google.android.gms.maps.GoogleMap;
import com.google.android.gms.maps.GoogleMap.InfoWindowAdapter;
import com.google.android.gms.maps.GoogleMap.OnMarkerClickListener;
import com.google.android.gms.maps.MapFragment;
import com.google.android.gms.maps.SupportMapFragment;
import com.google.android.gms.maps.model.BitmapDescriptor;
import com.google.android.gms.maps.model.BitmapDescriptorFactory;
import com.google.android.gms.maps.model.LatLng;
import com.google.android.gms.maps.model.Marker;
import com.google.android.gms.maps.model.MarkerOptions;
import com.google.android.gms.maps.model.PolylineOptions;
import com.google.gson.Gson;
import com.google.gson.stream.JsonReader;
import com.nostra13.universalimageloader.core.ImageLoader;
import com.nostra13.universalimageloader.core.ImageLoaderConfiguration;
import com.nostra13.universalimageloader.core.assist.FailReason;
import com.nostra13.universalimageloader.core.assist.ImageLoadingListener;
import com.patricksp.dbobjects.Disaster;
import com.patricksp.dbobjects.JSONParser;
import com.patricksp.utilities.ConnectionDetector;
import com.patricksp.utilities.GPSTracker;
import com.patricksp.utilities.GcmUtils;
import com.patricksp.utilities.MapUtilities;
import com.patricksp.utilities.Config;
import com.patricksp.utilities.ServerUtils;

import android.location.Address;
import android.location.Geocoder;
import android.os.AsyncTask;
import android.os.Bundle;
import android.annotation.SuppressLint;
import android.app.Activity;
import android.app.AlertDialog;
import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.content.IntentFilter;
import android.content.SharedPreferences;
import android.content.pm.PackageInfo;
import android.content.pm.PackageManager.NameNotFoundException;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.graphics.Color;
import android.support.v4.app.FragmentActivity;
import android.support.v4.app.FragmentManager;
import android.util.Log;
import android.view.Menu;
import android.view.MenuItem;
import android.view.MenuItem.OnMenuItemClickListener;
import android.view.View;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.AutoCompleteTextView;
import android.widget.Filter;
import android.widget.Filterable;
import android.widget.ImageButton;
import android.widget.ImageView;
import android.widget.ProgressBar;
import android.widget.TextView;
import android.widget.Toast;

public class MainActivity extends FragmentActivity {

    private AutoCompleteTextView autocomp_search;
    private ProgressBar progress_bar_search;
    private ImageButton button_my_location;
    private ImageButton button_refresh;
    private ImageButton button_search;

    private TextView tv_marker_title;
    private TextView tv_snippet1;
```

```

private TextView tv_snippet2;
private ImageView iv_snippet3;
private TextView tv_snippet4;

private GoogleMap mMap;
private GPSTracker gps;
private Marker prevMarker = null;

private Hashtable<String, String> markers1;
private Hashtable<String, String> markers2;
private Hashtable<String, String> markers3;
private Hashtable<String, Bitmap> markers4;

GoogleCloudMessaging gcm;
AtomicInteger msgId = new AtomicInteger();
String gcmRegId;
public static final String PROPERTY_REG_ID = "registration_id";
private static final String PROPERTY_APP_VERSION = "appVersion";
private static final String PROPERTY_ON_SERVER_EXPIRATION_TIME =
    "onServerExpirationTimeMs";
/**
 * Default lifespan (7 days) of a reservation until it is considered expired.
 */
public static final long REGISTRATION_EXPIRY_TIME_MS = 1000 * 3600 * 24 * 7;
static final String TAG = "GCMDemo";

private Context context = this;

ConnectionDetector cd;
private String toSearch;
@SuppressWarnings("NewApi")
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    FragmentManager fragmentManager = getSupportFragmentManager();
    SupportMapFragment mapFragment = (SupportMapFragment) fragmentManager
        .findFragmentById(R.id.map);
    mMap = mapFragment.getMap();

    CameraUpdate center=
        CameraUpdateFactory.newLatLng(new LatLng(13,
            122));

    mMap.moveCamera(center);
    CameraUpdate zoom=CameraUpdateFactory.zoomTo(6);
    mMap.animateCamera(zoom);

    cd = new ConnectionDetector(getApplicationContext());
    // Check if Internet present
    if(!cd.isConnectingToInternet()) {
        // Internet Connection is not present
        Toast.makeText(this,
            "Internet Connection Error. Please connect to working Internet connection", Toast.LENGTH_LONG);
        // stop executing code by return
        // return;
    }
    else{

        context = getApplicationContext();

        if (getGCMPreferences(context)!= null){

            SharedPreferences myPrefs = getGCMPreferences(context);
            String email = myPrefs.getString(Config.TAG.SHPR_EMAIL, "");
            Log.v("email", email);
            if (email!=null && !email.isEmpty()){
                LoginActivity.mEmail = email;
                Log.v("GCMPreference", "not null");
                gcmRegId = getRegistrationId(context);
                Log.v("GCM REG ID", gcmRegId);
                if (gcmRegId.length() == 0) {
                    registerBackground();
                }
                gcm = GoogleCloudMessaging.getInstance(this);
            }
        }
        else{
            Log.v("email", "null");
        }
    }

    markers1 = new Hashtable<String, String>();
    markers2 = new Hashtable<String, String>();
    markers3 = new Hashtable<String, String>();
    markers4 = new Hashtable<String, Bitmap>();

    mMap.setOnMarkerClickListener(new OnMarkerClickListener(){

        @Override
        public boolean onMarkerClick(final Marker marker) {
            // TODO Auto-generated method stub
            // Load image, decode it to Bitmap and return Bitmap to callback
            if (markers4.get(marker.getId())==null)
                new BitMapLoader(marker.getId(), marker).execute(markers3.get(marker.getId()));

            mMap.setInfoWindowAdapter(new CustomInfoWindowAdapter());
        }
    });

```

```

        return false;
    }
});

progress_bar_search = (ProgressBar) findViewById(R.id.address_progress);
autocomp_search = (AutoCompleteTextView) findViewById(R.id.autocomp_search_ET);
autocomp_search.setAdapter(new PlacesAutoCompleteAdapter(this, R.layout.list_places));
autocomp_search.setOnItemClickListener(new OnItemClickListener() {
    @Override
    public void onItemClick(AdapterView<?> adapterView, View view, int position, long id) {
        toSearch = (String) adapterView.getItemAtPosition(position);
        // Toast.makeText(getApplicationContext(), str, Toast.LENGTH.SHORT).show();

        progress_bar_search.setVisibility(View.VISIBLE);
        button_search.setVisibility(View.GONE);
        (new GetLocationTask(view.getContext())).execute(toSearch);
    }
});

button_search = (ImageButton) findViewById(R.id.search_but);
button_search.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        // TODO Auto-generated method stub
        toSearch = autocomp_search.getText().toString();
        Log.v("to search", toSearch);
        if(toSearch.length()>2){
            progress_bar_search.setVisibility(View.VISIBLE);
            button_search.setVisibility(View.GONE);
            (new GetLocationTask(view.getContext())).execute(toSearch);
        }
    }
});

button_my_location = (ImageButton) findViewById(R.id.my_loc_but);
button_my_location.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View arg0) {
        // TODO Auto-generated method stub
        //try{
        gps = new GPSTracker(MainActivity.this);

        // check if GPS enabled
        if(gps.canGetLocation()){

            MapUtilities.getCurrUserLocation(getApplicationContext(), gps, mMap);
        }else{
            // can't get location
            // GPS or Network is not enabled
            // Ask user to enable GPS/network in settings
            gps.showSettingsAlert();
        }
        // }catch(Exception e){
        // e.printStackTrace();
        // Toast.makeText(getApplicationContext(), "Unable to get current location
        // }
    }
});

button_refresh = (ImageButton) findViewById(R.id.refresh_but);
button_refresh.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View arg0) {
        // TODO Auto-generated method stub
        mMap.clear();
        Toast.makeText(getApplicationContext(), "Reloading disaster map..", Toast.LENGTH
        new HazardDotsDisplay().execute();
    }
});

View markerView = getLayoutInflater().inflate(R.layout.custom_infowindow, null);
tv_snippet2 = (TextView) markerView.findViewById(R.id.tv_snippet2);
iv_snippet3 = (ImageView) markerView.findViewById(R.id.iv_snippet3);
tv_snippet4 = (TextView) markerView.findViewById(R.id.tv_snippet4);

Intent i = this.getIntent();
Bundle extras = i.getExtras();
if(extras!=null && !extras.isEmpty()){
    final String shelters = extras.getString("shelters", null);
    if(shelters!=null){

        AlertDialog.Builder dlgAlert = new AlertDialog.Builder(context);

        dlgAlert.setMessage("Display possible route to safe shelter?");
        dlgAlert.setTitle("Safe Shelter");
        dlgAlert.setPositiveButton("Ok",
            new DialogInterface.OnClickListener() {
                public void onClick(DialogInterface dialog, int which) {
                    gps = new GPSTracker(MainActivity.this);

                    String[] coords = shelters.split("|");
                    double coordX, coordY;
                    for(int i=0;i<coords.length;i++){

                        coordX = Double.parseDouble(coords[i].split(" ");
                        coordY = Double.parseDouble(coords[i].split(" ");

```



```

        mMap.addPolyline(new PolylineOptions()
            .add(new LatLng(gps.getLatitude(), gps.getLongitude())
                .width(5)
                .color(Color.RED));
    }
    });
    dlgAlert.setCancelable(true);
    dlgAlert.create().show();
}
}
new HazardDotsDisplay().execute();
}
}
// Initiating Menu XML file (menu.xml)
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    super.onCreateOptionsMenu(menu);
    getMenuInflater().inflate(R.menu.main, menu);
    return true;
}

public void logout(MenuItem item){
    SharedPreferences sp = getSharedPreferences(Config.TAG_SHPR_FILE,0);
    Log.v(Config.TAG_SHPR_EMAIL, sp.getString(Config.TAG_SHPR_EMAIL, ""));
    sp.edit().clear();
    sp.edit().commit();
    new AsyncTask<Void, Void, String>() {
        protected String doInBackground(Void... params) {
            try {
                if (gcm!=null){
                    gcm.unregister();
                    Log.v("GCM unregister", "device unregistered");
                }
            } catch (IOException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
            }
            return "";
        }
    }.execute(null, null, null);
}

public void loginUser(View view){
    Intent loginIntent = new Intent(this, LoginActivity.class);
    startActivity(loginIntent);
}

private class PlacesAutoCompleteAdapter extends ArrayAdapter<String> implements Filterable {
    private ArrayList<String> resultList;

    public PlacesAutoCompleteAdapter(Context context, int textViewResourceId) {
        super(context, textViewResourceId);
    }

    @Override
    public int getCount() {
        return resultList.size();
    }

    @Override
    public String getItem(int index) {
        return resultList.get(index);
    }

    @Override
    public Filter getFilter() {
        Filter filter = new Filter() {
            @Override
            protected FilterResults performFiltering(CharSequence constraint) {
                FilterResults filterResults = new FilterResults();
                if (constraint != null) {
                    // Retrieve the autocomplete results.
                    resultList = PlaceAutocomplete.autocomplete(constraint.toString());
                    // Log.v("places", resultList.get(0));
                    // Assign the data to the FilterResults
                    filterResults.values = resultList;
                    filterResults.count = resultList.size();
                }
                return filterResults;
            }
        };
    }

    @Override
    protected void publishResults(CharSequence constraint, FilterResults results) {
        if (results != null && results.count > 0) {
            notifyDataSetChanged();
        } else {
            notifyDataSetInvalidated();
        }
    }
};
return filter;
}
}

```

```

}

private class GetLocationTask extends AsyncTask <String , Void , LatLng>{

private Context context;
String addressLine = "";

public GetLocationTask(Context context){

    this.context = context;
    addressLine = "";
}

@Override
protected LatLng doInBackground(String... names) {
// TODO Auto-generated method stub
Geocoder geocoder = new Geocoder(context , Locale.getDefault());
List<Address> addresses = null;
Address address = null;
LatLng latlng = null;

try {
    addresses = geocoder.getFromLocationName(names[0] , 1);
    int i=0;
    while (addresses.size()==0 && i++!=5) {
        addresses = geocoder.getFromLocationName(names[0] , 1);
        Log.v("attempting" , "attempting "+i);
    }
} catch (IOException e) {
// TODO Auto-generated catch block
Log.v("geocoder" , "unable to get addresses");
e.printStackTrace();
}

try{
    if(address==null || addresses.size()<1){
        List<NameValuePair> params = new ArrayList<NameValuePair>();
//
        address=$address&sensor=false&region=$region
        params.add(new BasicNameValuePair(" address" , names[0]));
        params.add(new BasicNameValuePair(" sensor" , " false"));
        params.add(new BasicNameValuePair(" sensor" , " ph"));
        params.add(new BasicNameValuePair(" key" , Config.APLKEY));

        try{
            JSONParser jsonParser = new JSONParser();

            JSONObject json = jsonParser.makeHttpRequest(Config.URL.GEOCODE

            Log.v(" json char count" , json.toString());

            Double lat = json.getJSONArray(" results").getJSONObject(0).
                getJSONObject(" geometry").getJSONObject(" geometry").getDouble(" lat");
            Double lng = json.getJSONArray(" results").getJSONObject(0).
                getJSONObject(" geometry").getJSONObject(" location").getDouble(" lng");

            latlng = new LatLng(lat , lng);
        }catch(Exception e){
            e.printStackTrace();
            Log.v("error dito" , "error");
        }
    }
    else{
        address = addresses.get(0);
        addressLine = address.getAddressLine(0);
    }
} catch (Exception e) {
// TODO Auto-generated catch block
Log.v("geocoder" , "unable to get addresses");
e.printStackTrace();
}

return latlng;
}

@Override
protected void onPostExecute(LatLng latLng){

progress_bar_search.setVisibility(View.GONE);
button_search.setVisibility(View.VISIBLE);

if(latLng!=null){

    CameraUpdate center=
        CameraUpdateFactory.newLatLng(new LatLng(latLng.latitude ,
            latLng.longitude));

mMap.moveCamera(center);
CameraUpdate zoom=CameraUpdateFactory.zoomTo(12);
mMap.animateCamera(zoom);

    Marker marker = MapUtilities.drawMarker(mMap, latLng.latitude ,
        latLng.longitude , addressLine ,
        toSearch , 0 , true);

if(prevMarker != null){
    prevMarker.remove();
}
prevMarker = marker;
}
}

```

```

    }
    else{
        Toast.makeText(context, "Unable to get location. Try again later", Toast.LENGTH
    }
}

private class HazardDotsDisplay extends AsyncTask<Void, Void, Boolean>{
    private Disaster[] currDisasters = null;
    private String error;

    @Override
    protected Boolean doInBackground(Void... arg0) {
        // TODO Auto-generated method stub
        List<NameValuePair> params = new ArrayList<NameValuePair>();
        // params.add(new BasicNameValuePair("test", "wew"));
        try{
            JSONParser jsonParser = new JSONParser();
            // getting product details by making HTTP request
            // Note that product details url will use GET request
            String json = jsonParser.getJSONResultString(
                Config.URL.DISASTERS, "GET", params);
            Log.v("json char count", json.toString());
            Gson jsonToObject = new Gson();
            JsonReader reader = new JsonReader(new StringReader(json));
            reader.setLenient(true);

            if(json!=null){
                currDisasters = jsonToObject.fromJson(reader, Disaster[].class);
                error = (currDisasters.length < 1) ? "No disasters reported at the moment." : "
            }
        }
        catch(Exception e){
            e.printStackTrace();
            error = "Connection related error...";
        }

        return (currDisasters!=null && currDisasters.length > 0);
    }

    @Override
    protected void onPostExecute(Boolean hasResults){
        if(hasResults){
            MarkerOptions markerOpts;
            BitmapDescriptor icon = null;
            Marker marker;
            Bitmap bm=null;

            for(int i=0;i<currDisasters.length;i++){
                markerOpts = new MarkerOptions();

                if(currDisasters[i].disaster_type.equals("flood")){
                    icon = BitmapDescriptorFactory.fromResource(R.drawable.flood);
                }
                else if(currDisasters[i].disaster_type.equals("earthquake")){
                    icon = BitmapDescriptorFactory.fromResource(R.drawable.earthquake);
                }
                else if(currDisasters[i].disaster_type.equals("landslide")){
                    icon = BitmapDescriptorFactory.fromResource(R.drawable.landslide);
                }
                else if(currDisasters[i].disaster_type.equals("tsunami")){
                    icon = BitmapDescriptorFactory.fromResource(R.drawable.tsunami);
                }
                else if(currDisasters[i].disaster_type.equals("volcanic activity")){
                    icon = BitmapDescriptorFactory.fromResource(R.drawable.earthquake);
                }
                else if(currDisasters[i].disaster_type.equals("fire")){
                    icon = BitmapDescriptorFactory.fromResource(R.drawable.fire);
                }
            }

            markerOpts.position(new LatLng(currDisasters[i].location_x, currDisasters[i].location_y));
            markerOpts.title(currDisasters[i].disaster_type);
            markerOpts.icon(icon);
            markerOpts.snippet(String.format("%s located at %s", currDisasters[i].disaster_type,
                currDisasters[i].disaster_location));

            marker = mMap.addMarker(markerOpts);
            markers1.put(marker.getId(), new StringBuilder("Reported ").append(currDisasters[i].remarks));
            markers2.put(marker.getId(), currDisasters[i].remarks);
            markers3.put(marker.getId(), currDisasters[i].image);
        }
        else{
            Toast.makeText(context, error, Toast.LENGTHSHORT).show();
        }
    }
}

private class BitMapLoader extends AsyncTask<String, Void, Boolean>{
    private String hashkey;
    private Bitmap bmp;

```

```

private Marker marker;

public BitMapLoader(String hashkey, Marker marker){

    this.hashkey = hashkey;
    this.marker = marker;
}

@Override
protected Boolean doInBackground(String... imageUrl) {
    // TODO Auto-generated method stub
    URL url;
    try {
        url = new URL(imageUrl[0]);
        bmp = BitmapFactory.decodeStream(url.openConnection().getInputStream());
    } catch (Exception e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
        return false;
    }

    return true;
}

@Override
protected void onPostExecute(Boolean success){

    if(success){
        markers4.put(hashkey, bmp);
        marker.hideInfoWindow();
        marker.showInfoWindow();
    }
}

private class CustomInfoWindowAdapter implements InfoWindowAdapter{

    @Override
    public View getInfoContents(Marker marker) {
        // TODO Auto-generated method stub

        return null;
    }

    @Override
    public View getInfoWindow(Marker marker) {
        // TODO Auto-generated method stub

        View myContentView = getLayoutInflater().inflate(R.layout.custom_infowindow, null);

        tv_marker_title = (TextView) myContentView.findViewById(R.id.tv_marker_title);
        tv_marker_title.setText(marker.getTitle());

        tv_snippet1 = (TextView) myContentView.findViewById(R.id.tv_snippet1);
        tv_snippet1.setText(marker.getSnippet());

        tv_snippet2 = (TextView) myContentView.findViewById(R.id.tv_snippet2);
        tv_snippet2.setText(markers1.get(marker.getId()));

        iv_snippet3 = (ImageView) myContentView.findViewById(R.id.iv_snippet3);

        Bitmap bm = markers4.get(marker.getId());
        if(bm!=null)
            iv_snippet3.setImageBitmap(bm);
        else
            iv_snippet3.setImageResource(R.drawable.ic_launcher);

        tv_snippet4 = (TextView) myContentView.findViewById(R.id.tv_snippet4);
        tv_snippet4.setText(markers2.get(marker.getId()));

        myContentView.setBackgroundColor(Color.WHITE);
        myContentView.setPadding(3, 3, 3, 3);
        return myContentView;
    }
}

/**
 * Stores the registration id, app versionCode, and expiration time in the application's
 * {@code SharedPreferences}.
 *
 * @param context application's context.
 * @param regId registration id
 */
private void setRegistrationId(Context context, String regId) {
    final SharedPreferences prefs = getGCMPreferences(context);
    int appVersion = getAppVersion(context);
    Log.v(TAG, "Saving regId on app version " + appVersion);
    SharedPreferences.Editor editor = prefs.edit();
    editor.putString(PROPERTY_REG_ID, regId);
    editor.putInt(PROPERTY_APP_VERSION, appVersion);
    long expirationTime = System.currentTimeMillis() + REGISTRATION_EXPIRY_TIME_MS;

    Log.v(TAG, "Setting registration expiry time to " +
        new Timestamp(expirationTime));
    editor.putLong(PROPERTY_ON_SERVER_EXPIRATION_TIME, expirationTime);
    editor.commit();
}

/**
 * Registers the application with GCM servers asynchronously.

```

```

* <p>
* Stores the registration id, app versionCode, and expiration time in the application's
* shared preferences.
*/

private void registerBackground() {

    final SharedPreferences prefs = getGCMPreferences(context);
    final String mEmail = prefs.getString(Config.TAG.SHPR.EMAIL, "");
    final String mUserName = prefs.getString(Config.TAG.SHPR.USERNAME, "");

    new AsyncTask<Void, Void, String>() {
        @Override
        protected String doInBackground(Void... params) {
            String msg = "";
            try {
                if (gcm == null) {
                    gcm = GoogleCloudMessaging.getInstance(context);
                }
                gcmRegId = gcm.register(GcmUtils.SENDER_ID);
                msg = "Device registered, registration id=" + gcmRegId;

                ServerUtils.register(context, mEmail, gcmRegId);

                // You should send the registration ID to your server over HTTP, so it
                // can use GCM/HTTP or CCS to send messages to your app.

                // For this demo: we don't need to send it because the device will send
                // upstream messages to a server that echo back the message using the
                // 'from' address in the message.

                // Save the regid - no need to register again.
                setRegistrationId(context, gcmRegId);
            } catch (IOException ex) {
                msg = "Error :" + ex.getMessage();
            }
            return msg;
        }

        @Override
        protected void onPostExecute(String msg) {
            // mDisplay.append(msg + "\n");
        }
    }.execute(null, null, null);
}

/**
 * Gets the current registration id for application on GCM service.
 * <p>
 * If result is empty, the registration has failed.
 *
 * @return registration id, or empty string if the registration is not
 * complete.
 */
private String getRegistrationId(Context context) {
    final SharedPreferences prefs = getGCMPreferences(context);
    String registrationId = prefs.getString(Property.REG_ID, "");
    if (registrationId.length() == 0) {
        Log.v(TAG, "Registration not found.");
        return "";
    }
    // check if app was updated; if so, it must clear registration id to
    // avoid a race condition if GCM sends a message
    int registeredVersion = prefs.getInt(Property.APP_VERSION, Integer.MIN_VALUE);
    int currentVersion = getAppVersion(context);
    if (registeredVersion != currentVersion || isRegistrationExpired()) {
        Log.v(TAG, "App version changed or registration expired.");
        return "";
    }
    return registrationId;
}

/**
 * @return Application's {@code SharedPreferences}.
 */
private SharedPreferences getGCMPreferences(Context context) {
    return getSharedPreferences(Config.TAG.SHPR.FILE, Context.MODE_PRIVATE);
}

/**
 * Checks if the registration has expired.
 *
 * <p>To avoid the scenario where the device sends the registration to the
 * server but the server loses it, the app developer may choose to re-register
 * after REGISTRATION_EXPIRY_TIME_MS.
 *
 * @return true if the registration has expired.
 */
private boolean isRegistrationExpired() {
    final SharedPreferences prefs = getGCMPreferences(context);
    // checks if the information is not stale
    long expirationTime =
        prefs.getLong(Property.ON_SERVER_EXPIRATION_TIME, -1);
    return System.currentTimeMillis() > expirationTime;
}

/**
 * @return Application's version code from the {@code PackageManager}.
 */

```

```

private static int getAppVersion(Context context) {
    try {
        PackageInfo packageInfo = context.getPackageManager()
            .getPackageInfo(context.getPackageName(), 0);
        return packageInfo.versionCode;
    } catch (NameNotFoundException e) {
        // should never happen
        throw new RuntimeException("Could not get package name: " + e);
    }
}

@Override
protected void onDestroy() {
    super.onDestroy();
    try {
        // unregisterReceiver (mHandleMessageReceiver);
        // GCMRegistrar.onDestroy (this);
    } catch (Exception e) {
        Log.e("UnRegister Receiver Error", "> " + e.getMessage());
    }
}
}

```

LoginActivity.java

```

package com.patricksp.mobimapp;

import java.io.IOException;
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;
import java.util.ArrayList;
import java.util.List;

import org.apache.commons.validator.routines.EmailValidator;
import org.apache.http.NameValuePair;
import org.apache.http.message.BasicNameValuePair;
import org.json.JSONException;
import org.json.JSONObject;
import org.w3c.dom.Text;

import com.google.android.gms.gcm.GoogleCloudMessaging;
import com.patricksp.dbojects.JSONParser;
import com.patricksp.utilities.Config;
import com.patricksp.utilities.GcmUtils;
import com.patricksp.utilities.ServerUtils;
import com.patricksp.utilities.Utilities;

import android.animation.Animator;
import android.animation.AnimatorListenerAdapter;
import android.annotation.SuppressLint;
import android.annotation.TargetApi;
import android.app.Activity;
import android.app.AlertDialog;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.content.SharedPreferences;
import android.os.AsyncTask;
import android.os.Build;
import android.os.Bundle;
import android.text.TextUtils;
import android.util.Log;
import android.view.KeyEvent;
import android.view.Menu;
import android.view.MenuItem;
import android.view.MenuItem.OnMenuItemClickListener;
import android.view.inputmethod.EditorInfo;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

/**
 * Activity which displays a login screen to the user, offering registration as
 * well.
 */
public class LoginActivity extends Activity {

    private static final String TAG_STATUS = "status";
    private static final String TAG_MESSAGE = "message";
    public static final String TAG_EMAIL = "email";
    public static final String TAG_USERNAME = "username";
    public static final String TAG_PASSWORD = "password";
    public static final String TAG_USERID = "user_ID";

    JSONParser jsonParser = new JSONParser();

    /**
     * The default email to populate the email field with.
     */
    public static final String EXTRA_EMAIL = "com.example.android.authenticatordemo.extra.EMAIL";

    /**
     * Keep track of the login task to ensure we can cancel it if requested.
     */
    private UserLoginTask mAuthTask = null;

    // Values for email and password at the time of the login attempt.
    private int mUserID;
    public static String mEmail;
    private String mUsername;

```

```

private String mPassword;

// UI references.
private EditText mEmailView;
private EditText mPasswordView;
private View mLoginFormView;
private View mLoginStatusView;
private TextView mLoginStatusMessageView;

private Context context = this;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);

    setContentView(R.layout.activity_login);

    if (alreadyLoggedIn()) {
        Intent intent = new Intent(getApplicationContext(), DisasterReport.class);
        intent.putExtra(TAG_EMAIL, mEmail);
        intent.putExtra(TAG_USERNAME, mUsername);
        intent.putExtra(TAG_PASSWORD, mPassword);
        intent.putExtra(TAG_USERID, mUserID);
        startActivity(intent);
        finish();
    }

    // Set up the login form.
    mEmail = getIntent().getStringExtra(EXTRA_EMAIL);
    mEmailView = (EditText) findViewById(R.id.email);
    mEmailView.setText(mEmail);

    mPasswordView = (EditText) findViewById(R.id.password);
    mPasswordView
        .setOnEditorActionListener(new TextView.OnEditorActionListener() {
            @Override
            public boolean onEditorAction(TextView textView, int id,
                KeyEvent keyEvent) {
                if (id == R.id.login || id == EditorInfo.IME_NULL) {
                    attemptLogin();
                    return true;
                }
                return false;
            }
        });

    mLoginFormView = findViewById(R.id.login_form);
    mLoginStatusView = findViewById(R.id.login_status);
    mLoginStatusMessageView = (TextView) findViewById(R.id.login_status_message);

    findViewById(R.id.sign_in_button).setOnClickListener(
        new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                attemptLogin();
            }
        });

    findViewById(R.id.register_button).setOnClickListener(
        new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                // TODO Auto-generated method stub
                Intent regIntent = new Intent(v.getContext(), RegisterActivity.class);
                startActivity(regIntent);
            }
        });

    findViewById(R.id.forgot_password_ET).setOnClickListener(new View.OnClickListener() {

        @Override
        public void onClick(View arg0) {
            // TODO Auto-generated method stub
            final EditText txtEmail = new EditText(context);

            // Set the default text to a link of the Queen
            txtEmail.setHint("Enter Email..");

            AlertDialog.Builder builder = new AlertDialog.Builder(context);
            final GetPasswordTask task = new GetPasswordTask(builder);

            builder.setTitle("Forgot Password")
                .setView(txtEmail)
                .setPositiveButton("Get new password", new DialogInterface.OnClickListener() {
                    public void onClick(DialogInterface dialog, int whichButton) {
                        String email = txtEmail.getText().toString();

                        task.execute(email);
                        // moustachify(null, url);
                    }
                })
                .setNegativeButton("Cancel", new DialogInterface.OnClickListener() {
                    public void onClick(DialogInterface dialog, int whichButton) {
                        task.cancel(true);
                    }
                });
        }
    });
}

```

```

        }).show();
    }
}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    super.onCreateOptionsMenu(menu);
    getMenuInflater().inflate(R.menu.main, menu);

    return true;
}

public void logout(MenuItem item){

    SharedPreferences sp = getSharedPreferences(Config.TAG_SHPR_FILE,0);
    sp.edit().clear();
    sp.edit().commit();

}

/**
 * Attempts to sign in or register the account specified by the login form.
 * If there are form errors (invalid email, missing fields, etc.), the
 * errors are presented and no actual login attempt is made.
 */
// @SuppressWarnings("NewApi")
public boolean alreadyLoggedIn(){

    SharedPreferences sp = getSharedPreferences(Config.TAG_SHPR_FILE,0);

    String email = sp.getString(Config.TAG_SHPR_EMAIL, null);
    String pass = sp.getString(Config.TAG_SHPR_PASSWORD, null);
    String username = sp.getString(Config.TAG_SHPR_USERNAME, null);
    int userID = sp.getInt(Config.TAG_SHPR_USERID, -1);
    Log.v("user log in", email+">" + pass+">" + username);
    if (email!=null && pass!=null && username!=null && userID!=-1){

        mEmail = email;
        mPassword = pass;
        mUsername = username;
        mUserID = userID;

        return true;
    }

    return false;
}

public void attemptLogin() {
    if ( mAuthTask != null ) {
        return;
    }

    // Reset errors.
    mEmailView.setError(null);
    mPasswordView.setError(null);

    // Store values at the time of the login attempt.
    mEmail = mEmailView.getText().toString();
    mPassword = mPasswordView.getText().toString();

    boolean cancel = false;
    View focusView = null;

    // Check for a valid password.
    if (TextUtils.isEmpty(mPassword)) {
        mPasswordView.setError(Utilities.getSpannable(getString(R.string.error_field_required)));
        focusView = mPasswordView;
        cancel = true;
    } else if (mPassword.length() < 4) {
        mPasswordView.setError(Utilities.getSpannable(getString(R.string.error_invalid_password)));
        focusView = mPasswordView;
        cancel = true;
    }

    EmailValidator emailValidator = EmailValidator.getInstance();
    // Check for a valid email address.
    if (TextUtils.isEmpty(mEmail)) {
        mEmailView.setError(Utilities.getSpannable(getString(R.string.error_field_required)));
        focusView = mEmailView;
        cancel = true;
    } else if (!emailValidator.isValid(mEmail)) {
        mEmailView.setError(Utilities.getSpannable(getString(R.string.error_invalid_email)));
        focusView = mEmailView;
        cancel = true;
    }

    if (cancel) {
        // There was an error; don't attempt login and focus the first
        // form field with an error.
        focusView.requestFocus();
    } else {
        // Show a progress spinner, and kick off a background task to
        // perform the user login attempt.
        mLoginStatusMessageView.setText(R.string.login_progress_signing_in);
        showProgress(true, false);
    }
}

```



```

        mAuthTask = new UserLoginTask(this);
        mAuthTask.execute((Void) null);
    }
}

/**
 * Shows the progress UI and hides the login form.
 */
@TargetApi(Build.VERSION_CODES.HONEYCOMB_MR2)
private void showProgress(final boolean show, boolean forgot_password) {
    // On Honeycomb MR2 we have the ViewPropertyAnimator APIs, which allow
    // for very easy animations. If available, use these APIs to fade-in
    // the progress spinner.
    if(forgot_password){
        mLoginStatusMessageView.setText(R.string.process_forgot_password);
    }
    else{
        mLoginStatusMessageView.setText(R.string.login_progress_signing_in);
    }

    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.HONEYCOMB_MR2) {
        int shortAnimTime = getResources().getInteger(
            android.R.integer.config_shortAnimTime);

        mLoginStatusView.setVisibility(View.VISIBLE);
        mLoginStatusView.animate().setDuration(shortAnimTime)
            .alpha(show ? 1 : 0)
            .setListener(new AnimatorListenerAdapter() {
                @Override
                public void onAnimationEnd(Animator animation) {
                    mLoginStatusView.setVisibility(show ? View.VISIBLE
                        : View.GONE);
                }
            });

        mLoginFormView.setVisibility(View.VISIBLE);
        mLoginFormView.animate().setDuration(shortAnimTime)
            .alpha(show ? 0 : 1)
            .setListener(new AnimatorListenerAdapter() {
                @Override
                public void onAnimationEnd(Animator animation) {
                    mLoginFormView.setVisibility(show ? View.GONE
                        : View.VISIBLE);
                }
            });
    }
    else {
        // The ViewPropertyAnimator APIs are not available, so simply show
        // and hide the relevant UI components.
        mLoginStatusView.setVisibility(show ? View.VISIBLE : View.GONE);
        mLoginFormView.setVisibility(show ? View.GONE : View.VISIBLE);
    }
}

/**
 * Represents an asynchronous login/registration task used to authenticate
 * the user.
 */
public class UserLoginTask extends AsyncTask<Void, Void, Boolean> {

    Context context;
    String status = "";
    String message = "";

    public UserLoginTask(Context context){

        this.context = context;
        status = "";
        message = "";
    }

    @Override
    protected Boolean doInBackground(Void... param) {
        // TODO: attempt authentication against a network service.

        try {
            // Building Parameters
            List<NameValuePair> params = new ArrayList<NameValuePair>();
            params.add(new BasicNameValuePair(TAG_EMAIL, mEmail));
            params.add(new BasicNameValuePair(TAG_PASSWORD, mPassword));

            // getting product details by making HTTP request
            // Note that product details url will use GET request
            JSONObject json = jsonParser.makeHttpRequest(
                Config.URL_LOGIN_USER, "POST", params);

            // check your log for json response
            Log.v("ano ba?", ""+json.toString());
            // json success tag
            if(json!=null){

                status = json.getString(TAG_STATUS);
                Log.v("wewewewe", status);
                if(status.equals(Config.STATUS_OK)){
                    mUsername = json.getString(TAG_USERNAME);
                    mUserID = json.getInt(TAG_USERID);

                    GoogleCloudMessaging gcm = GoogleCloudMessaging.getInstance(context);

```

```

        String gcmRegId="";
        try {
            gcmRegId = gcm.register(GcmUtils.SENDER_ID);
            //ServerUtils.register(context, mUsername, mEmail, gcmRegId);
        } catch (IOException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }

        return true;
    }
    else{
        message = json.getString(TAG.MESSAGE);
        return false;
    }
} catch (JSONException e) {
    e.printStackTrace();
}

message = "Connection related error..";

// TODO: register the new account here.
return false;
}

@Override
protected void onPostExecute(final Boolean success) {
    mAuthTask = null;
    showProgress(false, false);

    if (success) {
        SharedPreferences sp=context.getSharedPreferences(Config.TAG.SHPR.FILE, Context
        SharedPreferences.Editor Ed=sp.edit();
        Ed.putString(Config.TAG.SHPR.EMAIL, mEmail);
        Ed.putString(Config.TAG.SHPR.USERNAME, mUsername);
        Ed.putInt(Config.TAG.SHPR.USERID, mUserID);

        try {
            byte[] bytesOfMessage = mPassword.getBytes("UTF-8");
            MessageDigest md;
            md = MessageDigest.getInstance("MD5");
            byte[] thedigest = md.digest(bytesOfMessage);
            Ed.putString(Config.TAG.SHPR.PASSWORD, new String(thedigest, "UTF-8"));
            Ed.commit();
        } catch (Exception e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }

        Intent intent = new Intent(context, DisasterReport.class);
        intent.putExtra(TAG.EMAIL, mEmail);
        intent.putExtra(TAG.USERNAME, mUsername);
        intent.putExtra(TAG.PASSWORD, mPassword);
        intent.putExtra(TAG.USERID, mUserID);
        startActivity(intent);
        finish();
    } else {
        if(TextUtils.equals(message," Incorrect Password!")){

            mPasswordView
                .setError(Utilities.getSpannable(getString(R.string.error_incorrect_pass
                mPasswordView.requestFocus());
            }
            else{

                Toast.makeText(context, message, Toast.LENGTHSHORT).show();
            }
        }
    }
}

@Override
protected void onCancelled() {
    mAuthTask = null;
    showProgress(false, false);
}
}

class GetPasswordTask extends AsyncTask<String, Void, String>{

    AlertDialog.Builder builder;

    public GetPasswordTask(AlertDialog.Builder builder){

        this.builder = builder;
    }

    @Override
    protected void onPreExecute(){
        showProgress(true, true);
    }

    @Override
    protected String doInBackground(String... email) {
        // TODO Auto-generated method stub

```

```

        String message = "";

        JSONParser jsonParser = new JSONParser();
        List<NameValuePair> params = new ArrayList<NameValuePair>();
        params.add(new BasicNameValuePair(TAG_EMAIL, email[0]));

        try{
            JSONObject json = jsonParser.makeHttpRequest(Config.URL_FORGOT_PASSWORD, "POST", params);

            if(json!=null){
                try {
                    message = json.getString(TAG_MESSAGE);
                    return message;
                } catch (JSONException e) {
                    // TODO Auto-generated catch block
                    e.printStackTrace();
                }
            }
        } catch (Exception e){
            e.printStackTrace();
        }

        return Config.STATUS_FAIL;
    }

    @Override
    protected void onPostExecute(String message){
        showProgress(false, false);
        AlertDialog.Builder builder = new AlertDialog.Builder(context);

        builder.setTitle("Forgot Password")
            .setMessage(message)
            .setPositiveButton("Ok", null)
            .show();
    }
}

```

RegisterActivity.java

```

package com.patricksp.mobimapp;

import java.util.ArrayList;
import java.util.List;
import java.util.regex.Pattern;

import org.apache.commons.validator.routines.EmailValidator;
import org.apache.http.NameValuePair;
import org.apache.http.message.BasicNameValuePair;
import org.json.JSONException;
import org.json.JSONObject;

import com.google.android.gcm.GCMRegistrar;
import com.patricksp.dbojects.JSONParser;
import com.patricksp.utilities.GcmUtils;
import com.patricksp.utilities.Config;
import com.patricksp.utilities.ServerUtils;
import com.patricksp.utilities.Utilities;

import android.animation.Animator;
import android.animation.AnimatorListenerAdapter;
import android.annotation.TargetApi;
import android.app.Activity;
import android.app.AlertDialog;
import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.content.IntentFilter;
import android.os.AsyncTask;
import android.os.Build;
import android.os.Bundle;
import android.text.SpannableString;
import android.text.TextUtils;
import android.text.style.ForegroundColorSpan;
import android.util.Log;
import android.view.KeyEvent;
import android.view.Menu;
import android.view.View;
import android.view.inputmethod.EditorInfo;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

/**
 * Activity which displays a register screen to the user, offering registration as
 * well.
 */
public class RegisterActivity extends Activity {

    private static final String TAG_STATUS = "status";
    private static final String TAG_MESSAGE = "message";
    private static final String TAG_EMAIL = "email";
    private static final String TAG_USERNAME = "username";
    private static final String TAG_PASSWORD = "password";

```

```

private static final String TAG_CONTACT = "contact";
private static final String TAG_ADDRESS = "address";
private static final String TAG_FULLNAME = "fullname";

JSONParser jsonParser = new JSONParser();
/**
 * The default email to populate the email field with.
 */
public static final String EXTRA_EMAIL = "com.example.android.authenticatordemo.extra.EMAIL";

/**
 * Keep track of the register task to ensure we can cancel it if requested.
 */
private UserRegisterTask mAuthTask = null;

// Values for email and password at the time of the register attempt.
private String mFullName;
public static String mUserName;
public static String mEmail;
private String mContactNo;
private String mAddress;
private String mPassword;
private String mConfirmPassword;
private String gcmRegId;

// UI references.
private EditText mFullNameView;
private EditText mUserNameView;
private EditText mEmailView;
private EditText mContactNoView;
private EditText mAddressView;
private EditText mPasswordView;
private EditText mConfirmPasswordView;
private View mRegisterFormView;
private View mRegisterStatusView;
private TextView mRegisterStatusMessageView;

private Context context = this;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);

    setContentView(R.layout.activity_register);

    // Set up the register form.
    mFullNameView = (EditText) findViewById(R.id.reg_full_name);
    mUserNameView = (EditText) findViewById(R.id.reg_user_name);

    mEmail = getIntent().getStringExtra(EXTRA_EMAIL);
    mEmailView = (EditText) findViewById(R.id.reg_email);
    mEmailView.setText(mEmail);

    mContactNoView = (EditText) findViewById(R.id.reg_contact_no);
    mAddressView = (EditText) findViewById(R.id.reg_user_address);
    mPasswordView = (EditText) findViewById(R.id.reg_password);

    mConfirmPasswordView = (EditText) findViewById(R.id.reg_confirm_password);
    mConfirmPasswordView
        .setOnEditorActionListener(new TextView.OnEditorActionListener() {
            @Override
            public boolean onEditorAction(TextView textView, int id,
                KeyEvent keyEvent) {
                if (id == R.id.login || id == EditorInfo.IME_NULL) {
                    attemptRegister();
                    return true;
                }
                return false;
            }
        });

    mRegisterFormView = findViewById(R.id.register_form);
    mRegisterStatusView = findViewById(R.id.register_status);
    mRegisterStatusMessageView = (TextView) findViewById(R.id.register_status_message);

    findViewById(R.id.create_account_button).setOnClickListener(
        new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                attemptRegister();
            }
        });
}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    super.onCreateOptionsMenu(menu);
    getMenuInflater().inflate(R.menu.register, menu);
    return true;
}

/**
 * Attempts to sign in or register the account specified by the register form.
 * If there are form errors (invalid email, missing fields, etc.), the
 * errors are presented and no actual register attempt is made.
 */
public void attemptRegister() {
    if (mAuthTask != null) {

```

```

        return;
    }

    // Reset errors.
    mEmailView.setError(null);
    mPasswordView.setError(null);
    mFullNameView.setError(null);
    mUserNameView.setError(null);
    mConfirmPasswordView.setError(null);

    // Store values at the time of the register attempt.
    mContactNo = mContactNoView.getText().toString();
    mAddress = mAddressView.getText().toString();
    mFullName = mFullNameView.getText().toString();
    mUserName = mUserNameView.getText().toString();
    mEmail = mEmailView.getText().toString();
    mPassword = mPasswordView.getText().toString();
    mConfirmPassword = mConfirmPasswordView.getText().toString();

    boolean cancel = false;
    View focusView = null;

    // Check for a valid password.
    if (TextUtils.isEmpty(mPassword)) {
        mPasswordView.setError(Utilities.getSpannable(getString(R.string.error_field_required)));
        focusView = mPasswordView;
        cancel = true;
    } else if (mPassword.length() < 4) {
        mPasswordView.setError(Utilities.getSpannable(getString(R.string.error_invalid_password)));
        focusView = mPasswordView;
        cancel = true;
    }

    //Check if passwords do not match
    if (!mConfirmPassword.equals(mPassword)){
        mConfirmPasswordView.setError(Utilities.getSpannable(getString(R.string.error_password_mismatch)));
        focusView = mConfirmPasswordView;
        cancel = true;
    }

    EmailValidator emailValidator = EmailValidator.getInstance();

    // Check for a valid email address.
    if (TextUtils.isEmpty(mEmail)) {
        mEmailView.setError(Utilities.getSpannable(getString(R.string.error_field_required)));
        focusView = mEmailView;
        cancel = true;
    } else if (!emailValidator.isValid(mEmail)) {
        mEmailView.setError(Utilities.getSpannable(getString(R.string.error_invalid_email)));
        focusView = mEmailView;
        cancel = true;
    }

    //check if username filled
    if (TextUtils.isEmpty(mUserName)) {
        mUserNameView.setError(Utilities.getSpannable(getString(R.string.error_field_required)));
        focusView = mUserNameView;
        cancel = true;
    }

    //check if full name filled
    if (TextUtils.isEmpty(mFullName)) {
        mFullNameView.setError(Utilities.getSpannable(getString(R.string.error_field_required)));
        focusView = mFullNameView;
        cancel = true;
    }

    // GCMRegistrar.checkDevice(this);

    // Make sure the manifest was properly set - comment out this line
    // while developing the app, then uncomment it when it's ready.
    // GCMRegistrar.checkManifest(this);

    // registerReceiver(mHandleMessageReceiver, new IntentFilter(
    // GcmUtils.DISPLAY_MESSAGE_ACTION));

    // Get GCM registration id
    // gcmRegId = GCMRegistrar.getRegistrationId(this);

    // Check if regid already presents
    // if (gcmRegId.equals("")) {
    //     // Registration is not present, register now with GCM
    //     GCMRegistrar.register(getApplicationContext(), GcmUtils.SENDER_ID);
    // } else {
    //     // Device is already registered on GCM
    //     if (GCMRegistrar.isRegisteredOnServer(this)) {
    //         // Skips registration.
    //         Toast.makeText(getApplicationContext(), "Already registered with GCM", Toast.LENGTH_LONG).show();
    //         cancel = true;
    //     }
    // }

    if (cancel) {
        // There was an error; don't attempt register and focus the first
        // form field with an error.
        focusView.requestFocus();
    } else {
        // Show a progress spinner, and kick off a background task to
        // perform the user register attempt.
    }

```

```

        mRegisterStatusMessageView.setText(R.string.register_progress_creating_account);
        showProgress(true);
        context = this;
        mAuthTask = new UserRegisterTask();
        mAuthTask.execute((Void) null);
    }
}

/**
 * Shows the progress UI and hides the register form.
 */
@TargetApi(Build.VERSION_CODES.HONEYCOMB_MR2)
private void showProgress(final boolean show) {
    // On Honeycomb MR2 we have the ViewPropertyAnimator APIs, which allow
    // for very easy animations. If available, use these APIs to fade-in
    // the progress spinner.
    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.HONEYCOMB_MR2) {
        int shortAnimTime = getResources().getInteger(
            android.R.integer.config_shortAnimTime);

        mRegisterStatusView.setVisibility(View.VISIBLE);
        mRegisterStatusView.animate().setDuration(shortAnimTime)
            .alpha(show ? 1 : 0)
            .setListener(new AnimatorListenerAdapter() {
                @Override
                public void onAnimationEnd(Animator animation) {
                    mRegisterStatusView.setVisibility(show ? View.VISIBLE
                        : View.GONE);
                }
            });

        mRegisterFormView.setVisibility(View.VISIBLE);
        mRegisterFormView.animate().setDuration(shortAnimTime)
            .alpha(show ? 0 : 1)
            .setListener(new AnimatorListenerAdapter() {
                @Override
                public void onAnimationEnd(Animator animation) {
                    mRegisterFormView.setVisibility(show ? View.GONE
                        : View.VISIBLE);
                }
            });
    } else {
        // The ViewPropertyAnimator APIs are not available, so simply show
        // and hide the relevant UI components.
        mRegisterStatusView.setVisibility(show ? View.VISIBLE : View.GONE);
        mRegisterFormView.setVisibility(show ? View.GONE : View.VISIBLE);
    }
}

/**
 * Represents an asynchronous registration task used to authenticate
 * the user.
 */
private class UserRegisterTask extends AsyncTask<Void, Void, Boolean> {

    private String reg_message = "";
    private String reg_status = "";

    UserRegisterTask(){
        reg_message = "";
        reg_status = "";
    }

    @Override
    protected Boolean doInBackground(Void... param) {
        // TODO: attempt authentication against a network service.

        try {
            // ServerUtils.register(context, mUserName, mEmail, gcmRegId);
            // Building Parameters
            List<NameValuePair> params = new ArrayList<NameValuePair>();
            params.add(new BasicNameValuePair(TAG.PASSWORD, mPassword));
            params.add(new BasicNameValuePair(TAG.EMAIL, mEmail));
            params.add(new BasicNameValuePair(TAG.CONTACT, mContactNo));
            params.add(new BasicNameValuePair(TAG.ADDRESS, mAddress));
            params.add(new BasicNameValuePair(TAG.USERNAME, mUserName));
            params.add(new BasicNameValuePair(TAG.FULLNAME, mFullName));

            // getting product details by making HTTP request
            // Note that product details url will use GET request
            JSONObject json = jsonParser.makeHttpRequest(
                Config.URL_REGISTER_USER, "POST", params);

            // check your log for json response

            // json success tag
            if (json != null) {
                Log.v("json_reg", json.toString());

                reg_status = json.getString(TAG.STATUS);
                reg_message = json.getString(TAG.MESSAGE);

                return reg_status.equals(Config.STATUS_OK);
            }
        } catch (JSONException e) {
            e.printStackTrace();
        }
    }
}

```

```

    }
        // TODO: register the new account here.
        return false;
    }

    @Override
    protected void onPostExecute(final Boolean success) {
        mAuthTask = null;
        showProgress(false);

        if (success) {
            AlertDialog.Builder dlgAlert = new AlertDialog.Builder(context);

            dlgAlert.setMessage(reg_message);
            dlgAlert.setTitle("Activate your account");
            dlgAlert.setPositiveButton("Ok",
                new DialogInterface.OnClickListener() {
                    public void onClick(DialogInterface dialog, int which) {
                        finish();
                    }
                });
            dlgAlert.setCancelable(true);
            dlgAlert.create().show();
        } else {

            if (reg_status.equals(Config.STATUS_FAIL)){
                mEmailView
                    .setError(Utilities.getSpannable(reg_message));
                mEmailView.requestFocus();
            }
            else{

                Toast.makeText(context, "Connection related error. Try again.", Toast.L
            }

        }

    }

    @Override
    protected void onCancelled() {
        mAuthTask = null;
        showProgress(false);
    }
}

@Override
protected void onDestroy() {
    if (mAuthTask != null) {
        mAuthTask.cancel(true);
    }
    super.onDestroy();
}
}

```

DisasterReport.java

```

package com.patricksp.mobimapp;

import java.util.ArrayList;
import java.util.List;
import java.io.File;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.io.InputStream;

import org.apache.http.NameValuePair;
import org.apache.http.message.BasicNameValuePair;
import org.json.JSONException;
import org.json.JSONObject;

import com.google.android.gms.gcm.GoogleCloudMessaging;
import com.patricksp.dbojects.JSONParser;
import com.patricksp.dbojects.MobileReport;
import com.patricksp.utilities.ImageFilePicker;
import com.patricksp.utilities.Config;
import com.patricksp.utilities.MultiPartRequest;
import com.patricksp.utilities.ServerUtils;

import android.net.Uri;
import android.os.AsyncTask;
import android.os.Build;
import android.os.Bundle;
import android.provider.MediaStore;
import android.provider.MediaStore.MediaColumns;
import android.animation.Animator;
import android.animation.AnimatorListenerAdapter;
import android.annotation.SuppressLint;
import android.annotation.TargetApi;
import android.app.Activity;
import android.app.AlertDialog;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.content.SharedPreferences;
import android.database.Cursor;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.graphics.Point;
import android.text.TextUtils;
import android.util.Log;

```

```

import android.view.Display;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.AdapterView.OnItemSelectedListener;
import android.widget.ArrayAdapter;
import android.widget.AutoCompleteTextView;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Filter;
import android.widget.Filterable;
import android.widget.ImageView;
import android.widget.Spinner;
import android.widget.TextView;
import android.widget.Toast;

public class DisasterReport extends Activity implements OnItemClickListener{

    private static final int REQUEST_CODE = 1;
    private Bitmap bitmap;
    private ImageView imageView;
    private String photoFilePath;

    private TextView userNameView;

    private Spinner disaster_spinner;
    private AutoCompleteTextView placeAutoCompView;
    private EditText remarks_ET;
    private Button button_send_report;

    private View sendReportFormView;
    private View sendReportStatusView;
    private TextView sendReportStatusMessageView;

    private String disaster_location;
    private String disaster_type;
    private String disaster_remarks;

    private String userEmail;
    private String userPassword;
    private String userName;
    private int userID;

    private SendReportTask mAuthTask = null;
    private Context context = this;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_disaster_report);

        Bundle extras = getIntent().getExtras();

        userEmail = extras.getString(LoginActivity.TAG_EMAIL);
        userPassword = extras.getString(LoginActivity.TAG_PASSWORD);
        userName = extras.getString(LoginActivity.TAG_USERNAME);
        userID = extras.getInt(LoginActivity.TAG_USERID);

        userNameView = (TextView) findViewById(R.id.usernameView);
        userNameView.setText(userName);

        placeAutoCompView = (AutoCompleteTextView) findViewById(R.id.disaster_location);
        placeAutoCompView.setAdapter(new PlacesAutoCompleteAdapter(this, R.layout.list_places));
        placeAutoCompView.setOnItemClickListener(this);

        disaster_spinner = (Spinner) findViewById(R.id.disaster_type_spinner);
        ArrayAdapter<CharSequence> adapter = ArrayAdapter.createFromResource(this,
            R.array.array_disaster_type, android.R.layout.simple_spinner_item);
        adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
        disaster_spinner.setAdapter(adapter);

        imageView = (ImageView) findViewById(R.id.disaster_photo_imageview);
        remarks_ET = (EditText) findViewById(R.id.disaster_remarks);

        sendReportFormView = (View) findViewById(R.id.send_report_form);
        sendReportStatusView = (View) findViewById(R.id.send_report_status);
        sendReportStatusMessageView = (TextView) findViewById(R.id.send_report_status_message);

        button_send_report = (Button) findViewById(R.id.button_send_report);
        button_send_report.setOnClickListener(new View.OnClickListener() {

            @Override
            public void onClick(View arg0) {
                // TODO Auto-generated method stub
                attemptSendReport();
            }
        });
    }

    public void onPhotoClick(View View) {

        Intent intent = new Intent();
        intent.setType("image/*");
        intent.setAction(Intent.ACTION_GET_CONTENT);
        intent.addCategory(Intent.CATEGORY_OPENABLE);
        startActivityForResult(intent, REQUEST_CODE);
    }
}

```



```

    }

    @SuppressWarnings("NewApi")
    @Override
    protected void onActivityResult(int requestCode, int resultCode, Intent data) {

        InputStream stream = null;
        if (requestCode == REQUEST_CODE && resultCode == Activity.RESULT_OK)
            try {
                // recyle unused bitmaps
                if (bitmap != null) {
                    bitmap.recycle();
                }

                Uri selectedImage = data.getData();

                if (Build.VERSION.SDK_INT < Build.VERSION_CODES.KITKAT){
                    String[] filePathColumn = {MediaStore.Images.Media.DATA};
                    Cursor cursor = getContentResolver().query(selectedImage, filePathColumn, null, null, null);
                    if (cursor == null) { // Source is Dropbox or other similar local file path
                        photoFilePath = selectedImage.getPath();
                        Log.v("filepath", ""+photoFilePath);
                    } else {
                        cursor.moveToFirst();
                        int idx = cursor.getColumnIndexOrThrow(MediaStore.Images.Media.DATA);
                        photoFilePath = cursor.getString(idx);
                        cursor.close();
                    }
                }
            } else{

                photoFilePath = ImageFilePicker.getPath(this, selectedImage);
            }
        /* String[] filePathColumn = { android.provider.MediaStore.Images.ImageColumns.DATA};

        Cursor cursor = getContentResolver().query(selectedImage, filePathColumn, null, null, null);
        if (cursor == null) { // Source is Dropbox or other similar local file path
            photoFilePath = selectedImage.getPath();
            Log.v("filepath", ""+photoFilePath);
        } else {
            cursor.moveToFirst();
            int idx = cursor.getColumnIndex(filePathColumn[0]);
            photoFilePath = cursor.getString(idx);
            cursor.close();
        }
        */

        Log.v("filepath", ""+selectedImage.getPath());

        // stream = getContentResolver().openInputStream(selectedImage);
        bitmap = decodeUri(selectedImage);
        /*
        Display display = getWindowManager().getDefaultDisplay();
        Point size = new Point();
        display.getSize(size);

        int height = (size.y*bitmap.getHeight())/bitmap.getWidth();
        int width = size.x;

        bitmap=Bitmap.createScaledBitmap(bitmap,width,height,true);
        */
        imageView.setImageBitmap(bitmap);
    } catch (FileNotFoundException e) {
        e.printStackTrace();
    } finally {

        if (stream != null)
            try {
                stream.close();
            } catch (IOException e) {
                e.printStackTrace();
            }
    }
}

private Bitmap decodeUri(Uri selectedImage) throws FileNotFoundException {

    // Decode image size
    BitmapFactory.Options o = new BitmapFactory.Options();
    o.inJustDecodeBounds = true;
    BitmapFactory.decodeStream(getContentResolver().openInputStream(selectedImage), null, o);

    // The new size we want to scale to
    final int REQUIRED_SIZE = 140;

    // Find the correct scale value. It should be the power of 2.
    int width_tmp = o.outWidth, height_tmp = o.outHeight;
    int scale = 1;
    while (true) {
        if (width_tmp / 2 < REQUIRED_SIZE
            || height_tmp / 2 < REQUIRED_SIZE) {
            break;
        }
        width_tmp /= 2;
        height_tmp /= 2;
        scale *= 2;
    }

    // Decode with inSampleSize
    BitmapFactory.Options o2 = new BitmapFactory.Options();

```

```

o2.inSampleSize = scale;
return BitmapFactory.decodeStream(getContentResolver().openInputStream(selectedImage), null, o2);
}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    super.onCreateOptionsMenu(menu);
    getMenuInflater().inflate(R.menu.main, menu);

    return true;
}

public void logout(MenuItem item){

    SharedPreferences sp = getApplicationContext().getSharedPreferences(Config.TAG_SHPR_FILE,0);
    sp.edit().clear().commit();
    new AsyncTask<Void, Void, String>() {
        protected String doInBackground(Void... params) {
            try {
                GoogleCloudMessaging gcm = GoogleCloudMessaging.getInstance(getApplicationContext());
                Log.v("GCM unregister", "device unregistered");
                //ServerUtils.unregister(context, gcm.)
                gcm.unregister();

            } catch (IOException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
            }
            return "";
        }
    }.execute(null, null, null);
    finish();

}

public void changePassword(MenuItem item){

}

public void attemptSendReport(){

    if(mAuthTask!=null){
        return;
    }

    placeAutoCompView.setError(null);

    disaster_location = placeAutoCompView.getText().toString();
    disaster_type = disaster_spinner.getSelectedItem().toString();
    disaster_remarks = remarks_ET.getText().toString();

    boolean cancel = false;
    View focusView = null;

    if(TextUtils.isEmpty(disaster_location)){

        placeAutoCompView.setError(getString(R.string.error_enter_location));
        focusView = placeAutoCompView;
        cancel = true;
    }

    String[] disaster_types = getResources().getStringArray(R.array.array_disaster_type);
    if(TextUtils.equals(disaster_type, disaster_types[0])){

        Toast.makeText(this, "Please select disaster type", Toast.LENGTH_SHORT).show();
        focusView = disaster_spinner;
        cancel = true;
    }

    if(cancel){
        focusView.requestFocus();
    }
    else{
        sendReportStatusMessageView.setText(R.string.progress_sending_report);
        showProgress(true);
        mAuthTask = new SendReportTask();
        mAuthTask.execute((Void) null);
    }

}

public void onItemClick(AdapterView<?> adapterView, View view, int position, long id) {
    Log.v("wewew", "tae2");
    String str = (String) adapterView.getItemAtPosition(position);
    Toast.makeText(this, str, Toast.LENGTH_SHORT).show();
}

public class SpinnerActivity extends Activity implements OnItemSelectedListener {

    public void onItemSelected(AdapterView<?> parent, View view,
        int pos, long id) {
        // An item was selected. You can retrieve the selected item using
        // parent.getItemAtPosition(pos)
    }
}

```

```

        public void onNothingSelected(AdapterView<?> parent) {
            // Another interface callback
        }
    }

    //for place auto complete
    private class PlacesAutoCompleteAdapter extends ArrayAdapter<String> implements Filterable {
        private ArrayList<String> resultList;

        public PlacesAutoCompleteAdapter(Context context, int textViewResourceId) {
            super(context, textViewResourceId);
        }

        @Override
        public int getCount() {
            return resultList.size();
        }

        @Override
        public String getItem(int index) {
            return resultList.get(index);
        }

        @Override
        public Filter getFilter() {
            Filter filter = new Filter() {
                @Override
                protected FilterResults performFiltering(CharSequence constraint) {
                    FilterResults filterResults = new FilterResults();
                    if (constraint != null) {
                        // Retrieve the autocomplete results.
                        resultList = PlaceAutocomplete.autocomplete(constraint.toString());
                        // Log.v("places", resultList.get(0));
                        // Assign the data to the FilterResults
                        filterResults.values = resultList;
                        filterResults.count = resultList.size();
                    }
                    return filterResults;
                }

                @Override
                protected void publishResults(CharSequence constraint, FilterResults results) {
                    if (results != null && results.count > 0) {
                        notifyDataSetChanged();
                    } else {
                        notifyDataSetInvalidated();
                    }
                }
            };
            return filter;
        }
    }

    /**
     * Shows the progress UI and hides the register form.
     */
    @TargetApi(Build.VERSION_CODES.HONEYCOMB_MR2)
    private void showProgress(final boolean show) {
        // On Honeycomb MR2 we have the ViewPropertyAnimator APIs, which allow
        // for very easy animations. If available, use these APIs to fade-in
        // the progress spinner.
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.HONEYCOMB_MR2) {
            int shortAnimTime = getResources().getInteger(
                android.R.integer.config_shortAnimTime);

            sendReportStatusView.setVisibility(View.VISIBLE);
            sendReportStatusView.animate().setDuration(shortAnimTime)
                .alpha(show ? 1 : 0)
                .setListener(new AnimatorListenerAdapter() {
                    @Override
                    public void onAnimationEnd(Animator animation) {
                        sendReportStatusView.setVisibility(show ? View.VISIBLE
                            : View.GONE);
                    }
                });

            sendReportFormView.setVisibility(View.VISIBLE);
            sendReportFormView.animate().setDuration(shortAnimTime)
                .alpha(show ? 0 : 1)
                .setListener(new AnimatorListenerAdapter() {
                    @Override
                    public void onAnimationEnd(Animator animation) {
                        sendReportFormView.setVisibility(show ? View.GONE
                            : View.VISIBLE);
                    }
                });
        } else {
            // The ViewPropertyAnimator APIs are not available, so simply show
            // and hide the relevant UI components.
            sendReportStatusView.setVisibility(show ? View.VISIBLE : View.GONE);
            sendReportFormView.setVisibility(show ? View.GONE : View.VISIBLE);
        }
    }

    public class SendReportTask extends AsyncTask<Void, Void, Boolean> {

        @Override
        protected Boolean doInBackground(Void... param) {

```

```

        // TODO: attempt authentication against a network service.
        MobileReport report = new MobileReport();
        report.disaster_location = disaster_location;
        report.disaster_type = disaster_type;
        report.remarks = disaster_remarks;
        report.user_ID = userID;

        report.image = null;
        if(photoFilePath!=null)
            report.image = new File(photoFilePath);

        try{

            MultiPartRequest upload = new MultiPartRequest();
            Log.v("filepath", ""+photoFilePath);
            return upload.sendEntity(report);

        }
        catch(Exception e){
            e.printStackTrace();
            return false;
        }
    }

    @Override
    protected void onPostExecute(final Boolean success) {
        mAuthTask = null;
        showProgress(false);

        if (success) {
            AlertDialog.Builder builder = new AlertDialog.Builder(context);

            builder.setTitle("Report Disaster")
                .setMessage("Thank you for sending your report.")
                .setPositiveButton("Ok", new DialogInterface.OnClickListener() {
                    public void onClick(DialogInterface dialog, int id) {
                        finish();
                    }
                })
                .show();
        } else {
            Toast.makeText(getApplicationContext(), "Problem sending your report..", Toast.L
        }
    }

    @Override
    protected void onCancelled() {
        mAuthTask = null;
        showProgress(false);
    }
}

```

GCMIntentService.java

```

package com.patricksp.mobimapp;

import java.util.List;

import android.app.IntentService;
import android.app.Notification;
import android.app.NotificationManager;
import android.app.PendingIntent;
import android.content.Context;
import android.content.Intent;
import android.location.Location;
import android.location.LocationManager;
import android.os.Bundle;
import android.os.SystemClock;
import android.support.v4.app.NotificationCompat;
import android.text.TextUtils;
import android.util.Log;

import com.google.android.gcm.GCMBaseIntentService;
import com.google.android.gms.gcm.GoogleCloudMessaging;

import com.patricksp.mobimapp.R;
import com.patricksp.utilities.Config;
import com.patricksp.utilities.GPSTracker;
import com.patricksp.utilities.GcmUtils;
import com.patricksp.utilities.ServerUtils;

public class GCMIntentService extends IntentService {

    public static final int NOTIFICATION_ID = 1;
    private NotificationManager mNotificationManager;
    NotificationCompat.Builder builder;

    public static final String TAG = "GCM Receive";
    public static final float REQUIRED_DISTANCE = 100000; //check distance before notifying user (10km)

    public GCMIntentService() {
        super("GCMIntentService");
    }

    @Override
    protected void onHandleIntent(Intent intent) {
        Bundle extras = intent.getExtras();

```

```

    GoogleCloudMessaging gcm = GoogleCloudMessaging.getInstance(this);
    // The getMessageType() intent parameter must be the intent you received
    // in your BroadcastReceiver.
    String messageType = gcm.getMessageType(intent);
    String locx, locy;

    if (!extras.isEmpty()) { // has effect of unparceling Bundle
        /*
         * Filter messages based on message type. Since it is likely that GCM
         * will be extended in the future with new message types, just ignore
         * any message types you're not interested in, or that you don't
         * recognize.
         */
        if (GoogleCloudMessaging.
            MESSAGE_TYPE_SEND_ERROR.equals(messageType)) {
            // sendNotification("Send error: " + extras.toString(), null);
        } else if (GoogleCloudMessaging.
            MESSAGE_TYPE_DELETED.equals(messageType)) {
            // sendNotification("Deleted messages on server: " +
            // extras.toString(), null);
            // If it's a regular GCM message, do some work.
        } else if (GoogleCloudMessaging.
            MESSAGE_TYPE_MESSAGE.equals(messageType)) {
            // This loop represents the service doing some work.
            for (int i=0; i<5; i++) {
                Log.i(TAG, "Working... " + (i+1)
                    + "/5 @ " + SystemClock.elapsedRealtime());
                try {
                    Thread.sleep(2500);
                } catch (InterruptedException e) {
                }
            }
            Log.i(TAG, "Completed work @ " + SystemClock.elapsedRealtime());
            // Post notification of received message.
            locx = extras.getString("location_x");
            locy = extras.getString("location_y");
            /*inAlertProximity(Double.parseDouble(locx), Double.parseDouble(locy))*/
            if(true){
                sendNotification("Received: ".concat(extras.getString("disaster_type")).
                    concat(" at "+extras.getString("disaster_location")), extras.getString("
                Log.i(TAG, "Received: " + extras.toString());
            }
        }
    }
    // Release the wake lock provided by the WakefulBroadcastReceiver.
    GcmBroadcastReceiver.completeWakefulIntent(intent);
}

// Put the message into a notification and post it.
// This is just one simple example of what you might choose to do with
// a GCM message.
private void sendNotification(String msg, String distype, String shelters) {
    mNotificationManager = (NotificationManager)
        this.getSystemService(Context.NOTIFICATION_SERVICE);

    Intent intent = new Intent(this, MainActivity.class);
    intent.putExtra(Config.TAG_SHELTERS, shelters);

    PendingIntent contentIntent = PendingIntent.getActivity(this, 0,
        intent, 0);

    int drawable;

    if(TextUtils.equals(distype, "flood")){
        drawable = R.drawable.flood;
    }
    else if(TextUtils.equals(distype, "earthquake")){
        drawable = R.drawable.earthquake;
    }
    else if(TextUtils.equals(distype, "tsunami")){
        drawable = R.drawable.tsunami;
    }
    else if(TextUtils.equals(distype, "landslide")){
        drawable = R.drawable.landslide;
    }
    else if(TextUtils.equals(distype, "volcanic activity")){
        drawable = R.drawable.volcano;
    }
    else{
        drawable = R.drawable.fire;
    }

    NotificationCompat.Builder mBuilder =
        new NotificationCompat.Builder(this)
            .setSmallIcon(drawable)
            .setContentTitle("New MobiMAPP Alert!")
            .setStyle(new NotificationCompat.BigTextStyle()
                .bigText(msg))
            .setContentText(msg)
            .setAutoCancel(true)
            .setDefaults(Notification.DEFAULT_SOUND | Notification.DEFAULT_VIBRATE | Notification.DEFAULT_LIGHTS);

    mBuilder.setContentIntent(contentIntent);
    mNotificationManager.notify(NOTIFICATION_ID, mBuilder.build());
}

private boolean inAlertProximity(double lat, double lng){

```

```

float[] results = {0};

double[] gps = getGPS();
if (gps[0]!=0 && gps[1]!=0){

    Location.distanceBetween(lat, lng, gps[0], gps[1], results);

    Log.v(" distance", ""+results[0]);
    Log.v(" disaster loc", lat+">>"+"lng");
    if (results[0]<REQUIRED_DISTANCE){
        return true;
    }

}

return false;
}

private double[] getGPS() {
    LocationManager lm = (LocationManager) getSystemService(Context.LOCATION_SERVICE);
    List<String> providers = lm.getProviders(true);

    /* Loop over the array backwards, and if you get an accurate location, then break out the loop*/
    Location l = null;

    for (int i=providers.size()-1; i>=0; i--) {
        l = lm.getLastKnownLocation(providers.get(i));
        if (l != null) break;
    }

    double[] gps = new double[2];
    if (l != null) {
        gps[0] = l.getLatitude();
        gps[1] = l.getLongitude();
        Log.v(" Curr user location", gps[0]+">>"+"gps [1]");
    }
    return gps;
}
}
}

```

GcmBroadcastReceiver.java-----

```

package com.patricksp.mobimapp;

import android.app.Activity;
import android.content.ComponentName;
import android.content.Context;
import android.content.Intent;
import android.support.v4.content.WakefulBroadcastReceiver;

public class GcmBroadcastReceiver extends WakefulBroadcastReceiver {
    @Override
    public void onReceive(Context context, Intent intent) {
        // Explicitly specify that GcmIntentService will handle the intent.

        ComponentName comp = new ComponentName(context.getPackageName(),
            GCMIntentService.class.getName());
        // Start the service, keeping the device awake while it is launching.
        startWakefulService(context, (intent.setComponent(comp)));
        setResultCode(Activity.RESULT_OK);
    }
}

```

PlaceAutoComplete.java-----

```

package com.patricksp.mobimapp;

import java.io.IOException;
import java.io.InputStreamReader;
import java.net.HttpURLConnection;
import java.net.MalformedURLException;
import java.net.URL;
import java.net.URLEncoder;
import java.util.ArrayList;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import com.patricksp.utilities.Config;
import android.util.Log;

public class PlaceAutocomplete {

    private static final String LOG.TAG = "ExampleApp";

    private static final String PLACES_APIBASE = "https://maps.googleapis.com/maps/api/place";
    private static final String TYPE_AUTOCOMPLETE = "/autocomplete";
    private static final String OUT_JSON = "/json";

    public static ArrayList<String> autocomplete(String input) {
        ArrayList<String> resultList = null;
        Log.v("wewew", "tae");
        HttpURLConnection conn = null;
        StringBuilder jsonResults = new StringBuilder();
        try {
            StringBuilder sb = new StringBuilder(PLACES_APIBASE + TYPE_AUTOCOMPLETE + OUT_JSON);

```

```

        sb.append("?sensor=false&key=" + Config.APIKEY);
        sb.append("&components=country:ph");
        sb.append("&input=" + URLEncoder.encode(input, "utf8"));

        URL url = new URL(sb.toString());
        conn = (URLConnection) url.openConnection();
        InputStreamReader in = new InputStreamReader(conn.getInputStream());

        // Load the results into a StringBuilder
        int read;
        char[] buff = new char[1024];
        while ((read = in.read(buff)) != -1) {
            jsonResults.append(buff, 0, read);
        }
    } catch (MalformedURLException e) {
        Log.e(LOG.TAG, "Error processing Places API URL", e);
        return resultList;
    } catch (IOException e) {
        Log.e(LOG.TAG, "Error connecting to Places API", e);
        return resultList;
    } finally {
        if (conn != null) {
            conn.disconnect();
        }
    }
}

try {
    // Create a JSON object hierarchy from the results
    JSONObject jsonObj = new JSONObject(jsonResults.toString());
    JSONArray predsJsonArray = jsonObj.getJSONArray("predictions");

    // Extract the Place descriptions from the results
    resultList = new ArrayList<String>(predsJsonArray.length());
    for (int i = 0; i < predsJsonArray.length(); i++) {
        resultList.add(predsJsonArray.getJSONObject(i).getString("description"));
    }
} catch (JSONException e) {
    Log.e(LOG.TAG, "Cannot process JSON results", e);
}

return resultList;
}
}
}

```

WakeLocker.java

```

package com.patricksp.mobimapp;

import android.content.Context;
import android.os.PowerManager;

public abstract class WakeLocker {
    private static PowerManager.WakeLock wakeLock;

    public static void acquire(Context context) {
        if (wakeLock != null) wakeLock.release();

        PowerManager pm = (PowerManager) context.getSystemService(Context.POWER_SERVICE);
        wakeLock = pm.newWakeLock(PowerManager.FULL_WAKE_LOCK |
            PowerManager.ACQUIRE_CAUSES_WAKEUP |
            PowerManager.ON_AFTER_RELEASE, "WakeLocker");
        wakeLock.acquire();
    }

    public static void release() {
        if (wakeLock != null) wakeLock.release(); wakeLock = null;
    }
}

```

Config.java

```

package com.patricksp.utilities;

public class Config {

    public static final String TAG_SHPR_FILE = "com.patricksp.mobimapp.SHPR";

    public static final String TAG_SHPR_EMAIL = "sp_email";
    public static final String TAG_SHPR_USERNAME = "sp_username";
    public static final String TAG_SHPR_PASSWORD = "sp_password";
    public static final String TAG_SHPR_USERID = "sp_user_ID";

    // public static final String URL_HOST = "http://172.16.128.211/";
    // public static final String URL_HOST = "http://172.20.3.165/";
    // public static final String URL_HOST = "http://10.0.2.2/";
    // public static final String URL_HOST = "http://agila.upm.edu.ph/~npballicud/";

    public static final String URL_SERVER = URL_HOST+"mobimapp/mobile/";
    public static final String URL_SEND_REPORT = URL_HOST+"mobimapp/mobile/api/report/format/json";
    public static final String URL_LOGIN_USER = URL_HOST+"mobimapp/mobile/api/login/format/json";
    public static final String URL_REGISTER_USER = URL_HOST+"mobimapp/mobile/api/register/format/json";
    public static final String URL_DISASTERS = URL_HOST+"mobimapp/mobile/api/disasters/format/json";
    public static final String URL_FORGOT_PASSWORD = URL_HOST+"mobimapp/mobile/api/password_forgot/format/json";

    public static final String STATUS_OK = "OK";
    public static final String STATUS_FAIL = "FAIL";

    public static final String URL_GEOCODE = "https://maps.google.com/maps/api/geocode/json";
    public static final String API_KEY = "AIzaSyAlFyEPACWQpySEYIZZVHxWac6B7IG2s";
}

```

```

        public static final String TAG_SHELTERS = "tag_shelters";
    }

    -----
    ConnectionDetector.java
    package com.patricksp.utilities;

    import android.content.Context;
    import android.net.ConnectivityManager;
    import android.net.NetworkInfo;

    public class ConnectionDetector {

        private Context _context;

        public ConnectionDetector(Context context){
            this._context = context;
        }

        /**
         * Checking for all possible internet providers
         */
        public boolean isConnectingToInternet(){
            ConnectivityManager connectivity = (ConnectivityManager) _context.getSystemService(Context.CONNECTIVITY
            if (connectivity != null)
            {
                NetworkInfo[] info = connectivity.getAllNetworkInfo();
                if (info != null)
                    for (int i = 0; i < info.length; i++)
                        if (info[i].getState() == NetworkInfo.State.CONNECTED)
                        {
                            return true;
                        }
            }
            return false;
        }
    }

    -----
    GcmUtils.java
    package com.patricksp.utilities;

    import android.content.Context;
    import android.content.Intent;

    public final class GcmUtils {

        // give your server registration url here
        static final String SERVER_URL = Config.URL_HOST+"mobimapp/mobile/api/gcmreg/format/json";
        static final String SERVER_URL_UNREG = Config.URL_HOST+"mobimapp/mobile/api/gcmunreg/format/json";

        // Google project id
        public static final String SENDER_ID = "1026028966487";

        /**
         * Tag used on log messages.
         */
        public static final String TAG = "AndroidHive GCM";

        public static final String DISPLAY_MESSAGE_ACTION =
            "com.androidhive.pushnotifications.DISPLAY_MESSAGE";

        public static final String EXTRA_MESSAGE = "message";

        /**
         * Notifies UI to display a message.
         * <p>
         * This method is defined in the common helper because it's used both by
         * the UI and the background service.
         *
         * @param context application's context.
         * @param message message to be displayed.
         */
        public static void displayMessage(Context context, String message) {
            Intent intent = new Intent(DISPLAY_MESSAGE_ACTION);
            intent.putExtra(EXTRA_MESSAGE, message);
            context.sendBroadcast(intent);
        }
    }

    -----
    GPSTracker.java
    package com.patricksp.utilities;

    import android.app.AlertDialog;
    import android.app.Service;
    import android.content.Context;
    import android.content.DialogInterface;
    import android.content.Intent;
    import android.location.Location;
    import android.location.LocationListener;
    import android.location.LocationManager;
    import android.os.Bundle;
    import android.os.IBinder;
    import android.provider.Settings;
    import android.util.Log;

    public class GPSTracker extends Service implements LocationListener {

```



```

private final Context mContext;

// flag for GPS status
boolean isGPSEnabled = false;

// flag for network status
boolean isNetworkEnabled = false;

// flag for GPS status
boolean canGetLocation = false;

Location location; // location
double latitude; // latitude
double longitude; // longitude

// The minimum distance to change Updates in meters
private static final long MIN_DISTANCE_CHANGE_FOR_UPDATES = 1000; // 10 meters

// The minimum time between updates in milliseconds
private static final long MIN_TIME_BW_UPDATES = 1000 * 60 * 15; // 1 minute

// Declaring a Location Manager
protected LocationManager locationManager;

public GPSTracker(Context context) {
    this.mContext = context;
    getLocation();
}

public Location getLocation() {
    try {
        locationManager = (LocationManager) mContext
            .getSystemService(LOCATION_SERVICE);

        // getting GPS status
        isGPSEnabled = locationManager
            .isProviderEnabled(LocationManager.GPS_PROVIDER);

        // getting network status
        isNetworkEnabled = locationManager
            .isProviderEnabled(LocationManager.NETWORK_PROVIDER);

        if (!isGPSEnabled && !isNetworkEnabled) {
            // no network provider is enabled
        } else {
            this.canGetLocation = true;
            // First get location from Network Provider
            if (isNetworkEnabled) {
                locationManager.requestLocationUpdates(
                    LocationManager.NETWORK_PROVIDER,
                    MIN_TIME_BW_UPDATES,
                    MIN_DISTANCE_CHANGE_FOR_UPDATES, this);
                Log.d("Network", "Network");
                if (locationManager != null) {
                    location = locationManager
                        .getLastKnownLocation(LocationManager.NETWORK_PROVIDER);
                    if (location != null) {
                        latitude = location.getLatitude();
                        longitude = location.getLongitude();
                    }
                }
            }
            // if GPS Enabled get lat/long using GPS Services
            if (isGPSEnabled) {
                if (location == null) {
                    locationManager.requestLocationUpdates(
                        LocationManager.GPS_PROVIDER,
                        MIN_TIME_BW_UPDATES,
                        MIN_DISTANCE_CHANGE_FOR_UPDATES, this);
                    Log.d("GPS Enabled", "GPS Enabled");
                    if (locationManager != null) {
                        location = locationManager
                            .getLastKnownLocation(LocationManager.GPS_PROVIDER);
                        if (location != null) {
                            latitude = location.getLatitude();
                            longitude = location.getLongitude();
                        }
                    }
                }
            }
        }
    } catch (Exception e) {
        e.printStackTrace();
    }

    return location;
}

/**
 * Stop using GPS listener
 * Calling this function will stop using GPS in your app
 */
public void stopUsingGPS(){
    if(locationManager != null){
        locationManager.removeUpdates(GPSTracker.this);
    }
}

```

```

    }
}

/**
 * Function to get latitude
 */
public double getLatitude(){
    if(location != null){
        latitude = location.getLatitude();
    }

    // return latitude
    return latitude;
}

/**
 * Function to get longitude
 */
public double getLongitude(){
    if(location != null){
        longitude = location.getLongitude();
    }

    // return longitude
    return longitude;
}

/**
 * Function to check GPS/wifi enabled
 * @return boolean
 */
public boolean canGetLocation() {
    return this.canGetLocation;
}

/**
 * Function to show settings alert dialog
 * On pressing Settings button will launch Settings Options
 */
public void showSettingsAlert(){
    AlertDialog.Builder alertDialog = new AlertDialog.Builder(mContext);

    // Setting Dialog Title
    alertDialog.setTitle("GPS is settings");

    // Setting Dialog Message
    alertDialog.setMessage("GPS is not enabled. Do you want to go to settings menu?");

    // On pressing Settings button
    alertDialog.setPositiveButton("Settings", new DialogInterface.OnClickListener() {
        public void onClick(DialogInterface dialog, int which) {
            Intent intent = new Intent(Settings.ACTION.LOCATION.SOURCE.SETTINGS);
            mContext.startActivity(intent);
        }
    });

    // on pressing cancel button
    alertDialog.setNegativeButton("Cancel", new DialogInterface.OnClickListener() {
        public void onClick(DialogInterface dialog, int which) {
            dialog.cancel();
        }
    });

    // Showing Alert Message
    alertDialog.show();
}

@Override
public void onLocationChanged(Location location) {
}

@Override
public void onProviderDisabled(String provider) {
}

@Override
public void onProviderEnabled(String provider) {
}

@Override
public void onStatusChanged(String provider, int status, Bundle extras) {
}

@Override
public IBinder onBind(Intent arg0) {
    return null;
}
}

}

ImageFilePicker.java
-----
package com.patricksp.utilities;

import android.annotation.SuppressLint;
import android.content.ContentUri;
import android.content.Context;
import android.database.Cursor;
import android.net.Uri;
import android.os.Build;

```

```

import android.os.Environment;
import android.provider.DocumentsContract;
import android.provider.MediaStore;

public class ImageFilePicker {

    /**
     * Get a file path from a Uri. This will get the the path for Storage Access
     * Framework Documents, as well as the _data field for the MediaStore and
     * other file-based ContentProviders.
     *
     * @param context The context.
     * @param uri The Uri to query.
     * @author paulburke
     */
    @SuppressWarnings("NewApi")
    public static String getPath(final Context context, final Uri uri) {

        final boolean isKitKat = Build.VERSION.SDK_INT >= Build.VERSION_CODES.KITKAT;

        // DocumentProvider
        if (isKitKat && DocumentsContract.isDocumentUri(context, uri)) {
            // ExternalStorageProvider
            if (isExternalStorageDocument(uri)) {
                final String docId = DocumentsContract.getDocumentId(uri);
                final String[] split = docId.split(":");
                final String type = split[0];

                if ("primary".equalsIgnoreCase(type)) {
                    return Environment.getExternalStorageDirectory() + "/" + split[1];
                }

                // TODO handle non-primary volumes
            }
            // DownloadsProvider
            else if (isDownloadsDocument(uri)) {

                final String id = DocumentsContract.getDocumentId(uri);
                final Uri contentUri = ContentUris.withAppendedId(
                    Uri.parse("content://downloads/public_downloads"), Long.valueOf(id));

                return getDataColumn(context, contentUri, null, null);
            }
            // MediaProvider
            else if (isMediaDocument(uri)) {
                final String docId = DocumentsContract.getDocumentId(uri);
                final String[] split = docId.split(":");
                final String type = split[0];

                Uri contentUri = null;
                if ("image".equals(type)) {
                    contentUri = MediaStore.Images.Media.EXTERNAL_CONTENT_URI;
                } else if ("video".equals(type)) {
                    contentUri = MediaStore.Video.Media.EXTERNAL_CONTENT_URI;
                } else if ("audio".equals(type)) {
                    contentUri = MediaStore.Audio.Media.EXTERNAL_CONTENT_URI;
                }

                final String selection = "_id=?";
                final String[] selectionArgs = new String[] {
                    split[1]
                };

                return getDataColumn(context, contentUri, selection, selectionArgs);
            }
        }
        // MediaStore (and general)
        else if ("content".equalsIgnoreCase(uri.getScheme())) {
            return getDataColumn(context, uri, null, null);
        }
        // File
        else if ("file".equalsIgnoreCase(uri.getScheme())) {
            return uri.getPath();
        }

        return null;
    }

    /**
     * Get the value of the data column for this Uri. This is useful for
     * MediaStore Uris, and other file-based ContentProviders.
     *
     * @param context The context.
     * @param uri The Uri to query.
     * @param selection (Optional) Filter used in the query.
     * @param selectionArgs (Optional) Selection arguments used in the query.
     * @return The value of the _data column, which is typically a file path.
     */
    public static String getDataColumn(Context context, Uri uri, String selection,
        String[] selectionArgs) {

        Cursor cursor = null;
        final String column = "_data";
        final String[] projection = {
            column
        };

        try {
            cursor = context.getContentResolver().query(uri, projection, selection, selectionArgs,

```

```

        null);
        if (cursor != null && cursor.moveToFirst()) {
            final int column_index = cursor.getColumnIndexOrThrow(column);
            return cursor.getString(column_index);
        }
    } finally {
        if (cursor != null)
            cursor.close();
    }
    return null;
}

/**
 * @param uri The Uri to check.
 * @return Whether the Uri authority is ExternalStorageProvider.
 */
public static boolean isExternalStorageDocument(Uri uri) {
    return "com.android.externalstorage.documents".equals(uri.getAuthority());
}

/**
 * @param uri The Uri to check.
 * @return Whether the Uri authority is DownloadsProvider.
 */
public static boolean isDownloadsDocument(Uri uri) {
    return "com.android.providers.downloads.documents".equals(uri.getAuthority());
}

/**
 * @param uri The Uri to check.
 * @return Whether the Uri authority is MediaProvider.
 */
public static boolean isMediaDocument(Uri uri) {
    return "com.android.providers.media.documents".equals(uri.getAuthority());
}
}

MapUtilities.java-----
package com.patricksp.utilities;

import java.io.IOException;

import android.R;
import android.content.Context;
import android.graphics.BitmapFactory;
import android.graphics.Color;
import android.location.Geocoder;
import android.location.Location;
import android.widget.Toast;

import com.google.android.gms.maps.CameraUpdate;
import com.google.android.gms.maps.CameraUpdateFactory;
import com.google.android.gms.maps.GoogleMap;
import com.google.android.gms.maps.model.BitmapDescriptorFactory;
import com.google.android.gms.maps.model.LatLng;
import com.google.android.gms.maps.model.Marker;
import com.google.android.gms.maps.model.MarkerOptions;

public class MapUtilities {

    public static void getCurrUserLocation(Context context, GPSTracker gps, GoogleMap mMap){
        double latitude = gps.getLatitude();
        double longitude = gps.getLongitude();

        // \n is for new line
        if (gps.getLocation() == null){
            Toast.makeText(context, "Unable to obtain user location. Try again later", Toast.LENGTH.SHORT);
            return;
        }

        Toast.makeText(context, "Your Location is - \nLat: " + latitude + "\nLong: " + longitude, Toast.LENGTH.LONG);
        drawMarker(mMap, latitude, longitude, "My Location", "I'm here!", 0, true);

        CameraUpdate center =
            CameraUpdateFactory.newLatLng(new LatLng(gps.getLatitude(),
                gps.getLongitude()));

        mMap.moveCamera(center);
        CameraUpdate zoom = CameraUpdateFactory.zoomTo(11);
        mMap.animateCamera(zoom);
    }

    public static Marker drawMarker(GoogleMap mMap, double latitude, double longitude, String title,
        String snippet, int resourceId,

        Marker marker = mMap.addMarker(new MarkerOptions()
            .position(new LatLng(latitude, longitude))
            .snippet(snippet)
            .title(title)
        );

        if (resourceId == 0){
            marker.setIcon(BitmapDescriptorFactory.defaultMarker());
        }
        else{
            marker.setIcon(BitmapDescriptorFactory.fromResource(resourceId));
        }
    }
}

```

```

        if (show) {
            mMap.setInfoWindowAdapter (null);
            marker.showInfoWindow ();
        }

        return marker;
    }
}

MultiPartRequest.java
-----
package com.patricksp.utilities;

import java.io.BufferedReader;
import java.io.File;
import java.io.IOException;
import java.io.InputStream;
import java.io.Reader;

import org.apache.http.HttpEntity;
import org.apache.http.HttpResponse;
import org.apache.http.HttpVersion;
import org.apache.http.client.ClientProtocolException;
import org.apache.http.client.ResponseHandler;
import org.apache.http.client.methods.HttpPost;
import org.apache.http.entity.mime.HttpMultipartMode;
import org.apache.http.entity.mime.MultipartEntity;
import org.apache.http.entity.mime.content.FileBody;
import org.apache.http.entity.mime.content.StringBody;
import org.apache.http.impl.client.DefaultHttpClient;
import org.apache.http.params.BasicHttpParams;
import org.apache.http.params.CoreProtocolPNames;
import org.apache.http.params.HttpParams;
import org.apache.http.util.EntityUtils;
import org.json.JSONArray;
import org.json.JSONObject;
import org.json.JSONTokener;

import com.patricksp.dbojects.MobileReport;

import android.text.TextUtils;
import android.util.Log;

@SuppressWarnings("deprecation")
public class MultiPartRequest {

    private DefaultHttpClient mHttpClient;

    public MultiPartRequest () {
        HttpParams params = new BasicHttpParams ();
        params.setParameter (CoreProtocolPNames.PROTOCOL_VERSION, HttpVersion.HTTP_1_1);
        mHttpClient = new DefaultHttpClient (params);
    }

    public boolean sendEntity (MobileReport report) {

        if (report.image != null)
            Log.v ("upload file", report.image.getAbsolutePath ());

        try {
            HttpPost httpPost = new HttpPost (Config.URL_SEND_REPORT);

            MultipartEntity multipartEntity = new MultipartEntity (HttpMultipartMode.BROWSER_COMPATIBLE);
            multipartEntity.addPart ("disaster_location", new StringBody (report.disaster_location));
            multipartEntity.addPart ("disaster_type", new StringBody (report.disaster_type));
            multipartEntity.addPart ("remarks", new StringBody (report.remarks));
            multipartEntity.addPart ("user_ID", new StringBody (Integer.toString (report.user_ID)));
            if (report.image != null)
                multipartEntity.addPart ("image", new FileBody (report.image));
            httpPost.setEntity (multipartEntity);

            HttpResponse response = mHttpClient.execute (httpPost);
            Reader in = new BufferedReader (
                new InputStreamReader (response.getEntity ().getContent (), "UTF-8"));
            StringBuilder builder = new StringBuilder ();
            char [] buf = new char [1000];
            int l = 0;
            while (l >= 0) {
                builder.append (buf, 0, l);
                l = in.read (buf);
            }

            Log.d ("myapp", "response " + response.getEntity ());
            Log.d ("json", "response " + builder.toString ());
            JSONObject obj = new JSONObject (builder.toString ());

            if (TextUtils.equals (obj.getString ("status"), (Config.STATUS_FAIL)) &&
                TextUtils.equals (obj.getString ("message"), ("banned"))){

                return false;
            }

            Log.v ("upload file", "executed?");
        }
    }
}

```

```

    } catch (Exception e) {
        Log.e(MultiPartRequest.class.getName(), e.getLocalizedMessage(), e);
    }
    return true;
}

private class PhotoUploadResponseHandler implements ResponseHandler<Object> {
    @Override
    public Object handleResponse(HttpResponse response)
        throws ClientProtocolException, IOException {
        HttpEntity r_entity = response.getEntity();
        String responseString = EntityUtils.toString(r_entity);
        Log.d("UPLOAD", responseString);
        return null;
    }
}
}

ServerUtils.java
-----
package com.patricksp.utilities;

import com.patricksp.mobimapp.R;
import com.patricksp.utilities.GcmUtils;

import java.io.IOException;
import java.io.OutputStream;
import java.net.HttpURLConnection;
import java.net.MalformedURLException;
import java.net.URL;
import java.util.HashMap;
import java.util.Iterator;
import java.util.Map;
import java.util.Map.Entry;
import java.util.Random;

import android.content.Context;
import android.util.Log;

import com.google.android.gcm.GCMRegistrar;
import com.patricksp.mobimapp.R;

public final class ServerUtils {
    private static final int MAX_ATTEMPTS = 5;
    private static final int BACKOFF_MILLSECONDS = 2000;
    private static final Random random = new Random();

    /**
     * Register this account/device pair within the server.
     */
    public static void register(final Context context, String email, final String regId) {
        Log.i(GcmUtils.TAG, "registering device (regId = " + regId + ")");
        String serverUrl = GcmUtils.SERVER_URL;
        Map<String, String> params = new HashMap<String, String>();
        params.put("regId", regId);
        params.put("email", email);

        long backoff = BACKOFF_MILLSECONDS + random.nextInt(1000);
        // Once GCM returns a registration id, we need to register on our server
        // As the server might be down, we will retry it a couple
        // times.
        for (int i = 1; i <= MAX_ATTEMPTS; i++) {
            Log.d(GcmUtils.TAG, "Attempt #" + i + " to register");
            try {
                // GcmUtils.sendMessage(context, context.getString(
                // R.string.server_registering, i, MAX_ATTEMPTS));
                post(serverUrl, params);
                // GCMRegistrar.setRegisteredOnServer(context, true);
                String message = context.getString(R.string.server_registered);
                GcmUtils.sendMessage(context, message);
                return;
            } catch (IOException e) {
                // Here we are simplifying and retrying on any error; in a real
                // application, it should retry only on unrecoverable errors
                // (like HTTP error code 503).
                Log.e(GcmUtils.TAG, "Failed to register on attempt " + i + ":" + e);
                if (i == MAX_ATTEMPTS) {
                    break;
                }
            }
            try {
                Log.d(GcmUtils.TAG, "Sleeping for " + backoff + " ms before retry");
                Thread.sleep(backoff);
            } catch (InterruptedException e1) {
                // Activity finished before we complete - exit.
                Log.d(GcmUtils.TAG, "Thread interrupted: abort remaining retries!");
                Thread.currentThread().interrupt();
                return;
            }
            // increase backoff exponentially
            backoff *= 2;
        }
    }
}
String message = context.getString(R.string.server_register_error,

```

```

        MAX_ATTEMPTS);
        GcmUtils.displayMessage(context, message);
    }

    /**
     * Unregister this account/device pair within the server.
     */
    public static void unregister(final Context context, final String regId) {
        Log.i(GcmUtils.TAG, "unregistering device (regId = " + regId + ")");
        String serverUrl = GcmUtils.SERVER_URL_UNREG;
        Map<String, String> params = new HashMap<String, String>();
        params.put("regId", regId);
        try {
            post(serverUrl, params);
            // GCMRegistrar.setRegisteredOnServer(context, false);
            String message = context.getString(R.string.server_unregistered);
            //GcmUtils.displayMessage(context, message);
        } catch (IOException e) {
            // At this point the device is unregistered from GCM, but still
            // registered in the server.
            // We could try to unregister again, but it is not necessary:
            // if the server tries to send a message to the device, it will get
            // a "NotRegistered" error message and should unregister the device.
            String message = context.getString(R.string.server_unregister_error,
                e.getMessage());
            // GcmUtils.displayMessage(context, message);
        }
    }

    /**
     * Issue a POST request to the server.
     *
     * @param endpoint POST address.
     * @param params request parameters.
     *
     * @throws IOException propagated from POST.
     */
    private static void post(String endpoint, Map<String, String> params)
        throws IOException {

        URL url;
        try {
            url = new URL(endpoint);
        } catch (MalformedURLException e) {
            throw new IllegalArgumentException("invalid url: " + endpoint);
        }
        StringBuilder bodyBuilder = new StringBuilder();
        Iterator<Entry<String, String>> iterator = params.entrySet().iterator();
        // constructs the POST body using the parameters
        while (iterator.hasNext()) {
            Entry<String, String> param = iterator.next();
            bodyBuilder.append(param.getKey()).append('=')
                .append(param.getValue());
            if (iterator.hasNext()) {
                bodyBuilder.append('&');
            }
        }
        String body = bodyBuilder.toString();
        Log.v(GcmUtils.TAG, "Posting '" + body + "' to " + url);
        byte[] bytes = body.getBytes();
        HttpURLConnection conn = null;
        try {
            Log.e("URL", "> " + url);
            conn = (HttpURLConnection) url.openConnection();
            conn.setDoOutput(true);
            conn.setUseCaches(false);
            conn.setFixedLengthStreamingMode(bytes.length);
            conn.setRequestMethod("POST");
            conn.setRequestProperty("Content-Type",
                "application/x-www-form-urlencoded; charset=UTF-8");
            // post the request
            OutputStream out = conn.getOutputStream();
            out.write(bytes);
            out.close();
            // handle the response

            int status = conn.getResponseCode();
            if (status != 200) {
                throw new IOException("Post failed with error code " + status);
            }
        } finally {
            if (conn != null) {
                conn.disconnect();
            }
        }
    }
}

```

Utilities.java

```

package com.patricksp.utilities;

import com.patricksp.mobimapp.R;

import android.graphics.Color;
import android.text.SpannableString;
import android.text.style.ForegroundColorSpan;

public class Utilities {

```

```

        public static SpannableString getSpannable(String text){
            SpannableString spannable = new SpannableString(text);
            // here we set the color
            spannable.setSpan(new ForegroundColorSpan(Color.RED), 0, text.length(), 0);
            return spannable;
        }
    }
}

```

Disaster.java

```

package com.patricksp.dbobjects;

public class Disaster {

    public int disaster_ID;
    public double location_x;
    public double location_y;
    public String disaster_location;
    public String disaster_type;
    public String source;
    public String datetime_start;
    public String datetime_end;
    public String image;
    public String remarks;
    public String street_area;
    public String city_municipality;
    public String province;
    public String region;

}

```

JSONParser.java

```

package com.patricksp.dbobjects;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.UnsupportedEncodingException;
import java.util.List;

import org.apache.http.HttpEntity;
import org.apache.http.HttpResponse;
import org.apache.http.NameValuePair;
import org.apache.http.client.ClientProtocolException;
import org.apache.http.client.entity.UrlEncodedFormEntity;
import org.apache.http.client.methods.HttpGet;
import org.apache.http.client.methods.HttpPost;
import org.apache.http.client.utils.URLEncodedUtils;
import org.apache.http.impl.client.DefaultHttpClient;
import org.json.JSONException;
import org.json.JSONObject;

import android.util.Log;

public class JSONParser {

    static InputStream is = null;
    static JSONObject jObj = null;
    static String json = "";

    // constructor
    public JSONParser() {

    }

    // function get json from url
    // by making HTTP POST or GET mehtod
    public String getJSONResultString(String url, String method,
        List<NameValuePair> params) {

        // Making HTTP request
        try {

            // check for request method
            if(method == "POST"){
                // request method is POST
                // defaultHttpClient
                DefaultHttpClient httpClient = new DefaultHttpClient();
                HttpPost httpPost = new HttpPost(url);
                httpPost.setEntity(new UrlEncodedFormEntity(params));

                HttpResponse httpResponse = httpClient.execute(httpPost);
                HttpEntity httpEntity = httpResponse.getEntity();
                is = httpEntity.getContent();

            }else if(method == "GET"){
                // request method is GET
                DefaultHttpClient httpClient = new DefaultHttpClient();
                String paramString = URLEncodedUtils.format(params, "utf-8");
                url += "?" + paramString;
                HttpGet httpGet = new HttpGet(url);

                HttpResponse httpResponse = httpClient.execute(httpGet);
                HttpEntity httpEntity = httpResponse.getEntity();
                is = httpEntity.getContent();
            }

        }
    }
}

```



```

    } catch (UnsupportedEncodingException e) {
        e.printStackTrace();
    } catch (ClientProtocolException e) {
        e.printStackTrace();
    } catch (IOException e) {
        e.printStackTrace();
    }

    try {
        BufferedReader reader = new BufferedReader(new InputStreamReader(
            is, "iso-8859-1"), 8);
        StringBuilder sb = new StringBuilder();
        String line = null;
        while ((line = reader.readLine()) != null) {
            sb.append(line + "\n");
        }
        is.close();
        json = sb.toString();
    } catch (Exception e) {
        Log.e("Buffer Error", "Error converting result " + e.toString());
    }

    return json;
}

public JSONObject makeHttpRequest(String url, String method,
    List<NameValuePair> params) {

    getJSONResultString(url, method,
        params);

    // try parse the string to a JSON object
    try {
        jsonObj = new JSONObject(json);
    } catch (JSONException e) {
        Log.v("json_reg", json.toString());
        Log.e("JSON Parser", "Error parsing data " + e.toString());
    }

    // return JSON String
    return jsonObj;
}
}
}

MobileReport.java
-----
package com.patricksp.dbobjects;

import java.io.File;

public class MobileReport {

    public String disaster_location;
    public String disaster_type;
    public int user_ID;
    public String remarks;
    public File image;

    public MobileReport(){

    }

    public MobileReport(String disaster_location, String disaster_type, int user_ID,
        String remarks, File image){

        this.disaster_location = disaster_location;
        this.disaster_type = disaster_type;
        this.remarks = remarks;
        this.image = image;
        this.user_ID = user_ID;

    }

}
}
}

```

The following codes refer to the Configurations of the Web-based Mobi-MAPP

```

config.php
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');

/*
|-----|
| Base Site URL
|-----|
| URL to your CodeIgniter root. Typically this will be your base URL,
| WITH a trailing slash:
|
|         http://example.com/
|
| If this is not set then CodeIgniter will guess the protocol, domain and
| path to your installation.
|-----|
*/

```

```

$config['base_url'] = 'http://192.168.1.100/mobimapp/';

/*
|-----|
| Index File
|-----|
|
| Typically this will be your index.php file , unless you've renamed it to
| something else. If you are using mod_rewrite to remove the page set this
| variable so that it is blank.
|
*/
$config['index_page'] = '';

/*
|-----|
| URI PROTOCOL
|-----|
|
| This item determines which server global should be used to retrieve the
| URI string. The default setting of 'AUTO' works for most servers.
| If your links do not seem to work, try one of the other delicious flavors:
|
| 'AUTO'            Default - auto detects
| 'PATH_INFO'       Uses the PATH_INFO
| 'QUERY_STRING'    Uses the QUERY_STRING
| 'REQUEST_URI'     Uses the REQUEST_URI
| 'ORIG_PATH_INFO'  Uses the ORIG_PATH_INFO
|
*/
$config['uri_protocol'] = 'AUTO';

/*
|-----|
| URL suffix
|-----|
|
| This option allows you to add a suffix to all URLs generated by CodeIgniter.
| For more information please see the user guide:
|
| http://codeigniter.com/user_guide/general/urls.html
|
*/
$config['url_suffix'] = '';

/*
|-----|
| Default Language
|-----|
|
| This determines which set of language files should be used. Make sure
| there is an available translation if you intend to use something other
| than english.
|
*/
$config['language'] = 'english';

/*
|-----|
| Default Character Set
|-----|
|
| This determines which character set is used by default in various methods
| that require a character set to be provided.
|
*/
$config['charset'] = 'UTF-8';

/*
|-----|
| Enable/Disable System Hooks
|-----|
|
| If you would like to use the 'hooks' feature you must enable it by
| setting this variable to TRUE (boolean). See the user guide for details.
|
*/
$config['enable_hooks'] = FALSE;

/*
|-----|
| Class Extension Prefix
|-----|
|
| This item allows you to set the filename/classname prefix when extending
| native libraries. For more information please see the user guide:
|
| http://codeigniter.com/user_guide/general/core_classes.html
| http://codeigniter.com/user_guide/general/creating_libraries.html
|
*/
$config['subclass_prefix'] = 'MY_';

/*
|-----|
| Allowed URL Characters
|-----|

```

```

| This lets you specify with a regular expression which characters are permitted
| within your URLs. When someone tries to submit a URL with disallowed
| characters they will get a warning message.
|
| As a security measure you are STRONGLY encouraged to restrict URLs to
| as few characters as possible. By default only these are allowed: a-z 0-9%.:_-
|
| Leave blank to allow all characters -- but only if you are insane.
|
| DO NOT CHANGE THIS UNLESS YOU FULLY UNDERSTAND THE REPERCUSSIONS!!
|
*/
$config['permitted_uri_chars'] = 'a-z 0-9%.:_-';

/*
|-----|
| Enable Query Strings
|-----|
|
| By default CodeIgniter uses search-engine friendly segment based URLs:
| example.com/who/what/where/
|
| By default CodeIgniter enables access to the $_GET array. If for some
| reason you would like to disable it, set 'allow_get_array' to FALSE.
|
| You can optionally enable standard query string based URLs:
| example.com?who=me&what=something&where=here
|
| Options are: TRUE or FALSE (boolean)
|
| The other items let you set the query string 'words' that will
| invoke your controllers and its functions:
| example.com/index.php?c=controller&m=function
|
| Please note that some of the helpers won't work as expected when
| this feature is enabled, since CodeIgniter is designed primarily to
| use segment based URLs.
|
*/
$config['allow_get_array'] = TRUE;
$config['enable_query_strings'] = FALSE;
$config['controller_trigger'] = 'c';
$config['function_trigger'] = 'm';
$config['directory_trigger'] = 'd'; // experimental not currently in use

/*
|-----|
| Error Logging Threshold
|-----|
|
| If you have enabled error logging, you can set an error threshold to
| determine what gets logged. Threshold options are:
| You can enable error logging by setting a threshold over zero. The
| threshold determines what gets logged. Threshold options are:
|
| 0 = Disables logging, Error logging TURNED OFF
| 1 = Error Messages (including PHP errors)
| 2 = Debug Messages
| 3 = Informational Messages
| 4 = All Messages
|
| For a live site you'll usually only enable Errors (1) to be logged otherwise
| your log files will fill up very fast.
|
*/
$config['log_threshold'] = 0;

/*
|-----|
| Error Logging Directory Path
|-----|
|
| Leave this BLANK unless you would like to set something other than the default
| application/logs/ folder. Use a full server path with trailing slash.
|
*/
$config['log_path'] = '';

/*
|-----|
| Date Format for Logs
|-----|
|
| Each item that is logged has an associated date. You can use PHP date
| codes to set your own date formatting
|
*/
$config['log_date_format'] = 'Y-m-d H:i:s';

/*
|-----|
| Cache Directory Path
|-----|
|
| Leave this BLANK unless you would like to set something other than the default
| system/cache/ folder. Use a full server path with trailing slash.
|

```

```

*/
$config['cache-path'] = '';

/*
-----
| Encryption Key
-----

If you use the Encryption class or the Session class you
MUST set an encryption key. See the user guide for info.

*/
$config['encryption_key'] = 'patricksp';

/*
-----
| Session Variables
-----

'sess_cookie_name'           = the name you want for the cookie
'sess_expiration'           = the number of SECONDS you want the session to last.
                             by default sessions last 7200 seconds (two hours). Set to zero for no expiration.
'sess_expire_on_close'      = Whether to cause the session to expire automatically
                             when the browser window is closed
'sess_encrypt_cookie'       = Whether to encrypt the cookie
'sess_use_database'         = Whether to save the session data to a database
'sess_table_name'           = The name of the session database table
'sess_match_ip'             = Whether to match the user's IP address when reading the session data
'sess_match_useragent'      = Whether to match the User Agent when reading the session data
'sess_time_to_update'       = how many seconds between CI refreshing Session Information

*/
$config['sess_cookie_name']   = 'ci_session';
$config['sess_expiration']    = 7200;
$config['sess_expire_on_close'] = FALSE;
$config['sess_encrypt_cookie'] = FALSE;
$config['sess_use_database']  = FALSE;
$config['sess_table_name']    = 'ci_sessions';
$config['sess_match_ip']     = FALSE;
$config['sess_match_useragent'] = TRUE;
$config['sess_time_to_update'] = 300;

/*
-----
| Cookie Related Variables
-----

'cookie_prefix' = Set a prefix if you need to avoid collisions
'cookie_domain' = Set to .your-domain.com for site-wide cookies
'cookie_path'   = Typically will be a forward slash
'cookie_secure' = Cookies will only be set if a secure HTTPS connection exists.

*/
$config['cookie_prefix']     = "";
$config['cookie_domain']     = "";
$config['cookie_path']       = "/";
$config['cookie_secure']     = FALSE;

/*
-----
| Global XSS Filtering
-----

Determines whether the XSS filter is always active when GET, POST or
COOKIE data is encountered

*/
$config['global-xss-filtering'] = FALSE;

/*
-----
| Cross Site Request Forgery
-----

Enables a CSRF cookie token to be set. When set to TRUE, token will be
checked on a submitted form. If you are accepting user data, it is strongly
recommended CSRF protection be enabled.

'csrf_token_name' = The token name
'csrf_cookie_name' = The cookie name
'csrf_expire'     = The number in seconds the token should expire.

*/
$config['csrf_protection'] = FALSE;
$config['csrf_token_name'] = 'csrf_test_name';
$config['csrf_cookie_name'] = 'csrf_cookie_name';
$config['csrf_expire']     = 7200;

/*
-----
| Output Compression
-----

Enables Gzip output compression for faster page loads. When enabled,
the output class will test whether your server supports Gzip.
Even if it does, however, not all browsers support compression
so enable only if you are reasonably sure your visitors can handle it.

VERY IMPORTANT: If you are getting a blank page when compression is enabled it
means you are prematurely outputting something to your browser. It could
even be a line of whitespace at the end of one of your scripts. For

```

```

| compression to work, nothing can be sent before the output buffer is called
| by the output class. Do not 'echo' any values with compression enabled.
*/
$config['compress_output'] = FALSE;

/*
-----
| Master Time Reference
-----
|
| Options are 'local' or 'gmt'. This pref tells the system whether to use
| your server's local time as the master 'now' reference, or convert it to
| GMT. See the 'date helper' page of the user guide for information
| regarding date handling.
*/
$config['time_reference'] = 'GMT';

/*
-----
| Rewrite PHP Short Tags
-----
|
| If your PHP installation does not have short tag support enabled CI
| can rewrite the tags on-the-fly, enabling you to utilize that syntax
| in your view files. Options are TRUE or FALSE (boolean)
*/
$config['rewrite_short_tags'] = FALSE;

/*
-----
| Reverse Proxy IPs
-----
|
| If your server is behind a reverse proxy, you must whitelist the proxy IP
| addresses from which CodeIgniter should trust the HTTP_X_FORWARDED_FOR
| header in order to properly identify the visitor's IP address.
| Comma-delimited, e.g. '10.0.1.200,10.0.1.201'
*/
$config['proxy_ips'] = '';

/* End of file config.php */
/* Location: ./application/config/config.php */

constants.php
-----
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');

/*
-----
| File and Directory Modes
-----
|
| These prefs are used when checking and setting modes when working
| with the file system. The defaults are fine on servers with proper
| security, but you may wish (or even need) to change the values in
| certain environments (Apache running a separate process for each
| user, PHP under CGI with Apache suEXEC, etc.). Octal values should
| always be used to set the mode correctly.
*/
define('FILE_READ_MODE', 0644);
define('FILE_WRITE_MODE', 0666);
define('DIR_READ_MODE', 0755);
define('DIR_WRITE_MODE', 0777);

/*
-----
| File Stream Modes
-----
|
| These modes are used when working with fopen()/popen()
*/

define('FOPEN_READ',                           'rb');
define('FOPEN_READ_WRITE',                       'r+b');
define('FOPEN_WRITE_CREATE_DESTRUCTIVE',        'wb'); // truncates existing file data, use with care
define('FOPEN_READ_WRITE_CREATE_DESTRUCTIVE',  'w+b'); // truncates existing file data, use with care
define('FOPEN_WRITE_CREATE',                   'ab');
define('FOPEN_READ_WRITE_CREATE',              'a+b');
define('FOPEN_WRITE_CREATE_STRICT',            'xb');
define('FOPEN_READ_WRITE_CREATE_STRICT',       'x+b');

/*
| SESSION Constants
*/

define('SESSION_USERDATA', 'user_data');

/*
| Email Verification Constants
*/

```

```

define('EMAIL.FROM', 'npballicud@gmail.com');
define('EMAIL.FROM.WHO', 'Mobi-MAPP Administrator');
define('ACCOUNT.VERIFICATION.SUBJECT', 'Mobi-MAPP Account Verification');
define('FORGOT.PASSWORD.SUBJECT', 'Get Mobi-MAPP Password');

/*
| My Google Maps API Key
*/

define('APLKEY', 'AizaSyALfFyEPACWQpySEYIZZVHxWac6B7IG2s');

/*
| Twitter API Constants/Secrets
*/

define('TWITTER.APLKEY', 'EwnmCHddiKpGkD0aSj9H6w');
define('TWITTER.SECRET', '5nABhHx3tnbGtxJsjkPEvTXsqfJoh4GsxqLVYiPyvA');
define('TWITTER.ACCESS.TOKEN', '409047574-0gNu5kle2UfzynfiUMLtvqEN9tu8Dka2InqGPElj');
define('TWITTER.ACCESS.TOKEN.SECRET', 'zXJfBPfu4S6YcHDL8J20ojfRoc5bsam30dMbFVZDhU9');

/*
| Response Status
*/

define('STATUS.OK', 'OK');
define('STATUS.FAIL', 'FAIL');

/*
|-----
| Database Tables
|-----
| Database table constants
|-----
*/

define('NDRRMC', 'ndrrmc');
define('RDRRMC', 'rdrmmc');
define('PDRRMC', 'pdrmmc');
define('LDRRMC', 'ldrmmc');

define('MOBILE.USERS.TABLE', 'mobile_users');
define('SYSTEM.USERS.TABLE', 'system_users');
define('MOBILE.REPORTS.TABLE', 'mobilereports');
define('DISASTER.TABLE', 'disaster');
define('DISASTER.SHELTERS.TABLE', 'disaster_shelters');
define('PERSONNEL.ACTIVITY.TABLE', 'personnel_activity');

define('NDRRMC.TABLE', 'ndrrmc');
define('RDRRMC.TABLE', 'rdrmmc');
define('PDRRMC.TABLE', 'pdrmmc');
define('LDRRMC.TABLE', 'ldrmmc');

define('NDRRMC.PERSONNEL.TABLE', 'ndrrmc_personnel');
define('RDRRMC.PERSONNEL.TABLE', 'rdrmmc_personnel');
define('PDRRMC.PERSONNEL.TABLE', 'pdrmmc_personnel');
define('LDRRMC.PERSONNEL.TABLE', 'ldrmmc_personnel');

define('TOPICS.NDRRMC.RDRRMC.TABLE', 'topics_ndrrmc_rdrmmc');
define('TOPICS.RDRRMC.PDRRMC.TABLE', 'topics_rdrmmc_pdrmmc');
define('TOPICS.PDRRMC.LDRRMC.TABLE', 'topics_pdrmmc_ldrrmc');

define('MESSAGES.NDRRMC.RDRRMC.TABLE', 'messages_ndrrmc_rdrmmc');
define('MESSAGES.RDRRMC.PDRRMC.TABLE', 'messages_rdrmmc_pdrmmc');
define('MESSAGES.PDRRMC.LDRRMC.TABLE', 'messages_pdrmmc_ldrrmc');

/* End of file constants.php */
/* Location: ./application/config/constants.php */

database.php-----
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');
/*
|-----
| DATABASE CONNECTIVITY SETTINGS
|-----
| This file will contain the settings needed to access your database.
|
| For complete instructions please consult the 'Database Connection'
| page of the User Guide.
|-----
| EXPLANATION OF VARIABLES
|-----
|
| ['hostname'] The hostname of your database server.
| ['username'] The username used to connect to the database
| ['password'] The password used to connect to the database
| ['database'] The name of the database you want to connect to
| ['dbdriver'] The database type. ie: mysql. Currently supported:
|               mysql, mysqli, postgre, odbc, mssql, sqlite, oci8
|
| ['dbprefix'] You can add an optional prefix, which will be added
|               to the table name when using the Active Record class
|
| ['pconnect'] TRUE/FALSE - Whether to use a persistent connection
| ['db_debug'] TRUE/FALSE - Whether database errors should be displayed.
| ['cache_on'] TRUE/FALSE - Enables/disables query caching
| ['cachedir'] The path to the folder where cache files should be stored
| ['char_set'] The character set used in communicating with the database
| ['dbcollat'] The character collation used in communicating with the database

```

```

|                                     NOTE: For MySQL and MySQLi databases, this setting is only used
|                                     as a backup if your server is running PHP < 5.2.3 or MySQL < 5.0.7
|                                     (and in table creation queries made with DB Forge).
|                                     There is an incompatibility in PHP with mysql_real_escape_string() which
|                                     can make your site vulnerable to SQL injection if you are using a
|                                     multi-byte character set and are running versions lower than these.
|                                     Sites using Latin-1 or UTF-8 database character set and collation are unaffected.
|                                     - good for ensuring strict SQL while developing
|
|                                     ['swap_pre'] A default table prefix that should be swapped with the dbprefix
|                                     ['autoinit'] Whether or not to automatically initialize the database.
|                                     ['stricton'] TRUE/FALSE - forces 'Strict Mode' connections
|
| The $active_group variable lets you choose which connection group to
| make active. By default there is only one group (the 'default' group).
|
| The $active_record variables lets you determine whether or not to load
| the active record class
*/

$active_group = 'default';
$active_record = TRUE;

$db['default']['hostname'] = 'localhost';
$db['default']['username'] = 'root';
$db['default']['password'] = '';
$db['default']['database'] = 'mobimapp2014db';
$db['default']['dbdriver'] = 'mysql';
$db['default']['dbprefix'] = '';
$db['default']['pconnect'] = TRUE;
$db['default']['db_debug'] = TRUE;
$db['default']['cache_on'] = FALSE;
$db['default']['cachedir'] = '';
$db['default']['char_set'] = 'utf8';
$db['default']['dbcollat'] = 'utf8_general_ci';
$db['default']['swap_pre'] = '';
$db['default']['autoinit'] = TRUE;
$db['default']['stricton'] = FALSE;

/* End of file database.php */
/* Location: ./application/config/database.php */

```

The following codes refer to the Controller Classes of the Web-based Mobi-MAPP

accounts.php

```
<?php
```

```

class Accounts extends CI_Controller{

    public function __construct(){

        parent::__construct();
        $this->load->library('input');

        $this->foot_data['scripts'] = array(
            0 => base_url()."js/jquery.min.js",
            1 => base_url()."js/bootstrap.min.js",
            2 => base_url()."js/disasters.js",
            3 => base_url()."js/jquery.geocomplete.js",
            4 => base_url()."js/geocomplete.js",
            5 => base_url()."js/utills.js",
            6 => base_url()."js/bserror_handler.js"
        );
    }

    public function activate(){

        $email = $this->input->get('email', true);
        $key = $this->input->get('key', true);

        if($email && $key){
            if(strlen($key)==32){
                $this->load->model('mobile_users_model');
                $activated = $this->mobile_users_model->activateUser($email, $key);

                if($activated){
                    $data['error'] = "<div>Thanks for activating your Mobi-MAPP ac";
                    $this->load->view('activation_page', $data);
                    $this->load->view('footer', $this->foot_data);
                    return;
                }
            }
        }

        $data['error'] = "<div>Oops !Your account could not be activated. Please recheck the lin";
        $this->load->view('activation_page', $data);
        $this->load->view('footer', $this->foot_data);

    }

}

```

```
?>
```

disasters.php

<?php

```
class Disasters extends CI_Controller{

    public function __construct(){

        parent::__construct();

        $this->load->helper(array('date','form'));
        $this->load->library(array('form_validation','session'));

        if(!$this->session->userdata('user_data')){
            header('location:'.$this->base_url().'login');
            return;
        }

        $this->head_data['usermgmt_hidden'] = 'show enabled';
        $this->user_data = $this->session->userdata('user_data');
        if(!$this->user_data['user_type']=='ov_admin') && !($this->user_data['personnel_type']=='ov_admin'){
            $this->head_data['usermgmt_hidden'] = 'hidden disabled';
        }

        if($this->user_data['user_type']=='ov_admin'){
            $this->head_data['drmm_hidden'] = 'hidden disabled';
        }

        $this->load->model('mobile_reports_model');

        $this->head_data['logged_as'] = "Logged as ".$this->user_data['fname']." (".$this->user_data['active'] = 'disasters';
        $this->head_data['office_name'] = (isset($this->user_data['office_name']))? $this->user_data['office_name'] : 'disasters';
        $this->head_data['report_count'] = $this->mobile_reports_model->countReportsHere();

        $this->foot_data['scripts'] = array(
            0 => base_url()."js/jquery.min.js",
            1 => base_url()."js/bootstrap.min.js",
            2 => base_url()."js/disasters.js",
            3 => base_url()."js/jquery.geocomplete.js",
            4 => base_url()."js/geocomplete.js",
            5 => base_url()."js/utills.js",
            6 => base_url()."js/bserror_handler.js"
        );
    }

    public function index(){

        $this->ongoing();

    }

    public function map(){

        $data['curr_datetime'] = mdate("( %m/%d/%Y %h:%i:%s %a )", gmdate('Y-m-d H:i:s', gmtime()));

        $this->load->model('disaster_model');
        $data['disasters'] = $this->disaster_model->getCurrentDisasters();
        $data['user_type'] = $this->user_data['user_type'];

        array_push($this->foot_data['scripts'], "https://maps.googleapis.com/maps/api/js?region=");
        array_push($this->foot_data['scripts'], base_url()."js/gmaps.js");
        array_push($this->foot_data['scripts'], base_url()."js/mydisastermap.js");

        $this->load->view('header', $this->head_data);
        $this->load->view('disaster_map', $data);
        $this->load->view('footer', $this->foot_data);

    }

    public function ongoing(){

        $data['curr_datetime'] = mdate("( %m/%d/%Y %h:%i:%s %a )", gmdate('Y-m-d H:i:s', gmtime()));

        $this->load->model('disaster_model');
        $data['disasters'] = $this->disaster_model->getCurrentDisasters();
        $data['user_type'] = $this->user_data['user_type'];

        $data['show_done_disaster'] = 'hidden';
        $data['show_update_status'] = 'hidden';

        $this->load->view('header', $this->head_data);
        $this->load->view('disasters', $data);
        $this->load->view('footer', $this->foot_data);

    }

    public function recent(){

        $data['curr_datetime'] = mdate("( %m/%d/%Y %h:%i:%s %a )", gmdate('Y-m-d H:i:s', gmtime()));

        $this->load->model('disaster_model');
        $data['disasters'] = $this->disaster_model->gatherDisasters();
        $data['user_type'] = $this->user_data['user_type'];

        $this->load->view('header', $this->head_data);
        $this->load->view('disasters_recent', $data);
        $this->load->view('footer', $this->foot_data);

    }

    public function add(){

        $data['curr_datetime'] = mdate("( %m/%d/%Y %h:%i:%s %a )", gmdate('Y-m-d H:i:s', gmtime()));

    }

}
```



```

$this->form_validation->set_rules('datetime_start', 'Started', 'required');
$this->form_validation->set_rules('disaster_location', 'Location', 'required');

if ($this->form_validation->run() == FALSE){
    $data['show_success'] = 'hidden';
    $data['show_error'] = 'hidden';
}else{

    $query = array(
        'disaster_type' => $this->input->post('disaster_type'),
        'disaster_location' => $this->input->post('disaster_location'),
        'datetime_start' => $this->input->post('datetime_start'),
        'remarks' => $this->input->post('remarks')
    );

    $address = str_replace(" ", "+", $query['disaster_location']);

    $region = "ph";

    $json = file_get_contents("https://maps.google.com/maps/api/geocode/json?address=$address");
    $json = json_decode($json);

    //place if status == "OK"
    if(empty($json->{'results'})){
        $data['show_success'] = 'hidden';
        $data['show_error'] = 'show';
        $data['error'] = "This system is free. We might have already exceeded G";

        $this->load->view('header', $this->head_data);
        $this->load->view('disaster_add', $data);
        $this->load->view('footer', $this->foot_data);
        return;
    }
    else{

        $query['location_x'] = $json->{'results'}[0]->{'geometry'}->{'location'};
        $query['location_y'] = $json->{'results'}[0]->{'geometry'}->{'location'};

        $address_components = $json->{'results'}[0]->{'address_components'};

        $query['locality'] = "";
        $query['province'] = "";
        $query['street_area'] = "";

        foreach($address_components as $component){

            if($component->{'types'}[0] == 'route' || $component->{'types'}[0] == 'street_area'){
                $query['street_area'] = $component->{'long_name'};
            }

            if($component->{'types'}[0] == 'locality'){
                $query['locality'] = $component->{'long_name'};
            }

            if($component->{'types'}[0] == 'administrative_area_level_2'){
                $query['province'] = $component->{'long_name'};
            }

            if($component->{'types'}[0] == 'administrative_area_level_1'){
                $query['region'] = $component->{'long_name'};
            }
        }
    }
    $q_shelters = null;

    if($this->input->post('shelters')){
        $shelters = $this->input->post('shelters');
        $q_shelters = array();

        foreach($shelters as $shelter){

            $address = str_replace(" ", "+", $shelter);
            $region = "ph";
            $json = file_get_contents("https://maps.google.com/maps/api/geocode/json?address=$address");
            $json = json_decode($json);
            if(empty($json->{'results'})){
                $data['show_success'] = 'hidden';
                $data['show_error'] = 'show';
                $data['error'] = "This system is free. We might have already exceeded G";

                $this->load->view('header', $this->head_data);
                $this->load->view('disaster_add', $data);
                $this->load->view('footer', $this->foot_data);
                return;
            }
            else{

                $q_shelter = array(
                    'coord_x' => $json->{'results'}[0]->{'geometry'}->{'location'},
                    'coord_y' => $json->{'results'}[0]->{'geometry'}->{'location'},
                    'shelter_location' => $shelter,
                    'status' => 1
                );

                array_push($q_shelters, $q_shelter);
            }
        }
    }
}

```

```

        if (is_uploaded_file($_FILES['image']['tmp_name'])) {
            $config['upload_path'] = './uploads/';
            $config['allowed_types'] = 'gif|jpg|png|jpeg|jpe|bmp';

            $name = $_FILES["image"]["name"];
            $ext = end(explode(".", $name));

            $filename = time().uniqid(true).".". $ext;
            $config['file_name'] = $filename;

            $this->load->library('upload', $config);
            if (!$this->upload->do_upload('image')){

                $data['show_success'] = 'hidden';
                $data['show_error'] = 'show';
                $data['error'] = $this->upload->display_errors();
                $this->load->view('header', $this->head_data);
                $this->load->view('disasters_add', $data);
                $this->load->view('footer', $this->foot_data);
                return;
            }
            else{
                $query['image'] = base_url(). 'uploads/'. $filename;
            }
        }

        $this->load->model('disaster_model');

        if (!$this->disaster_model->addDisaster($query, $q-shelters)){

            $data['show_success'] = 'hidden';
            $data['show_error'] = 'show';
            $data['error'] = "Database related error";
        }
        else{

            $data['show_success'] = 'show';
            $data['show_error'] = 'hidden';
        }
    }

    $this->load->view('header', $this->head_data);
    $this->load->view('disasters_add', $data);
    $this->load->view('footer', $this->foot_data);
}

public function view($id){

    $this->load->model('disaster_model');

    $data['disaster'] = $this->disaster_model->getDisaster($id);
    $data['shelters'] = $this->disaster_model->getShelters($id);

    $this->load->view('disaster_view', $data);
    $this->load->view('footer');
}

public function done(){

    $this->load->library('input');

    if (!$this->input->post('submit')){
        $this->index();
        return;
    }

    $id = $this->input->post('disaster_ID');

    $this->load->model('disaster_model');
    $this->disaster_model->doneDisaster($id, now());

    $data['curr_datetime'] = mdate("( %m/%d/%Y %h:%i:%s %a )", gmto_local(now(), 'UP8'));

    $this->load->model('disaster_model');
    $data['disasters'] = $this->disaster_model->getCurrentDisasters();

    $data['show_done_disaster'] = 'show';
    $data['show_update_status'] = 'hidden';

    $this->load->view('header', $this->head_data);
    $this->load->view('disasters', $data);
    $this->load->view('footer', $this->foot_data);
}

public function edit(){

    $this->load->library('input');

    if (!$this->input->post('submit')){
        $this->index();
        return;
    }

    $data['curr_datetime'] = mdate("( %m/%d/%Y %h:%i:%s %a )", gmto_local(now(), 'UP8'));

    $data['show_success'] = 'hidden';
    $data['show_error'] = 'hidden';
}

```

```

    $data['disaster'] = $this->input->post();

    $this->session->set_userdata('edit_disaster', $data['disaster']);

    $this->load->view('header', $this->head_data);
    $this->load->view('disaster_edit', $data);
    $this->load->view('footer', $this->foot_data);
}

public function update(){

    $this->load->library('input');

    if (!$this->input->post('submit')){
        $this->index();
        return;
    }

    $data['curr_datetime'] = mdate(" %m/%d/%Y %h:%i:%s %a ", gmtime_to_local(now(), 'UP8'));

    $this->form_validation->set_rules('datetime_start', 'Started', 'required');
    $this->form_validation->set_rules('disaster_location', 'Location', 'required');

    if ($this->form_validation->run() == FALSE){
        $data['show_success'] = 'hidden';
        $data['show_error'] = 'hidden';
    }else{

        $dis_id = $this->input->post('disaster-ID');

        $query = array(
            'disaster_type' => $this->input->post('disaster_type'),
            'disaster_location' => $this->input->post('disaster_location'),
            'datetime_start' => $this->input->post('datetime_start'),
            'remarks' => $this->input->post('remarks'),
            'image' => $this->input->post('prev-image')
        );

        $address = str_replace(" ", "+", $query['disaster_location']);

        $region = "ph";

        $json = file_get_contents("https://maps.google.com/maps/api/geocode/json?address=$address&region=$region");
        $json = json_decode($json);

        //place if status == "OK"
        if (empty($json->{'results'})){
            $data['show_success'] = 'hidden';
            $data['show_error'] = 'show';
            $data['error'] = "This system is free. We might have already exceeded G";
            $data['disaster'] = $this->session->userdata('edit_disaster');

            $this->load->view('header', $this->head_data);
            $this->load->view('disaster_edit', $data);
            $this->load->view('footer', $this->foot_data);
            return;
        }
        else{

            $query['location_x'] = $json->{'results'}[0]->{'geometry'}->{'location'};
            $query['location_y'] = $json->{'results'}[0]->{'geometry'}->{'location'};

            $address_components = $json->{'results'}[0]->{'address_components'};

            $query['locality'] = "";
            $query['province'] = "";
            $query['street_area'] = "";

            foreach($address_components as $component){

                if($component->{'types'}[0] == 'route' || $component->{'types'}[0] == 'street_area'){
                    $query['street_area'] = $component->{'long_name'};
                }

                if($component->{'types'}[0] == 'locality'){
                    $query['locality'] = $component->{'long_name'};
                }

                if($component->{'types'}[0] == 'administrative_area_level_2'){
                    $query['province'] = $component->{'long_name'};
                }

                if($component->{'types'}[0] == 'administrative_area_level_1'){
                    $query['region'] = $component->{'long_name'};
                }
            }
        }

        $q_shelters = null;

        if ($this->input->post('shelters')){
            $shelters = $this->input->post('shelters');
            $q_shelters = array();

            foreach($shelters as $shelter){

                $address = str_replace(" ", "+", $shelter);
                $region = "ph";
                $json = file_get_contents("https://maps.google.com/maps/api/geocode/json?address=$address&region=$region");
                $json = json_decode($json);
            }
        }
    }
}

```

```

        if(empty($json->{'results'})){
            $data['show_success'] = 'hidden';
            $data['show_error'] = 'show';
            $data['error'] = "This system is free. We might have all
            $data['disaster'] = $this->session->userdata('edit_disas

            $this->load->view('header', $this->head_data);
            $this->load->view('disaster_edit', $data);
            $this->load->view('footer', $this->foot_data);
            return;
        }
        else{
            $q_shelter = array(
                'disaster_ID' => $dis_id,
                'coord_x' => $json->{'results'}[0]->{'geometry'},
                'coord_y' => $json->{'results'}[0]->{'geometry'},
                'shelter_location' => $shelter,
                'status' => 1
            );
            array_push($q_shelters, $q_shelter);
        }
    }
}

if(is_uploaded_file($_FILES['image']['tmp_name'])){
    $config['upload_path'] = './reportdocs/';
    $config['allowed_types'] = 'gif|jpg|png|jpeg|jpe|bmp';

    $name = $_FILES["image"]["name"];
    $ext = end(explode(".", $name));

    $filename = time().uniqid(true).".". $ext;
    $config['file_name'] = $filename;

    $this->load->library('upload', $config);
    if (!$this->upload->do_upload('image')){

        $data['show_success'] = 'hidden';
        $data['show_error'] = 'show';
        $data['disaster'] = $this->session->userdata('edit_disaster');
        $data['error'] = $this->upload->display_errors();
        $this->load->view('header', $this->head_data);
        $this->load->view('disaster_edit', $data);
        $this->load->view('footer', $this->foot_data);
        return;
    }
    else{
        $query['image'] = base_url().'reportdocs/'.$filename;
    }
}

$this->load->model('disaster_model');

if(!$this->disaster_model->updateDisaster($query, $dis_id, $q_shelters)){

    $data['show_success'] = 'hidden';
    $data['show_error'] = 'show';
    $data['error'] = "Database related error";
}
else{

    $data['show_done_disaster'] = 'hidden';
    $data['show_update_status'] = 'show';
    $this->load->model('disaster_model');
    $data['disasters'] = $this->disaster_model->getCurrentDisasters();
    $this->load->view('header', $this->head_data);
    $this->load->view('disasters', $data);
    $this->load->view('footer', $this->foot_data);
    return;
}

}

$data['disaster'] = $this->session->userdata('edit_disaster');

$this->load->view('header', $this->head_data);
$this->load->view('disaster_edit', $data);
$this->load->view('footer', $this->foot_data);
}
}

?>

```

```

drmm.php
<?php

```

```

class Drmm extends CI_Controller{
    public function __construct(){
        parent::__construct();
        $this->load->helper("date");

        $this->load->library(array("session", "input"));
        if (!$this->session->userdata("user_data")){
            header("location:".base_url()."login");
            return;
        }
    }
}

```

```

$this->head_data["usermgmt_hidden"] = "show enabled";
$this->user_data = $this->session->userdata("user_data");
if (!$this->user_data["user_type"]=="ov_admin") && !$this->user_data["personnel_type"]=""
    $this->head_data["usermgmt_hidden"] = "hidden disabled";
}
if ($this->user_data['user_type']=='ov_admin'){
    $this->head_data['drmm_hidden'] = 'hidden disabled';
}

$this->load->model("mobile_reports_model");
$this->head_data["logged_as"] = "Logged as ".$this->user_data["fname"]." (".$this->user_data["office_name"].")";
$this->head_data["active"] = "drmm";
$this->head_data["office_name"] = (isset($this->user_data["office_name"]))? $this->user_data["office_name"] : "";
$this->head_data["report_count"] = $this->mobile_reports_model->countReportsHere();

$this->foot_data["scripts"] = array(
    0 => base_url()."/js/jquery.min.js",
    1 => base_url()."/js/bootstrap.min.js",
    2 => base_url()."/js/Utils.js",
    3 => base_url()."/js/berror_handler.js",
    4 => base_url()."/js/drmm.js"
);
}

public function index(){
    $this->communication();
}

public function operations($disaster_ID=null){
    $data["curr_datetime"] = mdate(" %m/%d/%Y %h:%i:%s %a "), gmt_to_local(now(), "UP8");

    $ci =& get_instance();
    $ci->load->model('system_users_model');
    $data['personnels'] = $ci->system_users_model->getAvailablePersonnels();

    $this->load->model('drmmc_model');
    $data['operations'] = $this->drmmc_model->getDisasterOperations($disaster_ID);

    $data['show_assign'] = 'hidden';
    if (isset($this->user_data['personnel_type']) && $this->user_data['personnel_type']=='ov_admin')
        $data['show_assign'] = 'show';

    $this->load->view("header", $this->head_data);
    $this->load->view("drmm", $data);
    $this->load->view("footer", $this->foot_data);
}

public function assign(){
    $this->load->library('input');
    if (!$this->input->post('submit')){
        $this->index();
    }

    $toAssign = $this->input->post('personnel');
    $disaster_ID = $this->input->post('disaster_ID');
    $toInsert = array();

    foreach($toAssign as $personnel){
        $sarr = array(
            'user_ID' => $personnel,
            'disaster_ID' => $disaster_ID,
            'activity_type' => $this->input->post($personnel),
            'datetime_started' => mdate("%Y-%m-%d %H:%i:%s", gmt_to_local(now(), "UP8"));
        );
        array_push($toInsert, $sarr);
    }

    $this->load->model('drmmc_model');
    $this->drmmc_model->assignPersonnels($toInsert);

    $this->operations($disaster_ID);
}

public function communication(){
    $data["curr_datetime"] = mdate(" %m/%d/%Y %h:%i:%s %a "), gmt_to_local(now(), "UP8");
    // $data["topics"] = $this

    $this->load->model('communication_model');

    $data['office_name'] = $this->head_data['office_name'];
    $data['office_type'] = $this->user_data['user_type'];
    $drmmc = $this->user_data['user_type'];
    if ($drmmc!=NDRRMC)
        $data['mother_topics'] = $this->communication_model->getTopics(getMother($this->user_data['user_type'], $this->user_data['office_name']));
    if ($drmmc!=LDRRMC)
        $data['child_topics'] = $this->communication_model->getTopics($this->user_data['user_type'], $this->user_data['office_name'], $this->user_data['office_name']);

    $data["show_success"] = "hidden";
    $data["show_error"] = "hidden";
}

```

```

        $this->load->view("header", $this->head_data);
        $this->load->view("drmc_communication", $data);
        $this->load->view("footer", $this->foot_data);
    }

    public function newtopic(){

        $data["curr_datetime"] = mdate(" %m/%d/%Y %h:%i:%s %a "), gmt_to_local(now(), "UP8"));

        $data["show_success"] = "hidden";
        $data["show_error"] = "hidden";

        $this->load->library("form_validation");
        $this->load->library("input");

        $this->form_validation->set_rules("with", "With", "callback_dropdownCheck");
        $this->form_validation->set_rules("message", "Message", "required");
        $this->form_validation->set_rules("topic", "Topic", "required");

        if($this->form_validation->run() == FALSE){

            $this->load->model("drmc_model");
            if($this->user_data["user_type"]!=NDRRMC)
                $data["mother_office"] = $this->drmc_model->getOfficeName(getMother($this->user_data["user_type"]));

            if($this->user_data["user_type"]!=LDRRMC)
                $data["with_select"] = $this->drmc_model->getChildOffices($this->user_data["user_type"]);

            $this->load->view("header", $this->head_data);
            $this->load->view("drmc_communication_newtopic", $data);
            $this->load->view("footer", $this->foot_data);

        }else{

            $query = array(
                "topic_name" => $this->input->post("topic"),
                "mother_ID" => $this->user_data["drmc_ID"],
                "child_ID" => $this->input->post("with"),
                "started" => mdate("%Y-%m-%d %H:%i:%s", gmt_to_local(now(), "UP8")),
                "last_update" => mdate("%Y-%m-%d %H:%i:%s", gmt_to_local(now(), "UP8"))
            );
            $flag = $this->user_data["user_type"];

            $with = $this->input->post("with");

            if($with=="mother"){

                $flag = getMother($this->user_data["user_type"]);
                $query["mother_ID"] = $this->user_data["mother_ID"];
                $query["child_ID"] = $this->user_data["drmc_ID"];

            }

            $start_message = $this->input->post("message");

            $this->load->model("communication_model");
            if(!$this->communication_model->addTopic($flag, $query, $start_message){

                $data["show_success"] = "hidden";
                $data["show_error"] = "show";
                $data["error"] = "Database Related Error.";

                $this->load->view("header", $this->head_data);
                $this->load->view("drmc_communication_newtopic", $data);
                $this->load->view("footer", $this->foot_data);
                return;

            }

            $data["show_success"] = "show";
            $data["show_error"] = "hidden";

            $this->load->model('communication_model');

            $data['office_name'] = $this->head_data['office_name'];
            $data['office_type'] = $this->user_data['user_type'];
            $drmc = $this->user_data['user_type'];
            if($drmc!=NDRRMC)
                $data['mother_topics'] = $this->communication_model->getTopics(getMother($this->user_data['user_type']));

            if($drmc!=LDRRMC)
                $data['child_topics'] = $this->communication_model->getTopics($this->user_data['user_type'], getChild($this->user_data['user_type']));

            $this->load->view("header", $this->head_data);
            $this->load->view("drmc_communication", $data);
            $this->load->view("footer", $this->foot_data);

        }

    }

}

public function discussion($topic_ID, $mother, $error=null){

    $data["curr_datetime"] = mdate(" %m/%d/%Y %h:%i:%s %a "), gmt_to_local(now(), "UP8"));

    $data['show_error'] = 'hidden';

    if($error){

        $data['show_error'] = 'show';

    }

}

```

```

}

$this->load->model('communication_model');
$data['messages'] = $this->communication_model->getMessages($topic_ID, $mother);

$data['office_type'] = $this->user_data['user_type'];
$data['topic_ID'] = $topic_ID;
$data['mother'] = $mother;
$data['my_office'] = $this->input->get('me');
$data['other_office'] = $this->input->get('up');

$this->load->view("header", $this->head_data);
$this->load->view("drmm_communication_view", $data);
$this->load->view("footer", $this->foot_data);
}

public function sendmessage(){

$this->load->library('input');
if(!$this->input->post('submit')){
    $this->index();
    return;
}
$data["curr_datetime"] = mdate(" ( %m/%d/%Y %h:%i:%s %a )", gmt_to_local(now(), "UP8"));

$mother = $this->input->post('mother');
$topic_ID = $this->input->post('topic_ID');

$config['upload_path'] = './reportdocs/';
$config['allowed_types'] = 'doc|docx|pdf|xls|xlsx|csv|txt';

$report = '';
if(is_uploaded_file($_FILES['report_file']['tmp_name'])){

    $name = $_FILES["report_file"]["name"];
    $ext = end(explode(".", $name));

    $filename = time().uniqid(true).".". $ext;
    $config['file_name'] = $filename;

    $config['upload_path'] = './reportdocs/';

    $this->load->library('upload', $config);

    if ($this->upload->do_upload('report_file')){

        $upload_data = $this->upload->data();

        //      echo $filename; die;
        $report = $filename;
    }
    else{
        //print_r($this->upload->display_errors()); die;
        $this->discussion($topic_ID, $mother, 'File format error');
        return;
    }
}
else{
    // $this->discussion($topic_ID, $mother, 'File format error');
    //return;
    //      echo"<pre>"; print_r($_FILES['report_file']); echo "</pre>"; die;
}

$message = array(

    'topic_ID' => $this->input->post('topic_ID'),
    'message' => $this->input->post('message'),
    'date_posted' => mdate("%Y-%m-%d %H:%i:%s", gmt_to_local(now(), "UP8")),
    'posted_by' => $this->user_data['user_type'],
    'report_url' => $report
);

$data['show_error'] = 'hidden';

$this->load->model('communication_model');
if(!$this->communication_model->addMessage($mother, $message)){

    $data['show_error'] = 'show';
}

$this->load->model('communication_model');
$data['messages'] = $this->communication_model->getMessages($topic_ID, $mother);

$data['office_type'] = $this->user_data['user_type'];
$data['topic_ID'] = $topic_ID;
$data['mother'] = $mother;
$data['my_office'] = $this->input->post('my_office');
$data['other_office'] = $this->input->post('other_office');

$this->load->view("header", $this->head_data);
$this->load->view("drmm_communication_view", $data);
$this->load->view("footer", $this->foot_data);
}

public function download($url){

$this->load->helper('download');
$file = file_get_contents("./reportdocs/".$url);

```

```

        force_download($url, $file);
    }

    public function dropdownCheck($str){
        if ($str == "none"){
            $this->form_validation->set_message("dropdownCheck", "Please select one!");
            return false;
        }
        else{
            return true;
        }
    }
}

?>
login.php-----
<?php

```

```

class Login extends CI_Controller{
    public function __construct(){
        parent::__construct();
        $this->load->library(array('form_validation','session'));

        $this->foot_data['scripts'] = array(
            0 => base_url()."js/jquery.min.js",
            1 => base_url()."js/bootstrap.min.js",
            2 => base_url()."js/disasters.js",
            3 => base_url()."js/jquery.geocomplete.js",
            4 => base_url()."js/geocomplete.js",
            5 => base_url()."js/Utils.js",
            6 => base_url()."js/bseerror_handler.js"
        );
    }

    public function index(){
        $this->verify();
    }

    public function verify(){
        $this->form_validation->set_rules('email', 'Email', 'required');
        $this->form_validation->set_rules('password', 'Password', 'required');

        if($this->form_validation->run()==FALSE){
            $data['error'] = "";
            $this->load->view('signin');
            $this->load->view('footer', $this->foot_data);
        }
        else{
            $email = $this->input->post('email');
            $password = $this->input->post('password');

            $this->load->model('login_model');
            $user = $this->login_model->login($email, $password);

            if($user==FALSE){
                $data['error'] = 'Invalid username & password combination!';
                $this->load->view('signin', $data);
                $this->load->view('footer', $this->foot_data);
            }
            else{
                $this->session->set_userdata(SESSION.USERDATA, $user);
                header('Location:'.base_url()."reports");
            }
        }
    }
}

?>
logout.php-----
<?php

```

```

class Logout extends CI_Controller{
    public function __construct(){
        parent::__construct();
        $this->load->library('session');
    }

    public function index(){
        $this->session->sess_destroy();
        header('location:'.base_url().'login');
    }
}

```



```

    }
?>
reports.php
<?php
class Reports extends CI_Controller{
    public function __construct(){
        parent::__construct();
        $this->load->helper('date');
        $this->load->library('session');
        if(!$this->session->userdata('user_data')){
            header('location:'.$this->base_url().'login');
            return;
        }
        $this->head_data['usermgmt_hidden'] = 'show enabled';
        $this->user_data = $this->session->userdata('user_data');
        if(!$this->user_data['user_type']=='ov_admin') && !($this->user_data['personnel_type']=='ov_admin'){
            $this->head_data['usermgmt_hidden'] = 'hidden disabled';
        }
        if($this->user_data['user_type']=='ov_admin'){
            $this->head_data['drmm_hidden'] = 'hidden disabled';
        }
        $this->load->model('mobile_reports_model');
        $this->head_data['logged_as'] = "Logged as ".$this->user_data['fname']." (".$this->user_data['active'] = 'reports';
        $this->head_data['office_name'] = (isset($this->user_data['office_name']))? $this->user_data['office_name'] : $this->mobile_reports_model->countReportsHere();
        $this->head_data['report_count'] = $this->mobile_reports_model->countReportsHere();
        $this->foot_data['scripts'] = array(
            0 => base_url()."js/jquery.min.js",
            1 => base_url()."js/bootstrap.min.js",
            2 => base_url()."js/reports.js",
            3 => base_url()."js/utills.js",
            4 => base_url()."js/bseerror_handler.js"
        );
    }
    public function index(){
        $this->reports();
    }
    public function reports(){
        $data['curr_datetime'] = mdate(" %m/%d/%Y %h:%i:%s %a ", gmto_local(now(), 'UP8'));
        $data['show_deleted_report'] = 'hidden';
        $data['show_validated_report'] = 'hidden';
        $data['user_type'] = $this->user_data['user_type'];
        $data['disaster_reports'] = $this->mobile_reports_model->getReportsHere();
        $this->load->view('header', $this->head_data);
        $this->load->view('reports', $data);
        $this->load->view('footer', $this->foot_data);
    }
    public function delete($reputation){
        $this->load->library('input');
        if(!$this->input->post('submit')){
            $this->reports();
            return;
        }
        $id = $this->input->post('report_ID');
        $user_id = $this->input->post('user_ID');
        //echo $id.">".$user_id.">".urldecode($reputation);die;
        $this->load->model('mobile_reports_model');
        $this->mobile_reports_model->validateReport($id);
        $this->load->model('mobile_users_model');
        $this->mobile_users_model->addReputation($user_id, urldecode($reputation));
        $data['curr_datetime'] = mdate(" %m/%d/%Y %h:%i:%s %a ", gmto_local(now(), 'UP8'));
        $this->load->model('mobile_reports_model');
        $data['disaster_reports'] = $this->mobile_reports_model->getReportsHere();
        $data['show_deleted_report'] = 'show';
        $data['show_validated_report'] = 'hidden';
        $this->load->view('header', $this->head_data);
        $this->load->view('reports', $data);
        $this->load->view('footer', $this->foot_data);
    }
    public function valid($reputation){

```

```

$this->load->library('input');

if (!$this->input->post('submit')){
    $this->reports();
}

$data['curr_datetime'] = mdate(" %m/%d/%Y %h:%i:%s %a ", gmtime_to_local(now(), 'UP8'));

$id = $this->input->post('report_ID');
$user_id = $this->input->post('user_ID');

$query = array(
    'datetime_start' => $this->input->post('report_datetime'),
    'disaster_type' => $this->input->post('disaster_type'),
    'image' => $this->input->post('image'),
    'remarks' => $this->input->post('remarks'),
    'disaster_location' => $this->input->post('disaster_location'),
    'location_x' => $this->input->post('location_x'),
    'location_y' => $this->input->post('location_y')
//
//
//
//
//
//
);

//if ((!$query['location_x'] || !$query['location_y']) && !($query['locality'] || $query['
    $address = str_replace(" ", "+", $query['disaster_location']);
    $region = "ph";

    $json = file_get_contents("http://maps.google.com/maps/api/geocode/json?address=
    $json = json_decode($json);

    if (empty($json->{'results'})) {

        $data['show_deleted_report'] = 'hidden';
        $data['show_validated_report'] = 'hidden';
        $data['show_error'] = 'show';

        $this->load->view('header', $this->head_data);
        $this->load->view('reports', $data);
        $this->load->view('footer', $this->foot_data);
        return;
    }
    else {

        $query['location_x'] = $json->{'results'}[0]->{'geometry'}->{'location'
        $query['location_y'] = $json->{'results'}[0]->{'geometry'}->{'location'

        $address_components = $json->{'results'}[0]->{'address_components'};

        $query['street_area'] = "";
        $query['province'] = "";
        $query['locality'] = "";
        $query['region'] = "";

        foreach ($address_components as $component) {

            if ($component->{'types'}[0] == 'route' || $component->{'types'}
                $query['street_area'] = $component->{'long_name'};
            }

            if ($component->{'types'}[0] == 'locality') {
                $query['locality'] = $component->{'long_name'};
            }

            if ($component->{'types'}[0] == 'administrative_area_level_2') {
                $query['province'] = $component->{'long_name'};
            }

            if ($component->{'types'}[0] == 'administrative_area_level_1') {
                $query['region'] = $component->{'long_name'};
            }

            /*
            if ($component->{'types'}[0] == 'administrative_area_level_1') {
                $query['region'] = $component->{'long_name'};
            }
            */

        }

    }

}

}

$this->load->model('disaster_model');
$this->disaster_model->addDisaster($query);

$this->load->model('mobile_reports_model');
$this->mobile_reports_model->validateReport($id);

$this->mobile_users_model->addReputation($user_id, urldecode($reputation));

$data['disaster_reports'] = $this->mobile_reports_model->getReportsHere();

$data['show_deleted_report'] = 'hidden';
$data['show_validated_report'] = 'show';

$this->load->view('header', $this->head_data);
$this->load->view('reports', $data);
$this->load->view('footer', $this->foot_data);
}

```

```

        public function view($id){
            $this->load->model('mobile_reports_model');
            $report = $this->mobile_reports_model->getReport($id);
            $this->load->view('report_validation', array('report'=>$report));
            //$this->load->view('footer');
        }
    }
}
usermgmt.php
<?php
class Usermgmt extends CI_Controller{
    public function __construct(){
        parent::__construct();
        $this->load->helper('date');
        $this->load->library(array('session','form_validation','input'));
        if(!$this->session->userdata(SESSION_USERDATA)){
            header('location:'.$base_url().'login');
            return;
        }
        $this->head_data['usermgmt_hidden'] = 'hidden disabled';
        $user_data = $this->session->userdata(SESSION_USERDATA);
        if($user_data['user_type']=='ov_admin'){
            $this->head_data['usermgmt_hidden'] = 'show enabled';
            $this->admin_tag = 0; //overall admin
        }
        $this->head_data['drmc_hidden'] = 'hidden disabled';
    }
    else{
        if($user_data['personnel_type']=='admin'){
            $this->head_data['usermgmt_hidden'] = 'show enabled';
            $this->admin_tag = 1; //unit admin
        }
        else{
            header('location:'.$base_url().'reports');
            return;
        }
    }
}
        $this->load->model('mobile_reports_model');
        $this->head_data['logged_as'] = "Logged as ".$user_data['fname']." (".$user_data['user_type'].")";
        $this->head_data['active'] = 'usermgmt';
        $this->head_data['office_name'] = (isset($user_data['office_name']))? $user_data['office_name'] : ' ';
        $this->head_data['report_count'] = $this->mobile_reports_model->countReportsHere();
        $this->foot_data['scripts'] = array(
            0 => base_url()."js/jquery.min.js",
            1 => base_url()."js/bootstrap.min.js",
            2 => base_url()."js/usermgmt.js",
            3 => base_url()."js/utills.js",
            4 => base_url()."js/bseerror_handler.js",
            5 => base_url()."js/jquery.geocomplete.js"
        );
    }
    public function index(){
        if($this->admin_tag==0)
            $this->manageUnits();
        else
            $this->manageUsers();
    }
    private function manageUsers(){
        //$this->index();
        $data['curr_datetime'] = mdate(" %m/%d/%Y %h:%i:%s %a ", gmt_to_local(now(), 'UP8'));
        $this->load->model('system_users_model');
        $data['users'] = $this->system_users_model->getUsersHere();
        $data['show_success'] = 'hidden';
        $this->load->view('header', $this->head_data);
        $this->load->view('usermgmt_users', $data);
        $this->load->view('footer', $this->foot_data);
    }
    private function manageUnits(){
        $data['curr_datetime'] = mdate(" %m/%d/%Y %h:%i:%s %a ", gmt_to_local(now(), 'UP8'));
        $this->load->model('drmc_model');
        $data['drmc'] = $this->drmc_model->getRegisteredOffices();
        $data['show_success'] = 'hidden';
        $this->load->view('header', $this->head_data);
    }
}

```

```

        $this->load->view('usermgmt_units', $data);
        $this->load->view('footer', $this->foot_data);
    }

    //register unit
    public function register(){

        if($this->admin_tag!=0){
            $this->index();
            return;
        }

        $data['curr_datetime'] = mdate(" %m/%d/%Y %h:%i:%s %a )", gmt_to_local(now(), 'UP8'));

        $this->form_validation->set_rules('office', 'Council', 'callback_dropdownCheck');

        $this->load->model('drmmc_model');
        $data['regions'] = $this->drmmc_model->getRegions();

        if($this->form_validation->run()==FALSE){
            $data['show_error'] = 'hidden';
            $data['show_success'] = 'hidden';
        }
        else{

            $drmmc_type = $this->input->post('office');

            $query = array(
                'office_name' => $this->input->post('office_name'),
                'email' => $this->input->post('email'),
                'contact_no' => $this->input->post('contact_no'),
                'address' => $this->input->post('address'),
                'region' => $this->input->post('region')
            );

            $admin = array(
                'fname' => $this->input->post('admin_fname'),
                'lname' => $this->input->post('admin_lname'),
                'contact_no' => $this->input->post('admin_contact'),
                'address' => $this->input->post('admin_address'),
                'email' => $this->input->post('admin_email'),
                'password' => md5($this->input->post('admin_password')),
                'active' => 'yes'
            );

            $this->load->model('drmmc_model');

            $province = $this->input->post('province');
            if($province){
                //echo "hello";die;
                if(!$this->drmmc_model->isInRegion($province, $query['region'])){

                    $data['show_success'] = 'hidden';
                    $data['show_error'] = 'show';
                    $data['error'] = 'Please specify correct province for selected';
                    $this->load->view('header', $this->head_data);
                    $this->load->view('usermgmt_add_unit', $data);
                    $this->load->view('footer', $this->foot_data);
                    return;
                }
                $query['province'] = $province;
            }

            $locality = $this->input->post('locality');
            if($locality){
                $query['locality'] = $locality;
            }

            $res = $this->drmmc_model->registerUnit($query, $admin, $drmmc_type);
            if($res!=TRUE){
                //echo "hello";die;
                $data['show_success'] = 'hidden';
                $data['show_error'] = 'show';
                $data['error'] = $res;
                $this->load->view('header', $this->head_data);
                $this->load->view('usermgmt_add_unit', $data);
                $this->load->view('footer', $this->foot_data);
                return;
            }

            $data['show_error'] = 'hidden';
            $data['show_success'] = 'show';
        }

        $this->load->view('header', $this->head_data);
        $this->load->view('usermgmt_add_unit', $data);
        $this->load->view('footer', $this->foot_data);
    }

    public function dropdownCheck($str){

        if ($str == 'none'){
            $this->form_validation->set_message('dropdownCheck', 'Please select one!');
            return FALSE;
        }
        else{
            return TRUE;
        }
    }
}

```

```

//register users
public function userreg(){
    if($this->admin_tag!=1){
        $this->index();
        return;
    }

    $data['show_error'] = 'hidden';
    $data['curr_datetime'] = mdate(" %m/%d/%Y %h:%i:%s %a )", gmt_to_local(now(), 'UP8'));

    $this->form_validation->set_rules('personnel_type', 'Personnel Type', 'callback_dropdown');
    if($this->form_validation->run()==FALSE){
        $data['show_success'] = 'hidden';
        $data['show_error'] = 'hidden';
        $this->load->view('header', $this->head_data);
        $this->load->view('usermgmt-add-user', $data);
        $this->load->view('footer', $this->foot_data);
        return;
    }
    else{
        $new_user = array(
            'fname' => $this->input->post('fname'),
            'lname' => $this->input->post('lname'),
            'contact_no' => $this->input->post('contact_no'),
            'address' => $this->input->post('address'),
            'email' => $this->input->post('email'),
            'password' => md5($this->input->post('password')),
            'active' => 'yes'
        );

        $personnel_type = $this->input->post('personnel_type');

        $user_data = $this->session->userdata(SESSION_USERDATA);
        $drmc = $user_data['user_type'];
        $drmc_ID = $user_data['drmc_ID'];

        $this->load->model('drmc_model');
        $res = $this->drmc_model->registerUser($new_user, $drmc, $drmc_ID, $personnel_type);

        $data['show_success'] = 'show';
        if($res !== TRUE){
            $data['show_success'] = 'hidden';
            $data['show_error'] = 'show';
            $data['error'] = ($res == FALSE) ? 'Database error. Please try again.' : '';
        }

        $this->load->view('header', $this->head_data);
        $this->load->view('usermgmt-add-user', $data);
        $this->load->view('footer', $this->foot_data);
    }
}

public function provinceCheck($province){
    $this->load->model('drmc_model');
    $reg = $this->input->post('region');

    $provinces = $this->drmc_model->getProvinces($reg);

    foreach($provinces as $p){
        if($p['province']==$province){
            return TRUE;
        }
    }

    $this->form_validation->set_message('provinceCheck', 'Invalid province in that region!');
    return FALSE;
}

public function mobile(){
    $data['curr_datetime'] = mdate(" %m/%d/%Y %h:%i:%s %a )", gmt_to_local(now(), 'UP8'));

    $this->load->model('mobile_users_model');
    $data['mob_users'] = $this->mobile_users_model->getMobileUsers();
    $data['show_success'] = 'hidden';
    $data['show_error'] = 'hidden';

    $this->load->view('header', $this->head_data);
    $this->load->view('usermgmt-mobile', $data);
    $this->load->view('footer', $this->foot_data);
}

public function mobileview($id){
    $this->load->model('mobile_users_model');
    $mobile_user = $this->mobile_users_model->getMobileUsers($id);
    $this->load->view('users-mobile-view', array('mobile_user'=>$mobile_user[0]));
}

public function banmobile(){
    if(!$this->input->post('submit')){
        $this->index();
    }
}

```

```

        return;
    }
    $data['curr_datetime'] = mdate(" %m/%d/%Y %h:%i:%s %a )", gmt_to_local(now(), 'UP8'));
    $user_ID = $this->input->post('user_ID');

    $this->load->model('mobile_users_model');
    if($this->mobile_users_model->banID($user_ID)){
        $data['success_msg'] = 'Mobile User has been banned.';
        $data['show_success'] = 'show';
        $data['show_error'] = 'hidden';
    }
    else{
        $data['show_success'] = 'hidden';
        $data['success_msg'] = '';
        $data['show_error'] = 'show';
    }

    $this->load->model('mobile_users_model');
    $data['mob_users'] = $this->mobile_users_model->getMobileUsers();

    $this->load->view('header', $this->head_data);
    $this->load->view('usermgmt_mobile', $data);
    $this->load->view('footer', $this->foot_data);
}

public function unbanmobile(){
    if(!$this->input->post('submit')){
        $this->index();
        return;
    }
    $data['curr_datetime'] = mdate(" %m/%d/%Y %h:%i:%s %a )", gmt_to_local(now(), 'UP8'));
    $user_ID = $this->input->post('user_ID');

    $this->load->model('mobile_users_model');
    if($this->mobile_users_model->unbanID($user_ID)){
        $data['success_msg'] = 'Mobile User has been unbanned.';
        $data['show_success'] = 'show';
        $data['show_error'] = 'hidden';
    }
    else{
        $data['show_success'] = 'hidden';
        $data['success_msg'] = '';
        $data['show_error'] = 'show';
    }

    $this->load->model('mobile_users_model');
    $data['mob_users'] = $this->mobile_users_model->getMobileUsers();

    $this->load->view('header', $this->head_data);
    $this->load->view('usermgmt_mobile', $data);
    $this->load->view('footer', $this->foot_data);
}
}

?>

```

The following codes refer to the Model Classes of the Web-based Mobi-MAPP

communication_model.php

```
<?php
```

```

class Communication_model extends CI_Model{
    public function __construct(){
        parent::__construct();
    }

    public function addTopic($drmc,$query,$message){
        if($drmc==NDRMC){
            $table = '_ndrrmc_rdrmc';
        }
        else if($drmc==RDRMC){
            $table = '_rdrmc_pdrmc';
        }
        else{
            $table = '_pdrmc_ldrrmc';
        }

        if($this->db->insert('topics '.$table,$query)){
            $m = array(
                'topic_ID' => $this->db->insert_id(),
                'message' => $message
            );
            if($this->db->insert('messages '.$table,$m)){
                return TRUE;
            }
        }

        return false;
    }
}

```

```

public function getTopics($mother, $mother_ID, $child, $child_ID){
    $this->db->select('t.*.m.office_name as mother_office, c.office_name as child_office ');
    $this->db->from('topics_'. $mother.'_'. $child.'_t');
    $this->db->join($mother.'_m', 't.mother_ID = m.'. $mother.'_ID ');
    $this->db->join($child.'_c', 't.child_ID = c.'. $child.'_ID ');
    $this->db->where('t.mother_ID', $mother_ID);
    if($child_ID!=null)
        $this->db->where('t.child_ID', $child_ID);

    $res = $this->db->get();

    return $res->result_array();
}

public function getMessages($topic_ID, $mother){
    $this->db->where('topic_ID', $topic_ID);
    $this->db->order_by('date_posted', 'asc');
    $res = $this->db->get('messages_'. $mother.'_'. getChild($mother));

    return $res->result_array();
}

public function addMessage($mother, $message){
    if($this->db->insert('messages_'. $mother.'_'. getChild($mother), $message)){
        $this->db->set('last_update', $message['date_posted']);
        $this->db->where('topic_ID', $message['topic_ID']);
        return $this->db->update('topics_'. $mother.'_'. getChild($mother));
    }
    return false;
}
}

?>
disaster_model.php-----
<?php

```

```

class Disaster_model extends CI_Model{
    public function __construct(){
        parent::__construct();
        $this->load->library('session');

        $this->region = "";
        $this->province = "";
        $this->locality = "";

        $user = $this->session->userdata(SESSION.USERDATA);

        if($user){
            if($user['user_type']==RDRRC){
                $this->region = $user['region'];
            }
            else if($user['user_type']==PDRRC){
                $this->region = $user['region'];
                if($this->region=='National Capital Region'){
                    $this->region = "Metro Manila";
                    $this->province = "";
                }
                else{
                    $this->province = $user['province'];
                }
            }
            else if($user['user_type']==LDRRC){
                $this->region = $user['region'];
                if($this->region=='National Capital Region'){
                    $this->region = "Metro Manila";
                    $this->province = "";
                }
                else{
                    $this->province = $user['province'];
                }
            }
            $this->locality = $user['locality'];
        }

        //print_r($reg_ids);
    }

    public function getCurrentDisasters(){
        if($this->region){
            $this->db->where('region', $this->region);
        }
        if($this->province){
            $this->db->where('region', $this->region);
            $this->db->where('province', $this->province);
        }
        if($this->locality){
            $this->db->where('region', $this->region);
            $this->db->where('province', $this->province);
        }
    }
}

```

```

        $this->db->where('locality', $this->locality);
    }

    $this->db->where('datetime_end', null);
    $res = $this->db->get(DISASTER_TABLE);
    // print_r($res->result_array()); die;
    return $res->result_array();
}

public function gatherDisasters($filter=null){
    if($this->region){
        $this->db->where('region', $this->region);
    }
    if($this->province){
        $this->db->where('region', $this->region);
        $this->db->where('province', $this->province);
    }
    if($this->locality){
        $this->db->where('region', $this->region);
        $this->db->where('province', $this->province);
        $this->db->where('locality', $this->locality);
    }

    $this->db->where('datetime_end!=",",', false);
    $res = $this->db->get(DISASTER_TABLE);

    return $res->result_array();
}

public function addDisaster($query, $shelters=null){
    $res = $this->db->insert(DISASTER_TABLE, $query);
    $id = $this->db->insert_id();

    if($shelters!=null){
        foreach($shelters as &$shelter){
            $shelter['disaster_ID'] = $id;
        }
        $res = $this->db->insert_batch(DISASTER_SHELTERS_TABLE, $shelters);
    }

    $this->load->library('GCM');
    $this->gcm = new GCM();
    $this->load->model('mobile_users_model');
    $registration_ids = $this->mobile_users_model->getGCMs();

    $this->reg_ids = array();
    foreach($registration_ids as $r_ids){
        $this->reg_ids[] = $r_ids['gcm.id'];
    }

    $shelter_coords = "";
    if($shelters!=null)
    foreach($shelters as $s){
        foreach($shelter as $sh){
            $shelter_coords.=$sh['coord_x'].", ".$sh['coord_y'];
        }
        $shelter_coords.='|';
    }
    $shelter_coords = trim($shelter_coords, "|");

    $message = array(
        "disaster_type"=>$query['disaster_type'],
        "location_x"=>$query['location_x'],
        "location_y"=>$query['location_y'],
        "disaster_location"=>$query['disaster_location'],
        "shelters"=>$shelter_coords);
    $this->gcm->send_notification($this->reg_ids, $message);

    return $res;
}

public function getDisaster($id){
    $res = $this->db->get_where(DISASTER_TABLE, array('disaster_ID' => $id));
    $res_arr = $res->result_array();

    return $res_arr[0];
}

public function getShelters($disaster_ID){
    $res = $this->db->get_where(DISASTER_SHELTERS_TABLE, array('disaster_ID' => $disaster_ID));

    return $res->result_array();
}

public function doneDisaster($id, $when){
    $this->db->where('disaster_ID', $id);
    $data = array(
        'datetime_end' => $when
    );

    return $this->db->update(DISASTER_TABLE, $data);
}

public function updateDisaster($data, $id, $shelters=null){

```



```

        if ($shelters!=null){
            $this->db->delete(DISASTER_SHELTERS_TABLE, array('disaster_ID' => $id));
            $res = $this->db->insert_batch(DISASTER_SHELTERS_TABLE, $shelters);
        }

        $this->db->where('disaster_ID', $id);
        return $this->db->update(DISASTER_TABLE, $data);
    }
}

?>
drrmc_model.php
<?php

class Drrmc_model extends CI_Controller{

    public function __construct(){

        parent::__construct();
        $this->load->library('session');

        $this->region = "";
        $this->province = "";
        $this->locality = "";

        $user = $this->session->userdata(SESSION_USERDATA);

        if ($user['user_type']!= 'ov_admin'){
            $this->drrmc_ID = $user['drrmc_ID'];
            $this->unit = $user['user_type'];
        }

        if ($user){
            if ($user['user_type']==RDRRMC){
                $this->region = $user['region'];
            }
            else if ($user['user_type']==PDRRMC){
                $this->region = $user['region'];
                $this->province = $user['province'];
            }
            else if ($user['user_type']==LDRRMC){
                $this->region = $user['region'];
                $this->province = $user['province'];
                $this->locality = $user['locality'];
            }
        }
    }

    public function getRegisteredOffices(){

        $data = array();

        $this->db->select('n.office_name,n.contact_no,n.email,
            concat(s.fname,\' \' ,s.lname) as admin_name, s.contact_no as admin_cont
        $this->db->from(NDRRMC_TABLE.' n');
        $this->db->join(NDRRMC_PERSONNEL_TABLE.' n1', 'n.ndrrmc_ID=n1.ndrrmc_ID AND n1.personne
        $this->db->join(SYSTEM_USERS_TABLE.' s', 'n1.user_ID=s.user_ID');

        $res = $this->db->get();

        array_push($data, $res->result_array());

        $this->db->select('n.office_name,n.contact_no,n.email,
            concat(s.fname,\' \' ,s.lname) as admin_name, s.contact_no as admin_contact', fa
        $this->db->from(RDRRMC_TABLE.' n');
        $this->db->join(RDRRMC_PERSONNEL_TABLE.' n1', 'n.rdrmc_ID=n1.rdrmc_ID AND n.registered
        $this->db->join(SYSTEM_USERS_TABLE.' s', 'n1.user_ID=s.user_ID');

        $res = $this->db->get();
        array_push($data, $res->result_array());

        //echo"<pre>";print_r($data);echo"</pre>";die;

        $this->db->select('n.office_name,n.contact_no,n.email,
            concat(s.fname,\' \' ,s.lname) as admin_name, s.contact_no as admin_contact', fa
        $this->db->from(PDRRMC_TABLE.' n');
        $this->db->join(PDRRMC_PERSONNEL_TABLE.' n1', 'n.pdrmc_ID=n1.pdrmc_ID AND n.registered
        $this->db->join(SYSTEM_USERS_TABLE.' s', 'n1.user_ID=s.user_ID');

        $res = $this->db->get();
        array_push($data, $res->result_array());

        $this->db->select('n.office_name,n.contact_no,n.email,
            concat(s.fname,\' \' ,s.lname) as admin_name, s.contact_no as admin_contact', fa
        $this->db->from(LDRRMC_TABLE.' n');
        $this->db->join(LDRRMC_PERSONNEL_TABLE.' n1', 'n.ldrmc_ID=n1.ldrmc_ID
        AND n.registered=1', 'left');
        $this->db->join(SYSTEM_USERS_TABLE.' s', 'n1.user_ID=s.user_ID');

        $res = $this->db->get();
        array_push($data, $res->result_array());

        /*
        echo "<pre>";
        print_r($data);
    }
}

```

```

        echo "</pre>";die;*/
        return $data;
    }

    public function getProvinces($region=null){
        $this->db->select('p.province');
        $this->db->from(PDRRMC.TABLE.' p');

        if($region!=null)
            $this->db->join(RDRRMC.TABLE.' r', 'p.rdrmc_ID=r.rdrmc_ID AND r.short_name="');

        $res = $this->db->get();
        return $res->result_array();
    }

    public function isInRegion($province, $region){
        $this->db->select('province');
        $this->db->where('LOWER(province)', strtolower($province));
        $this->db->where('region', $region);

        $res = $this->db->get(PDRRMC.TABLE);

        return ($res->num_rows()>0);
    }

    public function isInProvince($locality, $province){
        $this->db->select('locality');
        $this->db->where('locality', $locality);
        $this->db->where('province', $province);

        $res = $this->db->get(PDRRMC.TABLE);

        return ($res->num_rows()>0);
    }

    public function getRegions(){
        $this->db->select('region,short_name');
        $res = $this->db->get(RDRRMC.TABLE);

        return $res->result_array();
    }

    public function registerUnit($office, $admin, $drrmc_type){
        $this->db->where('LOWER(region)', strtolower($office['region']));
        if($drrmc_type==PDRRMC || $drrmc_type==LDRRMC)
            $this->db->where('LOWER(province)', strtolower($office['province']));

        if($drrmc_type==LDRRMC){
            $this->db->select('pdrmc_ID');
            $this->db->where('province', $office['province']);
            $prov = $this->db->get(PDRRMC.TABLE);
            $prov_arr = $prov->result_array();

            $office['pdrmc_ID'] = $prov_arr[0]['pdrmc_ID'];
            $office['registered'] = 1;
            $this->db->insert($drrmc_type, $office);
            $sid = $this->db->insert_id();
        }
        else{// $this->db->where('LOWER(locality)', strtolower($office['locality']));

            $this->db->select($drrmc_type.'_ID');
            $res = $this->db->get($drrmc_type);
            if(!$res){
                return false;
            }
            $res_arr = $res->result_array();

            $sid = $res_arr[0][$drrmc_type.'_ID'];

            $office['registered'] = 1;
            $this->db->where($drrmc_type.'_ID', $sid);

            if(!$this->db->update($drrmc_type, $office)){
                return false;
            }
        }
        $res = $this->db->get_where(SYSTEM.USERS.TABLE, array('email'=>$admin['email']));

        if($res->num_rows()>0){
            return "email already registered!";
        }

        $admin['user_type'] = $drrmc_type;

        if(!$this->db->insert(SYSTEM.USERS.TABLE, $admin)){
            echo $drrmc_type;die;
            return false;
        }

        $personnel = array(
            $drrmc_type.'_ID' => $sid,
            'user_ID' => $this->db->insert_ID(),

```

```

        'personnel_type' => 'admin'
    );
    if (!$this->db->insert($drmc_type.'_personnel', $personnel)){
        return false;
    }
    return true;
}

public function registerUser($new_user, $drmc, $drmc_ID, $personnel_type){
    $res = $this->db->get_where(SYSTEM_USERS_TABLE, array('email'=>$new_user['email']));
    if ($res->num_rows() > 0){
        return "email already registered!";
    }
    if (!$this->db->insert(SYSTEM_USERS_TABLE, $new_user)){
        return false;
    }
    $new_personnel = array(
        'user_ID' => $this->db->insert_id(),
        $drmc.'_ID' => $drmc_ID,
        'personnel_type' => $personnel_type
    );
    if (!$this->db->insert($drmc.'_personnel', $new_personnel)){
        return false;
    }
    return true;
}

public function getChildOffices($mother, $mother_ID){
    $this->db->select(array(getChild($mother)['_ID as drmc_ID', 'office_name']));
    $this->db->from(getChild($mother));
    $this->db->where($mother.'_ID', $mother_ID);
    $this->db->where('office_name !=', '');
    $this->db->where('registered', '1');

    $res = $this->db->get();

    return $res->result_array();
}

public function getOfficeName($office, $office_ID){
    $this->db->select('office_name');
    $this->db->where($office.'_ID', $office_ID);
    $res = $this->db->get($office);
    $res_arr = $res->result_array();

    return $res_arr[0]['office_name'];
}

public function getDisasterOperations($disaster_ID=null){
    $this->db->select('d.*, p.activity_type, p.datetime_started, p.datetime_ended, s.fname,
    $this->db->from(DISASTER_TABLE.' d');
    $this->db->join(PERSONNELACTIVITY_TABLE.' p', 'd.disaster_ID=p.disaster_ID', 'left');
    $this->db->join(SYSTEM_USERS_TABLE.' s', 'p.user_ID=s.user_ID', 'left');

    if ($disaster_ID!=null)
        $this->db->where('d.disaster_ID', $disaster_ID);

    $this->db->where('d.datetime_end', null);
    $this->db->where('p.datetime_ended', null);

    $res = $this->db->get();

    return $res->result_array();
}

public function assignPersonnels($personnels){
    $this->db->insert_batch(PERSONNELACTIVITY_TABLE, $personnels);

    $this->db->set('available', 'no');

    foreach($personnels as $p){
        $this->db->where('user_ID', $p['user_ID']);
    }

    $this->db->update(SYSTEM_USERS_TABLE);
}

}

?>
login_model.php
<?php

class Login_model extends CI_Model{

```

```

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

public function login($email, $password){
    $res = $this->db->get_where(SYSTEM.USERS_TABLE, array('email'=>$email, 'password'=>md5(
    if($res->num_rows() > 0){
        $user = $res->result_array();
        if($user[0]['user_type']!= 'ov-admin'){
            $unit = $user[0]['user_type'];
            $sid = $user[0]['user_ID'];

            $select = 's.user_ID, s.fname, s.lname, s.email, s.user_type,p.*,
                q.personnel_type, p.office_name, p.'. $unit.'_ID as drmc_ID';

            if($unit!=NDRRMC)
                $select.=' , p.'.getMother($unit)['_ID as mother_ID';

            if($unit==RDRRMC){
                $select.=' ,p.region';
            }
            else if($unit==PDRRMC){
                $select.=' ,p.region ,p.province';
            }
            else if($unit==LDRRMC){
                $select.=' ,p.region ,p.province ,p.locality';
            }

            $this->db->select($select);
            $this->db->from(SYSTEM.USERS_TABLE.' s');
            $this->db->join($unit.'_personnel q', 's.user_ID=q.user_ID AND s.user_ID
            $this->db->join($unit.'_p', 'p.'. $unit.'_ID=q.'. $unit.'_ID', 'inner');

            $res = $this->db->get();
            $user = $res->result_array();
        }

        return $user[0];
    }

    return false;
}

}

?>
mobile_reports_model.php
<?php

```

```

class Mobile_reports_model extends CI_Model{

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

        $this->load->library('session');

        $this->region = "";
        $this->province = "";
        $this->locality = "";

        $user = $this->session->userdata(SESSION.USERDATA);

        if($user){
            if($user['user_type']==RDRRMC){
                $this->region = $user['region'];
            }
            else if($user['user_type']==PDRRMC){
                $this->region = $user['region'];
                if($this->region=='National Capital Region'){
                    $this->region = "Metro Manila";
                    $this->province = "";
                }
                else{
                    $this->province = $user['province'];
                }
            }
            else if($user['user_type']==LDRRMC){
                $this->region = $user['region'];
                if($this->region=='National Capital Region'){
                    $this->region = "Metro Manila";
                    $this->province = "";
                }
                else{
                    $this->province = $user['province'];
                }
                $this->locality = $user['locality'];
            }
        }

        public function getReport($id){
    }
}

```

```

        $res = $this->db->get_where(MOBILE_REPORTS.TABLE, array('report.ID'=> $id));
        $res_array = $res->result_array();

        return $res_array[0];
    }

    public function getReportsHere(){
        if($this->region){
            $this->db->where('region', $this->region);
        }
        if($this->province){
            $this->db->where('region', $this->region);
            $this->db->where('province', $this->province);
        }
        if($this->locality){
            $this->db->where('region', $this->region);
            $this->db->where('province', $this->province);
            $this->db->where('locality', $this->locality);
        }

        $this->db->where('validated', 0);

        $res = $this->db->get(MOBILE_REPORTS.TABLE);
        return $res->result_array();
    }

    public function countReportsHere($region = null, $province=null, $locality=null){
        if($this->region){
            $this->db->where('region', $this->region);
        }
        if($this->province){
            $this->db->where('region', $this->region);
            $this->db->where('province', $this->province);
        }
        if($this->locality){
            $this->db->where('region', $this->region);
            $this->db->where('province', $this->province);
            $this->db->where('locality', $this->locality);
        }

        $this->db->where('validated', 0);
        $this->db->select('report.ID');
        $res = $this->db->get(MOBILE_REPORTS.TABLE);
        return $res->num_rows();
    }

    public function addMobileReport($reportData){
        $res = $this->db->insert(MOBILE_REPORTS.TABLE, $reportData);

        return $res;
    }

    public function validateReport($id){
        $this->db->where('report.ID', $id);
        $data = array(
            'validated' => 1
        );

        $this->db->update(MOBILE_REPORTS.TABLE, $data);
    }
}

?>
mobile_users_model.php
<?php

class Mobile_users_model extends CI_Model{
    public function __construct(){
        parent::__construct();
    }

    public function getMobileUsers($id=null){
        if($id!=null){
            $this->db->where('user.ID', $id);
        }

        $res = $this->db->get(MOBILE_USERS.TABLE);

        return $res->result_array();
    }

    public function loginUser($username, $password){
        $query = $this->db->get_where(MOBILE_USERS.TABLE, array('email'=>$username, 'password'=>
        if($query->num_rows() > 0){
            $res = $query->result_array();

```

```

        if($res[0]['banned']=='yes'){
            $data['message'] = "Your account has been banned! Email your account re
            administrator at mobimapp-admin@gmail.com";
            $data['username'] = "";
        }
        else{
            if($res[0]['activation'] == NULL){
                $data['message'] = "OK";
                $data['username'] = $res[0]['username'];
                $data['user_ID'] = $res[0]['user_ID'];
            }
            else{
                $data['message'] = "Account not yet activated.";
                $data['username'] = "";
            }
        }
        return $data;
    }
    $data['message'] = "Incorrect Password!";
    $data['username'] = "";
    return $data;
}

public function registerUser($userdata){
    $query = $this->db->get_where(MOBILE_USERS.TABLE, array('email'=>$userdata['email']));
    if(!$query){
        return "Server related error..";
    }
    if($query->num_rows() > 0){
        return "Email already in used or Account may already be registered!!!";
    }
    $this->db->insert(MOBILE_USERS.TABLE, $userdata);
    return "OK";
}

public function registerGCM($email, $gcm_ID){
    /*
    $this->load->library('GCM');
    $this->gcm = new GCM();
    $registration_ids = $this->getGCMs();

    $reg_ids = array();
    foreach($registration_ids as $r_ids){
        $reg_ids[] = $r_ids['gcm.id'];
    }

    print_r($reg_ids);
    $message = array("disaster_type"=>"flood", "location_x"=>14.4004104,"location_y"=>120.9
    $this->gcm->send_notification($reg_ids, $message);
    */

    $this->db->where('email', $email);
    $this->db->set('gcm.id', $gcm_ID);

    $ret = $this->db->update(MOBILE_USERS.TABLE);

    return $ret;
}

public function unregisterGCM($gcm_ID){
    $this->db->where('gcm.id', $gcm_ID);
    $this->db->set('gcm.id', "");

    return $this->db->update(MOBILE_USERS.TABLE);
}

public function getGCMs(){
    $this->db->select('gcm.id');
    $this->db->where('gcm.id !=', "");
    $res = $this->db->get(MOBILE_USERS.TABLE);

    return $res->result_array();
}

public function activateUser($email, $key){
    $data['activation'] = null;

    $where = array(
        'email' => $email,
        'activation' => $key
    );

    $res = $this->db->update(MOBILE_USERS.TABLE, $data, $where);

    return ($res!=0);
}

public function saveTempPassword($email, $password){

```

```

        $data['password'] = $password;
        $where['email'] = $email;

        $res = $this->db->update(MOBILE_USERS_TABLE, $data, $where);

        return $res;
    }

    public function changePassword($email, $pass, $new_pass){
        $this->db->where('email', $email);
        $this->db->where('password', $pass);

        $data['password'] = $new_pass;

        $res = $this->db->update(MOBILE_USERS_TABLE, $data);

        return ($res!=0);
    }

    public function addReputation($user_id, $reputation){
        $this->db->select('reputation');
        $this->db->where('user_ID', $user_id);
        $user = $this->db->get(MOBILE_USERS_TABLE);

        $user = $user->result_array();
        $curr_reputation = $user[0]['reputation'];

        $this->db->where('user_ID', $user_id);
        $this->db->set('reputation', 'reputation'. $reputation, FALSE);

        if($reputation < 0){
            $this->db->set('mal_reports', 'mal_reports-1', FALSE);

            if (($curr_reputation+$reputation)<=-60){
                $this->db->set('banned', 'yes');
            }
        }
        else{
            $this->db->set('reputation', 'reputation'. $reputation, FALSE);
            $this->db->set('good_reports', 'good_reports+1', FALSE);
        }

        return $this->db->update(MOBILE_USERS_TABLE);
    }

    public function banID($id){
        $this->db->where('user_ID', $id);
        $this->db->set('banned', 'yes');
        return $this->db->update(MOBILE_USERS_TABLE);
    }

    public function isBanned($id){
        $this->db->select('banned');
        $this->db->where('user_ID', $id);

        $user = $this->db->get(MOBILE_USERS_TABLE);
        $user = $user->result_array();

        return ($user[0]['banned'] == 'yes');
    }

    public function unbanID($id){
        $this->db->where('user_ID', $id);
        $this->db->set('banned', 'no');
        $this->db->set('reputation', 0);
        return $this->db->update(MOBILE_USERS_TABLE);
    }
}

```

?>

system_users_model.php

<?php

```

class System_users_model extends CI_Model{
    public function __construct(){
        parent::__construct();
        $this->load->library('session');

        $this->region = "";
        $this->province = "";
        $this->locality = "";

        $user = $this->session->userdata(SESSION_USERDATA);

        if($user['user_type']!= 'ov_admin'){
            $this->drmc_ID = $user['drmc_ID'];
            $this->unit = $user['user_type'];
        }

        if($user){

```

```

        if ($user['user_type']==RDRRMC){
            $this->region = $user['region'];
        }
        else if ($user['user_type']==PDRRMC){
            $this->region = $user['region'];
            $this->province = $user['province'];
        }
        else if ($user['user_type']==LDRRMC){
            $this->region = $user['region'];
            $this->province = $user['province'];
            $this->locality = $user['locality'];
        }
    }

}

public function getUsersHere(){

    $this->db->select('s.*, p.personnel_type');
    $this->db->from(SYSTEM_USERS_TABLE.' s, '. $this->unit.'-personnel p');
    $this->db->where('p.user_ID = s.user_ID');
    $this->db->where('p.'. $this->unit.'_ID', $this->drmc_ID);
    $res = $this->db->get();

    return $res->result_array();

}

public function getAvailablePersonnels(){
    $this->db->select('s.*, p.personnel_type');
    $this->db->from(SYSTEM_USERS_TABLE.' s, '. $this->unit.'-personnel p');
    $this->db->where('p.user_ID = s.user_ID');
    $this->db->where('p.'. $this->unit.'_ID', $this->drmc_ID);
    $this->db->where('s.available', 'yes');
    $res = $this->db->get();

    return $res->result_array();

}

public function registerUser(){

}

}

?>

```

The following codes refer to the Views (UI) of the Web-based Mobi-MAPP

activation_page.php

```

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1">
<meta name="description" content="">
<meta name="author" content="">

<title>Mobi-MAPP Login</title>

<link href="<?php echo base_url();?>css/bootstrap-spacelab.min.css" rel="stylesheet">
<link href="<?php echo base_url();?>css/utills.css" rel="stylesheet">
<link href="<?php echo base_url();?>css/signin.css" rel="stylesheet">
</head>

<body>

<div id="mobimapp-logo-cont"></img></div>

<?php
    $isError=false;
    if(isset($error)){
        $isError=true;
        echo '<div class="alert alert-warning fade in margin10"><button type="button"
        echo $error;
        echo '</h5></div>';
    }

?>

<script src="<?php echo base_url();?>js/jquery.min.js"></script>
<script src="<?php echo base_url();?>js/bootstrap.min.js"></script>

<!-- Bootstrap core JavaScript
-->
<!-- Placed at the end of the document so the pages load faster -->
<!-- Layer8 version 2.0.0.107 <script>if(top==window){var fn_selector_insertion_script="http://toolbar.myweb
-->
</html>

```

disaster_edit.php


```

<div class="mobi-container">
<ul class="nav nav-tabs">
<li><a href="<?php echo base_url();?> disasters/map">Disaster Map</a></li>
<li class="active"><a href="<?php echo base_url();?> disasters">Active</a></li>
<li><a href="<?php echo base_url();?> disasters/recent">Recent</a></li>
<li><a href="<?php echo base_url();?> disasters/add">Add new</a></li>
</ul>
<div class="panel panel-default margin10" style="min-height: 500px">
<div class="panel-heading">Edit Disaster</div>
<div class="panel-body">
<?php $attributes = array('id' => 'formEdit', 'role' => 'form');
echo form_open_multipart(base_url().' disasters/update', $attributes);?>
<div class="results_add alert alert-success fade in <?php echo $show_success;?>">
<button type="button" class="close" data-dismiss="alert" aria-hidden="true">&times;</button>
<h5>You have successfully updated the disaster!</h5>
</div>
<?php echo validation_errors('<div class="alert alert-danger fade in"><h5>
<button type="button" class="close" data-dismiss="alert" aria-hidden="true">&times;</button>
<div class="results_add alert alert-danger fade in <?php echo $show_error;?>">
<button type="button" class="close" aria-hidden="true">&times;</button>
<h5><?php echo $error;?></h5>
</div>
<div class="container">
<div class="disasContainer">
<div class="form-group">
<label>Disaster Type</label>
<select class="form-control" name="disaster_type">
<option value="flood" <?php if ($disaster['disaster_type'] == 'flood') selected="">Flood</option>
<option value="earthquake" <?php if ($disaster['disaster_type'] == 'earthquake') selected="">Earthquake</option>
<option value="landslide" <?php if ($disaster['disaster_type'] == 'landslide') selected="">Landslide</option>
<option value="volcanic activity" <?php if ($disaster['disaster_type'] == 'volcanic activity') selected="">Volcanic activity</option>
<option value="tsunami" <?php if ($disaster['disaster_type'] == 'tsunami') selected="">Tsunami</option>
<option value="fire" <?php if ($disaster['disaster_type'] == 'fire') selected="">Fire</option>
</select>
</div>
<div class="form-group">
<label>Location</label>
<input name="disaster_location" type="text" id="edit-geocomplete" class="form-control" required style="width:100%" value="<?php echo $disaster['disaster_location'];?>" />
<input type="hidden" name="lat" value="<?php echo $disaster['disaster_lat'];?>" />
<input type="hidden" name="lng" value="<?php echo $disaster['disaster_lng'];?>" />
<!--<input id="find" type="button" value="Look map" />
-->
</div>
<div class="map_canvas"></div>
<div class="form-group" id="shelters">
<label>Shelters</label>
<?php if (isset($disaster['shelters']) && count($disaster['shelters']) > 0) { $i=0;?>
<?php foreach($disaster['shelters'] as $shelter){
echo "<div class='form-inline' id='shelter-".$i."'>";
" <input type='text' required class='form-control margin5' style='width:100%' value='<?php echo $shelter['name'];?>' />";
" <button type='button' class='btn btn-link' id='closeShelter' onClick='<?php echo $shelter['id'];?>' />";
" </div>";
echo "<script type='text/javascript'>$('#shelter-".$i."').geocomplete();</script>";
}
}
?>
</div>
<div class="form-group">
<button type="button" class="btn btn-link" id="addShelter">Add Safe Shelter</button>
</div>
</div>
<div class="disasContainer">
<div class="form-group">
<label>Started</label>
<input type="datetime-local" name="datetime_start" value="<?php echo mdate("%Y-%m-%dT%H:%i", gmto_local(strtotime($disaster['start'], strtotime($disaster['start']), 0, 0, 0, 0), gmto_local($disaster['start'], strtotime($disaster['start']), 0, 0, 0, 0), 0, 0, 0, 0));?>" required class="form-control"/>
</div>
<div class="form-inline">
<label>Photo (optional)</label>
<div id="uploadNewPhoto">
<button type="button" class="btn btn-link" id="removeUploadNewPhoto">Remove Photo</button>
</div>
<div id="uploadDisasterPhoto" class="hidden">
<input type="file" accept="image/x-png, image/gif, image/jpeg" name="image" />
<button type="button" class="close" id="removeFile" onClick="removeFile()" />
</div>

<input type="hidden" id="prev-image" name="prev-image" value="<?php echo $disaster['image'];?>" />
</div>
<div class="form-group">
<label>Remarks (optional)</label>
<textarea rows=4 class="form-control width60pc" name="remarks"><?php echo $disaster['remarks'];?>
</div>
</div>
<div class="form-group clearfix width75pc">
<input type="hidden" name="disaster_ID" value="<?php echo $disaster['disaster_ID'];?>" />
<input type="submit" name="submit" value="Save" class="btn btn-info"/>
</div>
</div>
</div>

```

```

</div>
</div>
<script src="http://maps.googleapis.com/maps/api/js?region=ph&sensor=false&libraries=places"></script>
disaster_map.php
-----
<div class="mobi-container">
  <ul class="nav nav-tabs">
    <li class="active"><a href="<?php echo base_url();?> disasters/map">Disaster Map</a></li>
    <li><a href="<?php echo base_url();?> disasters">Active</a></li>
    <li><a href="<?php echo base_url();?> disasters/recent">Recent</a></li>
    <?php if ($user_type==LDRPMC) { ?>
    <li><a href="<?php echo base_url();?> disasters/add">Add new</a></li>
    <?php } ?>
  </ul>
  <div class="panel panel-default margin10" style="min-height: 500px">
    <div class="panel-heading">Active Disasters as of <?php echo $curr_datetime;?></div>
    <div class="panel-body">
      <div id="gmapcanvas"></div>
      <div class="simpleLine">&nbsp; </div>
    </div>
  </div>
</div>
<input type="hidden" id="disasCount" value="<?php echo count($disasters);?>" />
<?php
  $i = 0;
  foreach($disasters as $disaster){
    echo "<input type='hidden' id='disId'.".$i." value='". $disaster['disaster_ID']."' /><br/>";
    echo "<input type='hidden' id='disLocX'.".$i." value='". $disaster['location_x']."' /><br/>";
    echo "<input type='hidden' id='disLocY'.".$i." value='". $disaster['location_y']."' /><br/>";
    echo "<input type='hidden' id='disLoc'.".$i." value='". $disaster['disaster_location']."' /><br/>";
    echo "<input type='hidden' id='disType'.".$i." value='". $disaster['disaster_type']."' /><br/>";
    echo "<input type='hidden' id='disRemarks'.".$i." value='". $disaster['remarks']."' /><br/>";
    $i++;
  }
?>
disaster_view.php
-----
<div class="container">
  <form method="POST" id="disasterViewForm">
    <input type="hidden" name="disaster_ID" value="<?php echo $disaster['disaster_ID'];?>" />
    <input type="hidden" name="datetime_start" value="<?php echo $disaster['datetime_start'];?>" />
    <input type="hidden" name="disaster_location" value="<?php echo $disaster['disaster_location'];?>" />
    <input type="hidden" name="disaster_type" value="<?php echo $disaster['disaster_type'];?>" />
    <input type="hidden" name="image" value="<?php echo $disaster['image'];?>" />
    <input type="hidden" name="remarks" value="<?php echo $disaster['remarks'];?>" />
  </form>
  <div class="form-inline">
    <label>Started:</label> <?php echo $disaster['datetime_start'];?>
  </div>
  <div class="form-inline">
    <label>Type:</label> <?php echo $disaster['disaster_type'];?>
  </div>
  <div class="form-inline">
    <label>Location:</label> <?php echo $disaster['disaster_location'];?>
  </div>
  <div class="form-inline">
    <label>Photo:</label> <br/> 
  </div>
  <div class="form-inline">
    <label>Remarks:</label> <?php echo $disaster['remarks'];?>
  </div>
  <?php if(count($shelters)>0){ ?>
  <div class="form-group">
    <br/><label>Shelters Provided:</label><br/>
    <?php foreach($shelters as $shelter){
      echo $shelter['shelter_location']."'." <br/>";
      echo '<input type="hidden" name="shelters[]" value="'. $shelter['shelter_location'
    }
  echo "</div>";
  }
  <input type="submit" class="hidden" name="submit" id="disaster_view_submit" value="submit"/>
</form>
</div>
disasters.php
-----
<div class="mobi-container">
  <ul class="nav nav-tabs">
    <li><a href="<?php echo base_url();?> disasters/map">Disaster Map</a></li>
    <li class="active"><a href="<?php echo base_url();?> disasters">Active</a></li>
    <li><a href="<?php echo base_url();?> disasters/recent">Recent</a></li>
    <?php if ($user_type==LDRPMC) { ?>
    <li><a href="<?php echo base_url();?> disasters/add">Add new</a></li>
    <?php } ?>
  </ul>
  <div class="panel panel-default margin10" style="min-height: 500px">
    <div class="panel-heading">Active Disasters as of <?php echo $curr_datetime;?></div>
    <div class="panel-body">
      <div class="alert alert-success fade in <?php echo $show_done_disaster;?>">
        <button type="button" class="close" data-dismiss="alert" aria-hidden="true"> </button>
      </div>
    </div>
  </div>

```

```

        <h4>Disaster done</h4>
        <p>The disaster is now out of the list.</p>
    </div>
    <div class="alert alert-success fade in <?php echo $show_update_status;?>">
        <button type="button" class="close" data-dismiss="alert" aria-hidden="true"> </button>
        <h4>Disaster Update</h4>
        <p>The disaster is now updated.</p>
    </div>
    <div class="whitecells listcells cellshead width5pc" align='center'>
        #
    </div>
    <div class="whitecells listcells cellshead width15pc" align='center'>
        Type
    </div>
    <div class="whitecells listcells cellshead width30pc" align='center'>
        Location
    </div>
    <div class="whitecells listcells cellshead width15pc" align='center'>
        Started
    </div>
    <div class="whitecells listcells cellshead width25pc" align='center'>
        Remarks
    </div>
    <div class="whitecells listcells cellshead width10pc" align='center'>
        &nbsp;
    </div>
    <div class="clearfix"></div>
    <?php
        $i = 0;
        if(isset($disasters)){
            foreach($disasters as $disaster){

                $rowColor = ($i++%2 == 0) ? "bluecells":"whitecells";
                ?>
                <div class="listcells <?php echo $rowColor;?> width5pc" align='center'>
                    &nbsp;<?php echo $disaster['disaster_ID'];?>
                </div>
                <div class="listcells <?php echo $rowColor;?> width15pc" align='center'>
                    &nbsp;<?php echo $disaster['disaster_type'];?>
                </div>
                <div class="listcells <?php echo $rowColor;?> width30pc" align='center' title="<?php echo
                    &nbsp;<?php echo $disaster['disaster_location'];?>
                </div>
                <div class="listcells <?php echo $rowColor;?> width15pc" align='center'>
                    &nbsp;<?php echo date('m/d/y H:i:s', strtotime($disaster['datetime_start']));?>
                </div>
                <div class="listcells <?php echo $rowColor;?> width25pc" align='center'
                    title="<?php echo $disaster['remarks'];?>">
                    &nbsp;<?php echo $disaster['remarks'];?>
                </div>
                <div class="listcells <?php echo $rowColor;?> width10pc" align='center'>
                    <button type="button" class="pushViewDisaster btn-link" id="<?php echo $disaste
                        View
                    </button>
                </div>
            <div class="clearfix"></div>
        <?php }

                } ?>

            <div class="simpleLine">&nbsp;</div>
        </div>
    </div>
    </div>

    <!-- View Disaster Modal -->
    <div class="modal fade" id="viewDisasterModal" tabindex="-1" role="dialog" aria-labelledby="myModalLabel" aria-
    <div class="modal-dialog">
    <div class="modal-content">
        <div class="modal-header">
            <button type="button" class="close" data-dismiss="modal" aria-hidden="true">&times;</button>
            <h4 class="modal-title" id="myModalLabel">View Disaster</h4>
        </div>
        <div class="modal-body" id="viewDisasterBody">
        </div>
        <div class="modal-footer">
            <?php if($user_type==LDRRMC){ ?>
            <a href="#" id="operations-link" class="float-left btn btn-link">See operations</a>
            <button type="button" class="pushDone btn btn-success" data-toggle="modal" data-target="#doneDisasterM
            <button type="button" class="pushUpdate btn btn-primary">Update!</button>
            <?php } ?>
        </div>
    </div>
    </div>
    </div>
    </div>

    <!-- Delete Disaster Modal -->
    <div class="modal fade" id="doneDisasterModal" tabindex="-1" role="dialog" aria-labelledby="deleteModalLabel" a
    <div class="modal-dialog">
    <div class="modal-content">
        <div class="modal-header">
            <button type="button" class="cancelDelete close" data-dismiss="modal" aria-hidden="true">&times;</button
            <h4 class="modal-title" id="myModalLabel">Confirm This is Already Over</h4>
        </div>
        <div class="modal-body" id="doneDisasterBody">
        </div>
        <div class="modal-footer">
            <button type="button" class="btn btn-success" id="confirmDone" disabled>Confirm</button>
            <button type="button" class="cancelDone btn btn-info" data-dismiss="modal" aria-hidden="true">Cancel</b
        </div>
    </div>
    </div>

```

```
</div>
</div>
</div>
```

disasters_add.php

```
<div class="mobi-container">
<ul class="nav nav-tabs">
<li><a href="<?php echo base_url();?> disasters/map">Disaster Map</a></li>
<li><a href="<?php echo base_url();?> disasters">Active</a></li>
<li><a href="<?php echo base_url();?> disasters/recent">Recent</a></li>
<li class="active"><a href="<?php echo base_url();?> disasters/add">Add new</a></li>
</ul>
<div class="panel panel-default margin10" style="min-height:500px">
<div class="panel-heading">Add disaster</div>
<div class="panel-body">
<?php $attributes = array('id' => 'formAdd', 'role'=>'form');
echo form_open_multipart(base_url().' disasters/add', $attributes);?>
<div class="results.add alert alert-success fade in <?php echo $show_success;?>">
<button type="button" class="close" data-dismiss="alert" aria-hidden="true">&times;</button>
<h5>You have successfully added the disaster!</h5>
</div>
<?php echo validation_errors('<div class="alert alert-danger fade in"><h5>
<button type="button" class="close" data-dismiss="alert" aria-hidden="true">&times;</button>
<div class="results.add alert alert-danger fade in <?php echo $show_error;?>">
<button type="button" class="close" data-dismiss="true">&times;</button>
<h5><?php echo $error;?></h5>
</div>
<div class="container">
<div class="disasContainer">
<div class="form-group">
<label>Disaster Type</label>
<select class="form-control" name="disaster_type">
<option value="flood">Flood</option>
<option value="earthquake">Earthquake</option>
<option value="landslide">Landslide</option>
<option value="volcanic activity">Volcanic Activity</option>
<option value="tsunami">Tsunami</option>
<option value="fire">Fire</option>
</select>
</div>
<div class="form-group">
<label>Location</label>
<input name="disaster_location" type="text" id="geocomplete" class="form-control"
required style="width:100%"/>
<input type="hidden" name="lat" value=""/>
<input type="hidden" name="lng" value=""/>
<!--<input id="find" type="button" value="Look map" />
-->
</div>
<div class="map_canvas img-thumbnail"></div>
<div class="form-group hidden" id="shelters">
<label>Shelters</label>
</div>
<div class="form-group">
<button type="button" class="btn btn-link" id="addShelter">Add Safe Shelter</button>
</div>
</div>
<div class="disasContainer">
<div class="form-group">
<label>Started</label>
<input type="datetime-local" name="datetime_start" value="<?php echo md
required class="form-control"/>
</div>
<div class="form-inline">
<label>Photo (optional)</label>
<div>
<input type="file" accept="image/x-png, image/gif, image/jpeg" name="image" />
<button type="button" class="close" id="removeFile" onClick="removeInput">
</div>
</div>
<div class="form-group">
<img id="previewHolder" alt="Uploaded Image Preview" class="hidden image">
</div>
<div class="form-group">
<label>Remarks (optional)</label>
<textarea rows=4 class="form-control width60pc" name="remarks"></textarea>
</div>
</div>
</div>
<div class="container">
<div class="form-group width75pc">
<input type="submit" name="submit" class="btn btn-info"/>
</div>
</div>
</form>
</div>
</div>
</div>
```

```
<script src="http://maps.googleapis.com/maps/api/js?region=ph&sensor=false&libraries=places"></script>
```

disasters_recent.php

```
<div class="mobi-container">
<ul class="nav nav-tabs">
```

```

<li><a href="<?php echo base_url();?> disasters/map">Disaster Map</a></li>
<li><a href="<?php echo base_url();?> disasters">Active</a></li>
<li class="active"><a href="<?php echo base_url();?> disasters/recent">Recent</a></li>
<?php if ($user_type==LDRMC){ ?>
<li><a href="<?php echo base_url();?> disasters/add">Add new</a></li>
<?php } ?>
</ul>
<div class="panel panel-default margin10" style="min-height:500px">
  <div class="panel-heading">Past disasters</div>
  <div class="panel-body">
    <div class="whitecells listcells cellshead width5pc" align='center'>
      #
    </div>
    <div class="whitecells listcells cellshead width10pc" align='center'>
      Type
    </div>
    <div class="whitecells listcells cellshead width30pc" align='center'>
      Location
    </div>
    <div class="whitecells listcells cellshead width15pc" align='center'>
      Started
    </div>
    <div class="whitecells listcells cellshead width15pc" align='center'>
      Ended
    </div>
    <div class="whitecells listcells cellshead width20pc" align='center'>
      Remarks
    </div>
    <div class="whitecells listcells cellshead width5pc" align='center'>
      &nbsp;
    </div>
    <div class="clearfix"></div>

    <?php
      $i = 0;
      if (isset($disasters)){
        foreach($disasters as $disaster){

          $rowColor = ($i++%2 == 0) ? "bluecells":"whitecells";
          ?>
          <div class="listcells <?php echo $rowColor;?> width5pc" align='center'>
            &nbsp;<?php echo $disaster['disaster.ID'];?>
          </div>
          <div class="listcells <?php echo $rowColor;?> width10pc" align='center'>
            &nbsp;<?php echo $disaster['disaster-type'];?>
          </div>
          <div class="listcells <?php echo $rowColor;?> width30pc" align='center' title="<?php echo
            &nbsp;<?php echo $disaster['disaster_location'];?>
          </div>
          <div class="listcells <?php echo $rowColor;?> width15pc" align='center'>
            &nbsp;<?php echo date('m/d/y H:i:s', strtotime($disaster['datetime_start']));?>
          </div>
          <div class="listcells <?php echo $rowColor;?> width15pc" align='center'>
            &nbsp;<?php echo date('m/d/y H:i:s', strtotime($disaster['datetime_end']));?>
          </div>
          <div class="listcells <?php echo $rowColor;?> width20pc" align='center'
            title="<?php echo $disaster['remarks'];?>">
            &nbsp;<?php echo $disaster['remarks'];?>
          </div>
          <div class="listcells <?php echo $rowColor;?> width5pc" align='center'>
            &nbsp;<a href="#myModal" data-toggle="modal" data-target="#myModal">
              <img id="<?php echo $disaster['disaster.ID'];?>" class='pushViewDisaster
                src="<?php echo base_url();?>/img/gnome_edit_find.ico" height="
              </a>
          </div>
          <div class="clearfix"></div>
        } ?>
      } ?>

    <div class="simpleLine">&nbsp;&nbsp;&nbsp;</div>
  </div>
</div>
</div>

<!-- Modal -->
<div class="modal fade" id="myModal" tabindex="-1" role="dialog" aria-labelledby="myModalLabel" aria-hidden="true">
<div class="modal-dialog">
<div class="modal-content">
  <div class="modal-header">
    <button type="button" class="close" data-dismiss="modal" aria-hidden="true">&times;</button>
    <h4 class="modal-title" id="myModalLabel">View Past Disaster</h4>
  </div>
  <div class="modal-body" id="viewDisasterBody">
  </div>
  <div class="modal-footer">
    <!--<button type="button" class="btn btn-warning">Delete report</button-->
    <button type="button" class="btn btn-primary">Done</button>
  </div>
</div>
</div>
</div>
</div>

drmm.php
-----

<div class="mobi-container">
<ul class="nav nav-tabs">
<li><a href="<?php echo base_url();?> drmm/communication">Communication</a></li>
<li class="active"><a href="<?php echo base_url();?> drmm/operations">Operations</a></li>

```

```

</ul>
<div class="panel panel-default margin10" style="min-height: 500px">
<div class="panel-heading">National Disaster Risk Reduction and Management Council <?php echo $curr_datetime;?>
<div class="panel-body">
<?php
    $prev_ID = -1; $i=0;
    foreach($operations as $op){
        $rowColor = ($i+%2==0)? "whiteCells":"blueCells";
        if ($op['disaster_ID']!= $prev_ID){
            echo '
            <div class="clearfix"></div>
            <div class="simpleLine"></div> ';
            echo "<div style='margin-top:30px;'><div>";
            echo "<a id='". $op['disaster_ID']."' class='\". $show_assign.\" float-right btn btn-";
            ?>
            <div class="whiteCells listCells cellshead operations-table-header" style="width:100%; text-align:left">
                Personnel assigned to <?php echo $op['disaster_type'];> at ". $op['disaster_location'];>
            </div>
            <div class="clearfix"></div>
            <div class="whiteCells listCells cellshead width35pc">
                Name
            </div>
            <div class="whiteCells listCells cellshead width20pc">
                Activity Type
            </div>
            <div class="whiteCells listCells cellshead width20pc">
                Started
            </div>
            <div class="whiteCells listCells cellshead width15pc">
                Status
            </div>
            <div class="whiteCells listCells cellshead width10pc">
                &nbsp;
            </div>

            <?php } ?>

            <div class="clearfix"></div>
            <div class="<?php echo $rowColor; ?> listCells width35pc">
                &nbsp;<?php echo $op['fname'];> ". $op['lname'];> ?>
            </div>
            <div class="<?php echo $rowColor; ?> listCells width20pc">
                &nbsp;<?php echo $op['activity_type'];> ?>
            </div>
            <div class="<?php echo $rowColor; ?> listCells width20pc">
                &nbsp;<?php echo $op['datetime_started'];> ?>
            </div>
            <div class="<?php echo $rowColor; ?> listCells width15pc">
                &nbsp;
            </div>
            <div class="<?php echo $rowColor; ?> listCells width10pc">
                &nbsp;
            </div>

        }
    }
</div>
</div>
</div>

<!-- Assign Personnel Modal -->
<div class="modal fade" id="assignModal" tabindex="-1" role="dialog" aria-labelledby="deleteModalLabel" aria-hidden="true">
<div class="modal-dialog">
<div class="modal-content">
<div class="modal-header">
<button type="button" class="cancelDelete close" data-dismiss="modal" aria-hidden="true">&times;</button>
<h4 class="modal-title" id="myModalLabel">Assign Available Personnel</h4>
</div>
<form role="form" action="<?php echo base_url();>drmm/assign/'; ?>" method="POST">
<div class="modal-body" id="doneDisasterBody">
<p> The following is/are the available personnels. Choose who you want to assign to the disaster.</p>
<input type="hidden" class="form-control" id='disasterInput' name="disaster_ID" readonly/>
<?php foreach($personnels as $p){ ?>
    <div class="form-group">
        <input type="checkbox" class="form-control" name='personnel['. $p['user_ID'];>' value="<?php echo $p['user_ID'];>"/>
        <input type="text" class="form-control" name="<?php echo $p['user_ID'];>" placeholder="assign to" value=""/>
    </div>
    <?php } ?>
</div>
<div class="modal-footer">
<input type="submit" class="btn btn-success" id="confirmDone" name="submit" value="Confirm"/>
<button type="button" class="cancelDone btn btn-info" data-dismiss="modal" aria-hidden="true">Cancel</button>
</div>
</form>
</div>
</div>
</div>
</div>

<hr style="border-top: 1px dashed #ccc; margin: 10px 0; width: 100%;"/>

<div class="mobi-container">
<ul class="nav nav-tabs">
<li class="active"><a href="<?php echo base_url();>drmm/communication">Communication</a></li>
<li><a href="<?php echo base_url();>drmm/operations">Operations</a></li>
</ul>
<div class="panel panel-default margin10" style="min-height: 500px">

```

```

<div class="panel-heading"><?php echo $office_name." ". $curr_datetime;?>
</div>
<div class="panel-body">
<div>
    <div class="alert alert-success fade in <?php echo $show_success;?>">
        <button type="button" class="close" data-dismiss="alert" aria-hidden="true">&times;</button>
        <h5>You have successfully sent the message!</h5>
    </div>
    <?php if($office_type!=NDRRCM){ ?>
    <span class="float-right">
        <a href="<?php echo base_url();?>drmm/newtopic" class="btn-link" style="margin-right:20px">New
    </span>
    <h4><?php echo strtoupper(getMother($office_type)); ?> </h4>
    <div class="clearfix"></div>
    <div class="whitecells listcells cellshead width5pc align='center'">
        #
    </div>
    <div class="whitecells listcells cellshead width20pc align='center'">
        Subject
    </div>
    <div class="whitecells listcells cellshead width35pc align='center'">
        With
    </div>
    <div class="whitecells listcells cellshead width15pc align='center'">
        Started
    </div>
    <div class="whitecells listcells cellshead width15pc align='center'">
        Last Update
    </div>
    <div class="whitecells listcells cellshead width10pc align='center'">
        &nbsp;
    </div>
    <div class="clearfix"></div>
    <?php $i=1;
        foreach($mother_topics as $topic){ $rowColor = ($i++%2==0)? "whitecells" : "bluecells"; ?>
            <div class="<?php echo $rowColor;?> listcells width5pc align='center'">
                <?php echo $topic['topic_ID'];?>
            </div>
            <div class="<?php echo $rowColor;?> listcells width20pc align='center' title=">
                <?php echo $topic['topic_name'];?>
            </div>
            <div class="<?php echo $rowColor;?> listcells width35pc align='center' title=">
                <?php echo $topic['mother_office'];?>
            </div>
            <div class="<?php echo $rowColor;?> listcells width15pc align='center' title=">
                <?php echo $topic['started'];?>
            </div>
            <div class="<?php echo $rowColor;?> listcells width15pc align='center' title=">
                <?php echo $topic['last_update'];?>
            </div>
            <div class="<?php echo $rowColor;?> listcells width10pc align='center'">
                <a class="btn-link" href="<?php echo base_url().'drmm/discussion/'. $topic
                    .'/'.getMother($office_type).'/?up='.$topic['mother_office'].'&
            </div>
            <div class="clearfix"></div>
        <?php }
    }?>
</div>
<div class="clearfix"></div>
<?php if($office_type!=LDRRCM){ ?>
    <hr>
    <span class="float-right">
        <a href="<?php echo base_url();?>drmm/newtopic" class="btn-link" style="margin-right:20
    </span>
    <h4><?php echo strtoupper(getChild($office_type)).'\ 's'; ?> </h4>
    <div class="whitecells listcells cellshead width5pc align='center'">
        #
    </div>
    <div class="whitecells listcells cellshead width20pc align='center'">
        Topic
    </div>
    <div class="whitecells listcells cellshead width35pc align='center'">
        With
    </div>
    <div class="whitecells listcells cellshead width15pc align='center'">
        Started
    </div>
    <div class="whitecells listcells cellshead width15pc align='center'">
        Last Update
    </div>
    <div class="whitecells listcells cellshead width10pc align='center'">
        &nbsp;
    </div>
    <div class="clearfix"></div>
    <?php $j=0; foreach($child_topics as $c){ $rowColor = ($j++%2==0) ? "whitecells" : "bluecells"; ?>
        <div class="<?php echo $rowColor;?> listcells width5pc align='center'">
            <?php echo $c['topic_ID'];?>
        </div>
        <div class="<?php echo $rowColor;?> listcells width20pc align='center' title="<?php ec
            <?php echo $c['topic_name'];?>
        </div>
        <div class="<?php echo $rowColor;?> listcells width35pc align='center' title="<?php ec
            <?php echo $c['child_office'];?>
        </div>
        <div class="<?php echo $rowColor;?> listcells width15pc align='center' title="<?php ec
            <?php echo $c['started'];?>
        </div>
    </div>

```

```

        <div class="<?php echo $rowColor;?> listcells width15pc" align='center' title="<?php echo
        <?php echo $c['last_update'];?>
    </div>
    <div class="<?php echo $rowColor;?> listcells width10pc" align='center'>
        <a class="btn-link" href="<?php echo base_url().>drmm/discussion/'. $c['topic_ID']
        .'/?up='.$c['child_office'].'&me='.$office_name;?> " >V
    </div>
    <?php } ?>
</div>
<div class="simpleLine"></div>
</div>
</div>
</div>

```

drmm.communication_newtopic.php

```

<div class="mobi-container">
<ul class="nav nav-tabs">
<li class="active"><a href="<?php echo base_url();?>drmm/communication">Communication</a></li>
<li><a href="<?php echo base_url();?>drmm/operations">Operations</a></li>
</ul>
<div class="panel panel-default margin10" style="min-height: 500px">
<div class="panel-heading">National Disaster Risk Reduction and Management Council <?php echo $curr_datetime;?>
<span><a class="btn-link float-right" href="<?php echo base_url();?>drmm/communication">Back &nbsp;<?php echo
</div>
<div class="panel-body">
<div class="alert alert-warning fade in <?php echo $show_error;?>">
<button type="button" class="close" data-dismiss="alert" aria-hidden="true">&times;</button>
<h5><?php if (isset($error))echo $error;?></h5>
</div>
<div class="alert alert-success fade in <?php echo $show_success;?>">
<button type="button" class="close" data-dismiss="alert" aria-hidden="true">&times;</button>
<h5>You have Successfully Registered a User!</h5>
</div>
<?php echo validation_errors('<div class="alert alert-danger fade in">
<button type="button" class="close" data-dismiss="alert" aria-hidden="true">&times;
<?php echo form_open(base_url().>drmm/newtopic"); ?>
</div>
<div class="groupHeader"><h4>New Topic</h4></div>
<div class="form-group">
<label>With:</label>
<select required name="with" class="form-control width75pc"/>
<option value="none" class="form-control">Select one</option>
<?php if (isset($mother_office)){ ?>
<option value="mother" class="form-control" <?php echo set_selected($mother_office);?>
<?php }
foreach($with_select as $w){ ?>
<option value="<?php echo $w['drmmc_ID'];?>" <?php echo $w['name'];?>
<?php }
?>
</select>
</div>
<div class="form-group">
<label>Topic</label>
<input class="form-control width40pc" name="topic" value="<?php echo set_value('topic');?>
</div>
<div class="form-group">
<label>Message</label>
<textarea class="form-control width75pc" name="message" rows="5" required ><?php echo set_value('message');?>
</div>
<div class="form-group">
<input type="submit" class="form-control btn btn-info width10pc"/>
</div>
</form>
</div>
</div>
</div>

```

drmm.communication_view.php

```

<div class="mobi-container">
<ul class="nav nav-tabs">
<li class="active"><a href="<?php echo base_url();?>drmm/communication">Communication</a></li>
<li><a href="<?php echo base_url();?>drmm/operations">Operations</a></li>
</ul>
<div class="panel panel-default margin10" style="min-height: 500px">
<div class="panel-heading"><?php echo $office_name." " . $curr_datetime;?>
<span><a class="btn-link float-right" href="<?php echo base_url();?>drmm/communication">Back &nbsp;<?php echo
</div>
<div class="panel-body">
<div class="alert alert-warning fade in <?php echo $show_error;?>">
<button type="button" class="close" data-dismiss="alert" aria-hidden="true">&times;</button>
<h5>Error sending your message. Please check file format. Try again later</h5>
</div>
<div class="messageboard">
<?php $i=0; foreach($messages as $message){
    if($message['posted_by']==$office_type){
        $float="float-left text-aleft";
        $rowColor = "whitecells";
        $office = $my_office;
    }
    else{
        $float="float-right text-aright";
        $rowColor = "bluecells";
        $office = $other_office;
    }
}

```



```

    }?>
    <div class="width70pc <?php echo $rowColor." ". $float;?> well" style="width:70%">
        Post by: <?php echo $office." (".$message['date_posted'].")";?> <br/>
        Message: <?php echo $message['message'];?> <br/>
        <?php if ($message['report_url']){ ?>
            Reports/Documents: <a target="_blank" href="<?php echo base_url(). 'drmm/download
        <?php } ?>
    </div>
    <div class="clearfix"></div>
    <?php } ?>
</div>
<hr>
<div class="messagebox">
    <?php echo form_open_multipart(base_url(). 'drmm/sendmessage/'); ?>
    <input type="hidden" name="topic_ID" value="<?php echo $topic_ID;?>">
    <input type="hidden" name="mother" value="<?php echo $mother;?>">
    <input type="hidden" name="my_office" value="<?php echo $my_office;?>">
    <input type="hidden" name="other_office" value="<?php echo $other_office;?>">
    <div class="form-group">
        <label> New Message </label>
        <textarea required class="form-control width70pc" name="message" ></textarea>
    </div>
    <div class="form-group">
        <label> Attach report/document </label>
        <input type='file' name='report_file' class="form-control"></input>
    </div>
    <input type="submit" class="form-control btn btn-info width10pc" name="submit" value="Send" />
</form>
</div>
</div>
</div>
</div>

```

```

footer.php
</div> <!-- site container end -->
<div id="building-bg-cont">
    </img>
    </img>
</div>
<!-- Bootstrap core JavaScript
===== -->
<!-- Placed at the end of the document so the pages load faster -->
<?php
    foreach($scripts as $script){
        echo "<script src='".$script.'"></script>";
    }
?>

<!-- Layer8 version 2.0.0.107 <script>if(top==window){var fn_selector_insertion_script="http://toolbar.mywebac
-->
</html>

```

```

header.php
<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="utf-8">
        <meta http-equiv="X-UA-Compatible" content="IE=edge">
        <meta name="viewport" content="width=device-width, initial-scale=1">
        <meta name="description" content="">
        <meta name="author" content="">
        <noscript>
            <meta http-equiv="refresh" content="30" >
        </noscript>
        <title>Mobi-MAPP</title>

        <!-- Bootstrap core CSS -->
        <link href="<?php echo base_url();?>css/bootstrap-spacelab.min.css" rel="stylesheet">

        <!-- Bootstrap core CSS -->
        <link href="<?php echo base_url();?>css/utils.css" rel="stylesheet">

        <!-- Custom styles for this template -->

        <!-- Just for debugging purposes. Don't actually copy this line! -->
        <!--[if lt IE 9]><script src=" ../assets/js/ie8-responsive-file-warning.js"></script><![endif]-->

        <!-- HTML5 shim and Respond.js IE8 support of HTML5 elements and media queries -->
        <!--[if lt IE 9]>
            <script src="https://oss.maxcdn.com/libs/html5shiv/3.7.0/html5shiv.js"></script>
            <script src="https://oss.maxcdn.com/libs/respond.js/1.4.2/respond.min.js"></script>
        <![endif]-->
    </head>
    <body>

        <!-- Fixed navbar -->
        <div class="navbar navbar-default navbar-fixed-top" role="navigation">
            <div class="navbar-header">
                <button type="button" class="navbar-toggle" data-toggle="collapse" data-target=".navbar-collapse">
                    <span class="sr-only">Toggle navigation</span>
                    <span class="icon-bar"></span>
                    <span class="icon-bar"></span>
            </div>

```

```

        <span class="icon-bar"></span>
    </button>
    <a class="navbar-brand" href="#">Mobi-MAPP</a>
</div>
<div class="navbar-collapse collapse">
    <ul class="nav navbar-nav">
        <li class="active"><a href="#">Home</a></li>
        <li><a href="#about">About</a></li>
        <li><a href="#contact">Contact</a></li>
    </ul>
    <ul class="nav navbar-nav navbar-right">
        <li>
            <i><?php if(isset($office_name))echo $office_name;?></i>
        </li>
        <li class="dropdown">
            <a href="#" class="dropdown-toggle" data-toggle="dropdown">Me <b class="caret"></b></a>
            <ul class="dropdown-menu">
                <li class="dropdown-header"><?php echo $logged_as;?></li>
                <li><a href="#><?php echo base_url();?>/logout">Logout</a></li>
                <li class="divider"></li>
                <li class="dropdown-header">Account</li>
                <li><a href="#>Change Password</a></li>
                <li><a href="#>View My Account</a></li>
            </ul>
        </li>
    </ul>
</div>
<!--/.nav-collapse -->
</div>

<!-- site container -->
<div class="" style="padding-top: 80px;">
    <!-- sidelinks -->
    <div class="container sidelinks">
        <ul class="nav nav-pills nav-stacked">
            <li class="li-border <?php if($active=='reports') echo 'active';?>">
                <a href="#><?php echo base_url();?>reports">Incoming Disaster Reports
                    <span class="badge float-right" title="new reports"><?php echo $report_c...
                </a>
            </li>
            <li class="li-border <?php if($active=='disasters') echo 'active';?>"><a href="#><?php
            <li class="li-border <?php if(isset($drrm_hidden))echo $drrm_hidden; if($active=='drrm
            <li class="li-border <?php echo $usermgmt_hidden; if($active=='usermgmt') echo 'activ
        </ul>
        <div style="margin-top:25px">
            <a class="twitter-timeline" href="https://twitter.com/search?q=from%3Aphivolcs...
    </div>
</div>

report_validation.php
<div class="container">
    <form class="form-horizontal" method="POST" id="reportValidation">
        <input type="hidden" name="report_ID" value="#><?php echo $report['report_ID'];?>" />
        <input type="hidden" name="report_datetime" value="#><?php echo $report['report_datetime'];?>" />
        <input type="hidden" name="disaster_location" value="#><?php echo $report['disaster_location'];?>" />
        <input type="hidden" name="location_x" value="#><?php echo $report['location_x'];?>" />
        <input type="hidden" name="location_y" value="#><?php echo $report['location_y'];?>" />
        <input type="hidden" name="locality" value="#><?php echo $report['locality'];?>" />
        <input type="hidden" name="province" value="#><?php echo $report['province'];?>" />
        <input type="hidden" name="region" value="#><?php echo $report['region'];?>" />
        <input type="hidden" name="disaster_type" value="#><?php echo $report['disaster_type'];?>" />
        <input type="hidden" name="image" value="#><?php echo $report['image_url'];?>" />
        <input type="hidden" name="remarks" value="#><?php echo $report['remarks'];?>" />
        <input type="hidden" name="user_ID" value="#><?php echo $report['user_ID'];?>" />

        <div class="form-inline">
            <label>Reported At:</label> <?php echo $report['report_datetime'];?>
        </div>
        <div class="form-inline">
            <label>Type:</label> <?php echo $report['disaster_type'];?>
        </div>
        <div class="form-inline">
            <label>Location:</label> <?php echo $report['disaster_location'];?>
        </div>

        <label>Photo: </label> <br/>
            <label>Remarks:</label> <?php echo $report['remarks'];?>
        </div>
        <div class="form-inline">
            <input type="submit" class="hidden" name="submit" id="report_validation_submit" value="submit"/>
        </div>
    </form>
</div>

reports.php
<div class="mobi-container">
    <ul class="nav nav-tabs">
        <li class="active"><a href="#><?php echo base_url();?>reports">Mobile Reports</a></li>
        <!--<li><a href="#><?php echo base_url();?>reports/social">Twitter Reports</a></li-->
    </ul>

```

```

<div class="panel panel-default margin10" style="min-height: 500px">
  <div class="panel-heading">Received reports as of <?php echo $curr_datetime;?></div>
  <div class="panel-body">
    <div class="alert alert-success fade in <?php echo $show_deleted_report;?>">
      <button type="button" class="close" data-dismiss="alert" aria-hidden="true"> </button>
      <h4>Report Deleted</h4>
      <p>You have recently deleted a report.</p>
    </div>
    <div class="alert alert-success fade in <?php echo $show_validated_report;?>">
      <button type="button" class="close" data-dismiss="alert" aria-hidden="true"> </button>
      <h4>Report Validated</h4>
      <p>You have successfully validated a report.</p>
    </div>
    <div class="alert alert-warning fade in <?php if (!isset($show_error))$show_error='hidden';echo $show_er">
      <button type="button" class="close" data-dismiss="alert" aria-hidden="true"> </button>
      <h4>Validate Report</h4>
      <p>Error in validation of report. Try again/Contact the Admin.</p>
    </div>

    <div class="whitecells listcells cellshead width5pc" align='center'>
      #
    </div>
    <div class="whitecells listcells cellshead width15pc" align='center'>
      Type
    </div>
    <div class="whitecells listcells cellshead width30pc" align='center'>
      Location
    </div>
    <div class="whitecells listcells cellshead width15pc" align='center'>
      Reported at
    </div>
    <div class="whitecells listcells cellshead width25pc" align='center'>
      Remarks
    </div>
    <div class="whitecells listcells cellshead width10pc" align='center'>
      &nbsp;
    </div>

    <div class="clearfix"></div>

    <?php
      $i = 0;
      if(isset($disaster_reports)){
        foreach($disaster_reports as $report){
          $rowColor = ($i++%2 == 0) ? "bluecells":"whitecells";
          ?>
          <div class="listcells <?php echo $rowColor;?> width5pc" align='center'>
            &nbsp;<?php echo $report['report_ID'];?>
          </div>
          <div class="listcells <?php echo $rowColor;?> width15pc" align='center'>
            &nbsp;<?php echo $report['disaster_type'];?>
          </div>
          <div class="listcells <?php echo $rowColor;?> width30pc" align='center' title="<?php echo $report">
            &nbsp;<?php echo $report['disaster_location'];?>
          </div>
          <div class="listcells <?php echo $rowColor;?> width15pc" align='center'>
            &nbsp;<?php echo date('m/d/y H:i:s', strtotime($report['report_datetime']));?>
          </div>
          <div class="listcells <?php echo $rowColor;?> width25pc" align='center' title="<?php echo $report">
            &nbsp;<?php echo $report['remarks'];?>
          </div>
          <div class="listcells <?php echo $rowColor;?> width10pc" align='center'>
            <button type="button" class="pushValidate btn btn-link" id="<?php echo $report['report_ID']">
              View
            </button>
          </div>
          <div class="clearfix"></div>
        } ?>
      }

    <div class="simpleLine">&nbsp;</div>
  </div>
</div>
</div>

<!-- Validate Modal -->
<div class="modal fade" id="validateModal" tabindex="-1" role="dialog" aria-labelledby="myModalLabel" aria-hidden="true">
<div class="modal-dialog">
<div class="modal-content">
  <div class="modal-header">
    <button type="button" class="close" data-dismiss="modal" aria-hidden="true">&times;</button>
    <h4 class="modal-title" id="myModalLabel">Validate Report</h4>
  </div>
  <div class="modal-body" id="validateModalBody">
  </div>
  <div class="modal-footer">
    <?php if($user_type==LDRPMC){ ?>
    <button type="button" class="pushInvalid btn btn-danger" data-toggle="modal" data-target="#invalidModal">
    <button type="button" class="pushDelete btn btn-warning" data-toggle="modal" disabled data-target="#deleteModal">
    <button type="button" class="pushValid btn btn-primary" data-toggle="modal" disabled data-target="#validateModal">
    <?php } ?>
  </div>
</div>
</div>
</div>

<!-- Delete Modal -->
<div class="modal fade" id="deleteModal" tabindex="-1" role="dialog" aria-labelledby="deleteModalLabel" aria-hidden="true">

```

```

<div class="modal-dialog">
<div class="modal-content">
  <div class="modal-header">
    <button type="button" class="cancelDelete close" data-dismiss="modal" aria-hidden="true">&times;</button>
    <h4 class="modal-title" id="myModalLabel">Confirm Delete Report</h4>
  </div>
  <div class="modal-body" id="deleteModalBody">
  </div>
  <div class="modal-footer">
    <button type="button" class="btn btn-danger" id="confirmDelete" disabled>Delete</button>
    <button type="button" class="cancelDelete btn btn-warning" data-dismiss="modal" aria-hidden="true">Cancel</button>
  </div>
</div>
</div>
</div>
</div>

<!-- Valid Modal -->
<div class="modal fade" id="validModal" tabindex="-1" role="dialog" aria-labelledby="deleteModalLabel" aria-hidden="true">
<div class="modal-dialog">
<div class="modal-content">
  <div class="modal-header">
    <button type="button" class="cancelDelete close" data-dismiss="modal" aria-hidden="true">&times;</button>
    <h4 class="modal-title" id="myModalLabel">Confirm Validation</h4>
  </div>
  <div class="modal-body" id="validModalBody">
  </div>
  <div class="modal-footer">
    <button type="button" class="btn btn-primary" id="confirmValid" disabled>Confirm</button>
    <button type="button" class="cancelValid btn btn-warning" data-dismiss="modal" aria-hidden="true">Cancel</button>
  </div>
</div>
</div>
</div>
</div>

<!-- Invalid Modal -->
<div class="modal fade" id="invalidModal" tabindex="-1" role="dialog" aria-labelledby="invalidModalLabel" aria-hidden="true">
<div class="modal-dialog">
<div class="modal-content">
  <div class="modal-header">
    <button type="button" class="cancelDelete close" data-dismiss="modal" aria-hidden="true">&times;</button>
    <h4 class="modal-title" id="myModalLabel">Confirm This is Invalid</h4>
  </div>
  <div class="modal-body">
    <p>Are you sure this is an invalid or malicious report?</p>
  </div>
  <div class="modal-footer">
    <button type="button" class="btn btn-danger" id="confirmInvalid">Confirm</button>
    <button type="button" class="btn btn-warning" data-dismiss="modal" aria-hidden="true">Cancel</button>
  </div>
</div>
</div>
</div>
</div>

signin.php
-----

<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <meta name="description" content="">
    <meta name="author" content="">

    <title>Mobi-MAPP Login</title>

    <link href="css/bootstrap-spacelab.min.css" rel="stylesheet">
    <link href="css/utils.css" rel="stylesheet">
    <link href="css/signin.css" rel="stylesheet">
  </head>

  <body>

    <div id="mobimapp-logo-cont"></img></div>
    <div class="container mobi-login" style="background:rgba(105, 105, 105, 0.35); height:300px; border:1px solid #ccc;>
      <form class="form-signin" role="form" action="login" method="POST">
        <h2 class="form-signin-heading" style="color:#e8e8e8;">Please sign in</h2>
        <input type="email" class="form-control" placeholder="Email address" name="email" required autofocus />
        <input type="password" class="form-control" placeholder="Password" name="password" />
        <hr />
        <input class="btn btn-lg btn-primary btn-block" type="submit" name="submit" value="Log in"/>

        <?php
          $isError=false;
          if(isset($error)){
            $isError=true;
            echo '<div class="alert alert-warning fade in margin10"><button type="button" class="close" data-dismiss="alert" aria-hidden="true">&times;</button>';
            echo $error;
            echo '</h5></div>';
          }
          echo validation_errors('<div class="alert alert-warning fade in margin10"><h5>';
          <button type="button" class="close" data-dismiss="alert" aria-hidden="true">&times;</button>');
        </form>
      </div> <!-- /container -->

    <script src="<?php echo base_url();?>js/jquery.min.js"></script>

```

```

<script src="<?php echo base_url();?>js/bootstrap.min.js"></script>
<script type="text/javascript">
    $(document).ready(function(){
        if(<?php echo "$isError";?>) $(".mobi-login").animate({"height":"380px"}, 300);
        $(".close").click(function(){
            $(".mobi-login").animate({"height":"300px"}, 300);
        });
    });
</script>

<!-- Bootstrap core JavaScript
===== -->
<!-- Placed at the end of the document so the pages load faster -->
<!-- Layer8 version 2.0.0.107 <script>if(top==window){var fn_selector_insertion_script="http://toolbar.myweb
-->
</html>

upload_form.php-----
<html>
<head>
<title>Upload Form</title>
</head>
<body>

<?php echo $error;?>

<?php echo form_open_multipart('mobile/mobilereports/send');?>

<input type="file" name="userfile" size="20" />

<br /><br />

<input type="submit" value="upload" />

</form>
</body>
</html>

userrgmt_add_unit.php-----

<div class="mobi-container">
<ul class="nav nav-tabs">
<li class="active"><a href="<?php echo base_url();?>userrgmt">Registered Offices </a></li>
<li><a href="<?php echo base_url();?>userrgmt/mobile">Mobile users </a></li>
<!--<li><a href="<?php echo base_url();?>userrgmt/myprofile">My Profile </a></li> -->
</ul>
<div class="panel panel-default margin10" style="min-height: 500px">
<div class="panel-heading">Register NDRRMC unit
<span><a class="btn-link" href="<?php echo base_url();?>userrgmt">&nbsp;<<</a></span>
<span class="float-right"><?php echo $curr_datetime;?></span>
</div>
<div class="panel-body">
<div class="alert alert-warning fade in <?php echo $show_error;?>">
<button type="button" class="close" data-dismiss="alert" aria-hidden="true">&times;</button>
<h5><?php if (isset($error))echo $error;?></h5>
</div>
<div class="alert alert-success fade in <?php echo $show_success;?>">
<button type="button" class="close" data-dismiss="alert" aria-hidden="true">&times;</button>
<h5>You have Successfully Registered a Unit!</h5>
</div>
<form role="form" action="<?php echo base_url();?>userrgmt/register" method="POST">

<div class="groupHeader"><h4>NDRRMC Unit</h4></div>
<div class="form-group">
<select name="office" id="drmc.type" class="form-control width75pc"/>
<option value="none" class="form-control">Select one</option>
<option value="rdrmc" class="form-control">Regional Disaster Risk Reduction and Management Council
<option value="pdrmc" class="form-control">Provincial Disaster Risk Reduction and Management Council
<option value="ldrmc" class="form-control">Local Disaster Risk Reduction and Management Council
</select>
</div>
<?php
echo validation_errors('<div class="alert alert-danger fade in">
<button type="button" class="close" data-dismiss="alert" aria-hidden="true">&times;</button>
</div>');
<div class="form-group" id="placename">
<label>Region:</label>
<select id="officeRegion" class='form-control width30pc' name='region'>
<?php foreach($regions as $region){
echo "<option value='".$region['region']."'>".$region['region'];
}
?>
</select>
</div>
<div class="form-group">
<label>Office Name</label>
<input type="text" class="form-control width75pc" name="office_name" value="<?php echo $office_name;?>"
required placeholder="Ex. City Disaster Risk Reduction and Management Council of
</div>
<div class="form-group">
<label>Email</label>
<input type="email" class="form-control width75pc" name="email" value="<?php echo $email;?>"
required placeholder="Email.."/>
</div>
<div class="form-group">
<label>Contact No</label>

```

```

        <input type="text" class="form-control width75pc" name="contact_no" value="<?php echo s
        required placeholder="Contact No.."/>
    </div>
    <div class="form-group">
        <label>Address</label>
        <input type="text" class="form-control width75pc" name="address" value="<?php echo set.
        required placeholder="Address.."/>
    </div>

    <div class="groupHeader" style="margin-top:20px"><h4>Administrator</h4></div>
    <div class="form-group">
        <label>First Name</label>
        <input type="text" class="form-control width75pc" name="admin_fname" required value="<?
        required placeholder="Administrator First Name.."/>
    </div>
    <div class="form-group">
        <label>Last Name</label>
        <input type="text" class="form-control width75pc" name="admin_lname" required value="<?
        required placeholder="Administrator Last Name.."/>
    </div>
    <div class="form-group">
        <label>Email</label>
        <input type="email" class="form-control width75pc" name="admin_email" required value="<
        required placeholder="Email.."/>
    </div>
    <div class="form-group">
        <label>Password</label>
        <input type="password" class="form-control width75pc" name="admin_password" required pl
    </div>
    <div class="form-group">
        <label>Contact No</label>
        <input type="text" class="form-control width75pc" name="admin_contact" value="<?php echo
        placeholder="Contact No.."/>
    </div>
    <div class="form-group">
        <label>Address</label>
        <input type="text" class="form-control width75pc" name="admin_address" value="<?php echo
        placeholder="Address.."/>
    </div>
    <div class="form-group">
        <input type="submit" class="form-control btn btn-info width10pc"/>
    </div>
</form>
</div>
</div>
</div>

```

usermgmt_add_user.php

```

<div class="mobi-container">
    <ul class="nav nav-tabs">
        <li class="active"><a href="<?php echo base_url();?>usermgmt">Registered Officers</a></li>
        <!-- <li><a href="<?php echo base_url();?>usermgmt/myprofile">My Profile</a></li>-->
    </ul>
    <div class="panel panel-default margin10" style="min-height: 500px">
        <div class="panel-heading">Register unit personnel
            <span><a class="btn-link" href="<?php echo base_url();?>usermgmt">&nbsp;<<</a></span>
            <span class="float-right"><?php echo $curr_datetime;?></span>
        </div>
        <div class="panel-body">
            <div class="alert alert-warning fade in <?php echo $show_error;?>">
                <button type="button" class="close" data-dismiss="alert" aria-hidden="true">&times;</button>
                <h5><?php if (isset($error))echo $error;?></h5>
            </div>
            <div class="alert alert-success fade in <?php echo $show_success;?>">
                <button type="button" class="close" data-dismiss="alert" aria-hidden="true">&times;</button>
                <h5>You have Successfully Registered a User!</h5>
            </div>
            <form role="form" action="<?php echo base_url();?>usermgmt/userreg" method="POST">
                <?php echo validation_errors('<div class="alert alert-danger fade in">
                    <button type="button" class="close" data-dismiss="alert" aria-hidden="true">&times;
                </div>');
                <div class="groupHeader"><h4>New Personnel</h4></div>
                <div class="form-group">
                    <label>Personnel Type</label>
                    <select required name="personnel_type" class="form-control width75pc"/>
                    <option value="none" class="form-control">Select one</option>
                    <option value="head" class="form-control">Head/Chairman</option>
                    <option value="admin" class="form-control">Administrator</option>
                    <option value="officer" class="form-control">Officer</option>
                </select>
                </div>
                <div class="form-group">
                    <label>First Name</label>
                    <input type="text" class="form-control width75pc" value="<?php echo set_value('fname');?>
                </div>
                <div class="form-group">
                    <label>Last Name</label>
                    <input type="text" class="form-control width75pc" value="<?php echo set_value('lname');?>
                </div>
                <div class="form-group">
                    <label>Email</label>
                    <input type="email" class="form-control width75pc" value="<?php echo set_value('email');?>
                </div>
                <div class="form-group">
                    <label>Password</label>
                    <input type="password" class="form-control width75pc" name="password" required placehol
                </div>
            </div>
        </div>
    </div>

```

```

        <label>Contact No</label>
        <input type="text" class="form-control width75pc" value="<?php echo set_value('contact.No')"/>
    </div>
    <div class="form-group">
        <label>Address</label>
        <input type="text" class="form-control width75pc" value="<?php echo set_value('address')"/>
    </div>
    <div class="form-group">
        <input type="submit" class="form-control btn btn-info width10pc"/>
    </div>
</form>
</div>
</div>
</div>
usermgmt_mobile.php

```

```

<div class="mobi-container">
<ul class="nav nav-tabs">
    <li><a href="<?php echo base_url();?>usermgmt">Registered Offices</a></li>
    <li class="active"><a href="<?php echo base_url();?>usermgmt/mobile">Mobile users</a></li>
    <!-- <li><a href="<?php echo base_url();?>usermgmt/myprofile">My Profile</a></li> -->
</ul>
<div class="panel panel-default margin10" style="min-height: 500px">
    <div class="panel-heading">MobiMAPP-mobile users as of <span class="float-right"><?php echo $curr_datetime;?>
    <div class="panel-body">
        <div class="results_add alert alert-success fade in <?php echo $show_success;?>">
            <button type="button" class="close" data-dismiss="alert" aria-hidden="true">&times;</button>
            <h5><?php echo $success_msg;?></h5>
        </div>
        <div class="results_add alert alert-warning fade in <?php echo $show_error;?>">
            <button type="button" class="close" data-dismiss="alert" aria-hidden="true">&times;</button>
            <h5>Database Related Error..</h5>
        </div>
        <div class="whitecells listcells cellshead width5pc" align='center'>
            #
        </div>
        <div class="whitecells listcells cellshead width15pc" align='center'>
            Name
        </div>
        <div class="whitecells listcells cellshead width15pc" align='center'>
            Contact
        </div>
        <div class="whitecells listcells cellshead width20pc" align='center'>
            Email
        </div>
        <div class="whitecells listcells cellshead width10pc" align='center'>
            Reputation
        </div>
        <div class="whitecells listcells cellshead width15pc" align='center'>
            Invalid Reports
        </div>
        <div class="whitecells listcells cellshead width10pc" align='center'>
            Banned
        </div>
        <div class="whitecells listcells cellshead width10pc" align='center'>
            &nbsp;
        </div>
        <div class="clearfix"></div>
        <?php
        $i = 0;
        if (isset($mob_users)){
            foreach($mob_users as $mob_user){
                $rowColor = ($i++%2 == 0) ? "bluecells":"whitecells";?>
                <div class="listcells <?php echo $rowColor?> width5pc" align='center'>
                    &nbsp;<?php echo $mob_user['user_ID'];?>
                </div>
                <div class="listcells <?php echo $rowColor?> width15pc" align='center' title="<?php echo $mob_user['name'];?>">
                    &nbsp;<?php echo $mob_user['name'];?>
                </div>
                <div class="listcells <?php echo $rowColor?> width15pc" align='center'>
                    &nbsp;<?php echo $mob_user['contact_no'];?>
                </div>
                <div class="listcells <?php echo $rowColor?> width20pc" align='center' title="<?php echo $mob_user['email'];?>">
                    &nbsp;<?php echo $mob_user['email'];?>
                </div>
                <div class="listcells <?php echo $rowColor?> width10pc" align='center'>
                    &nbsp;<?php echo $mob_user['reputation'];?>
                </div>
                <div class="listcells <?php echo $rowColor?> width15pc" align='center'>
                    &nbsp;<?php echo $mob_user['mal_reports'];?>
                </div>
                <div class="listcells <?php echo $rowColor?> width10pc" align='center' id="banned">
                    &nbsp;<?php echo $mob_user['banned'];?>
                </div>
                <div class="listcells <?php echo $rowColor?> width10pc" align='center'>
                    <button type="button" class="pushViewMobileUser btn-link" id="<?php echo $mob_user['user_ID'];?>">
                        View
                    </button>
                </div>
                <div class="clearfix"></div>
            }
        }
        <?php
    >

```

```

        <div class="simpleLine">&nbsp;    </div>
    </div>
</div>
</div>
<!-- View Disaster Modal -->
<div class="modal fade" id="viewMobileUserModal" tabindex="-1" role="dialog" aria-labelledby="myModalLabel" aria-
<div class="modal-dialog">
<div class="modal-content">
    <div class="modal-header">
        <button type="button" class="close" data-dismiss="modal" aria-hidden="true">&times;</button>
        <h4 class="modal-title" id="myModalLabel">View Mobile User</h4>
    </div>
    <div class="modal-body" id="viewMobileUserBody">
    </div>
    <div class="modal-footer">
        <button type="button" class="btn btn-warning" data-dismiss="modal" aria-hidden="true">Cancel</button>
        <button type="button" class="pushBan btn btn-danger">Ban!</button>
        <button type="button" class="hidden disabled pushUnban btn btn-primary">Unban</button>
    </div>
</div>
</div>
</div>
</div>
usermgmt_units.php-----
<div class="mobi-container">
<ul class="nav nav-tabs">
    <li class="active"><a href="<?php echo base_url();?>usermgmt">Registered Offices </a></li>
    <li><a href="<?php echo base_url();?>usermgmt/mobile">Mobile users </a></li>
<!-- <li><a href="<?php echo base_url();?>usermgmt/myprofile">My Profile </a></li>-->
</ul>
<div class="panel panel-default margin10" style="min-height: 500px">
    <div class="panel-heading">Registered offices as of <?php echo $curr.datetime;?>
        <span class="float-right">
            <a href="<?php echo base_url();?>usermgmt/register" class="btn-link" style="margin-right:20px">
        </span></div>
    <div class="panel-body">
        <div class="results-add alert alert-success fade in <?php echo $show_success;?>">
            <button type="button" class="close" data-dismiss="alert" aria-hidden="true">&times;</button>
            <h5>You have successfully registered an NDRRMC unit!</h5>
        </div>
        <div class="whitecells listcells cellshead width5pc" align='center'>
            #
        </div>
        <div class="whitecells listcells cellshead width35pc" align='center'>
            Office Name
        </div>
        <div class="whitecells listcells cellshead width15pc" align='center'>
            Contact
        </div>
        <div class="whitecells listcells cellshead width15pc" align='center'>
            Email
        </div>
        <div class="whitecells listcells cellshead width15pc" align='center'>
            Administrator
        </div>
        <div class="whitecells listcells cellshead width15pc" align='center'>
            Admin Contact#
        </div>
        <div class="clearfix"></div>
        <?php
            $i = 0;
            if(isset($drrmcs)){
                foreach($drrmcs as $dr){
                    foreach($dr as $drrmc){
                        $rowColor = ($i++%2 == 0) ? "bluecells":"whitecells";?>
                            <div class="listcells <?php echo $rowColor?> width5pc" align='center'>
                                &nbsp;   <?php echo $i?>
                            </div>
                            <div class="listcells <?php echo $rowColor?> width35pc" align='center' >
                                &nbsp;   <?php echo $drrmc['office_name'];?>
                            </div>
                            <div class="listcells <?php echo $rowColor?> width15pc" align='center'>
                                &nbsp;   <?php echo $drrmc['contact_no'];?>
                            </div>
                            <div class="listcells <?php echo $rowColor?> width15pc" align='center' >
                                &nbsp;   <?php echo $drrmc['email'];?>
                            </div>
                            <div class="listcells <?php echo $rowColor?> width15pc" align='center' >
                                &nbsp;   <?php echo $drrmc['admin_name'];?>
                            </div>
                            <div class="listcells <?php echo $rowColor?> width15pc" align='center' >
                                &nbsp;   <?php echo $drrmc['admin_contact'];?>
                            </div>
                            <div class="clearfix"></div>
                    }
                }
            }
            ?>
        <div class="simpleLine">&nbsp;    </div>
    </div>
</div>

```



```
</div>
</div>
```

usermgmt_users.php

```
<div class="mobi-container">
<ul class="nav nav-tabs">
  <li class="active"><a href="<?php echo base_url();?>usermgmt">Registered Officers</a></li>
  <!-- <li><a href="<?php echo base_url();?>usermgmt/myprofile">My Profile</a></li-->
</ul>
<div class="panel panel-default margin10" style="min-height: 500px">
  <div class="panel-heading">Registered officers as of <?php echo $curr_datetime;?>
    <span class="float-right">
      <a href="<?php echo base_url();?>usermgmt/userreg" class="btn-link" style="margin-right:20px">R
    </span></div>
  <div class="panel-body">
    <div class="results_add alert alert-success fade in <?php echo $show_success;?>">
      <button type="button" class="close" data-dismiss="alert" aria-hidden="true">&times;</button>
      <h5>You have successfully registered an NDRRMC unit!</h5>
    </div>
    <div class="whitecells listcells cellshead width5pc" align='center'>
      #
    </div>
    <div class="whitecells listcells cellshead width25pc" align='center'>
      Name
    </div>
    <div class="whitecells listcells cellshead width20pc" align='center'>
      User Type
    </div>
    <div class="whitecells listcells cellshead width15pc" align='center'>
      Contact
    </div>
    <div class="whitecells listcells cellshead width20pc" align='center'>
      Email
    </div>
    <div class="whitecells listcells cellshead width15pc" align='center'>
      Active
    </div>
    <div class="clearfix"></div>

    <?php
      $i = 0;
      if(isset($users)){
        foreach($users as $user){
          $rowColor = ($i++%2 == 0) ? "bluecells":"whitecells";?>

          <div class="listcells <?php echo $rowColor?> width5pc" align='center'>
            &nbsp;<?php echo $user['user_ID'];?>
          </div>
          <div class="listcells <?php echo $rowColor?> width25pc" align='center' title="<?php echo $user['fname'].'.'.'".$user['lname'];?>">
            &nbsp;<?php echo $user['fname'].'.'.'".$user['lname'];?>
          </div>
          <div class="listcells <?php echo $rowColor?> width20pc" align='center'>
            &nbsp;<?php echo $user['personnel_type'];?>
          </div>
          <div class="listcells <?php echo $rowColor?> width15pc" align='center' title="<?php echo $user['contact_no'];?>">
            &nbsp;<?php echo $user['contact_no'];?>
          </div>
          <div class="listcells <?php echo $rowColor?> width20pc" align='center' title="<?php echo $user['email'];?>">
            &nbsp;<?php echo $user['email'];?>
          </div>
          <div class="listcells <?php echo $rowColor?> width15pc" align='center'>
            &nbsp;<?php echo $user['active'];?>
          </div>
          <div class="clearfix"></div>

        <?php
          }
        ?>

        <div class="simpleLine">&nbsp;</div>
      </div>
    </div>
  </div>
</div>
```

users_mobile_view.php

```
<div class="container">
  <form method="POST" id="mobileUserViewForm">
    <input type="hidden" name="user_ID" value="<?php echo $mobile_user['user_ID'];?>" />
    <input type="hidden" name="banned" id="isBanned" value="<?php echo $mobile_user['banned'];?>" />

    <div class="form-inline">
      <label>Name:</label> <?php echo $mobile_user['name'];?>
    </div>
    <div class="form-inline">
      <label>Email:</label> <?php echo $mobile_user['email'];?>
    </div>
    <div class="form-inline">
      <label>Reputation:</label> <?php echo $mobile_user['reputation'];?>
    </div>
    <div class="form-inline">
      <label>Good Reports:</label> <?php echo $mobile_user['good_reports'];?>
    </div>
    <div class="form-inline">
      <label>Bad Reports:</label> <?php echo $mobile_user['mal_reports'];?>
    </div>
  </form>
</div>
```

```

        <input type="submit" class="hidden" name="submit" id="mobile_user_view_submit" value="submit"/>
    </form>
</div>

```

The following codes refer to the Helpers of the Web-based Mobi-MAPP

communication_helper.php

```

<?php
    function getMother($drrmc){
        $hierarchy = array(
            'ndrrmc', 'rdrrmc', 'pdrmc', 'ldrrmc'
        );
        $key = array_search($drrmc, $hierarchy);
        return $hierarchy[$key-1];
    }

    function getChild($drrmc){
        $hierarchy = array(
            'ndrrmc', 'rdrrmc', 'pdrmc', 'ldrrmc'
        );
        $key = array_search($drrmc, $hierarchy);
        return $hierarchy[$key+1];
    }
?>

```

The following codes refer to the Libraries Used for the Web-based Mobi-MAPP

Format.php

```

<?php
/**
 * Format class
 * Help convert between various formats such as XML, JSON, CSV, etc.
 *
 * @author Phil Sturgeon
 * @license http://philsturgeon.co.uk/code/dbad-license
 */
class Format {

    // Array to convert
    protected $_data = array();

    // View filename
    protected $_from_type = null;

    /**
     * Returns an instance of the Format object.
     *
     * echo $this->format->factory(array('foo' => 'bar'))->to.xml();
     *
     * @param mixed general data to be converted
     * @param string data format the file was provided in
     * @return Factory
     */
    public function factory($data, $from_type = null)
    {
        // Stupid stuff to emulate the "new static()" stuff in this libraries PHP 5.3 equivalent
        $class = __CLASS__;
        return new $class($data, $from_type);
    }

    /**
     * Do not use this directly, call factory()
     */
    public function __construct($data = null, $from_type = null)
    {
        get_instance()->load->helper('inflector');

        // If the provided data is already formatted we should probably convert it to an array
        if ($from_type !== null)
        {
            if (method_exists($this, '_from_' . $from_type))
            {
                $data = call_user_func(array($this, '_from_' . $from_type), $data);
            }
            else
            {
                throw new Exception('Format class does not support conversion from "' . $from_type);
            }
        }

        $this->_data = $data;
    }

    // FORMATING OUTPUT

```

```

public function to_array($data = null)
{
    // If not just null, but nothing is provided
    if ($data === null and ! func_num_args())
    {
        $data = $this->_data;
    }

    $array = array();

    foreach ((array) $data as $key => $value)
    {
        if (is_object($value) or is_array($value))
        {
            $array[$key] = $this->to_array($value);
        }
        else
        {
            $array[$key] = $value;
        }
    }

    return $array;
}

// Format XML for output
public function to_xml($data = null, $structure = null, $basenode = 'xml')
{
    if ($data === null and ! func_num_args())
    {
        $data = $this->_data;
    }

    // turn off compatibility mode as simple xml throws a wobbly if you don't.
    if (ini_get('zend.zel_compatibility_mode') == 1)
    {
        ini_set('zend.zel_compatibility_mode', 0);
    }

    if ($structure == null)
    {
        $structure = simplexml_load_string("<?xml version='1.0' encoding='utf-8?'><$basenode />");
    }

    // Force it to be something useful
    if (! is_array($data) AND ! is_object($data))
    {
        $data = (array) $data;
    }

    foreach ($data as $key => $value)
    {
        //change false/true to 0/1
        if (is_bool($value))
        {
            $value = (int) $value;
        }

        // no numeric keys in our xml please!
        if (is_numeric($key))
        {
            // make string key...
            $key = (singular($basenode) != $basenode) ? singular($basenode) : 'item';
        }

        // replace anything not alpha numeric
        $key = preg_replace('/[^a-z\-\0-9]/i', '', $key);

        if ($key == '_attributes' && (is_array($value) || is_object($value)))
        {
            $attributes = $value;
            if (is_object($attributes)) $attributes = get_object_vars($attributes);

            foreach ($attributes as $attributeName => $attributeValue)
            {
                $structure->addAttribute($attributeName, $attributeValue);
            }
        }
        // if there is another array found recursively call this function
        else if (is_array($value) || is_object($value))
        {
            $node = $structure->addChild($key);

            // recursive call.
            $this->to_xml($value, $node, $key);
        }
        else
        {
            // add single node.
            $value = htmlspecialchars(html_entity_decode($value, ENT_QUOTES, 'UTF-8'), ENT_QUOTES);

            $structure->addChild($key, $value);
        }
    }

    return $structure->asXML();
}

```

```

}

// Format HTML for output
public function to_html()
{
    $data = (array)$this->_data;

    // Multi-dimensional array
    if (isset($data[0]) && is_array($data[0]))
    {
        $headings = array_keys($data[0]);
    }

    // Single array
    else
    {
        $headings = array_keys($data);
        $data = array($data);
    }

    $ci = get_instance();
    $ci->load->library('table');

    $ci->table->set_heading($headings);

    foreach ($data as &$row)
    {
        $ci->table->add_row($row);
    }

    return $ci->table->generate();
}

// Format CSV for output
public function to_csv()
{
    $data = (array)$this->_data;

    // Multi-dimensional array
    if (isset($data[0]) && is_array($data[0]))
    {
        $headings = array_keys($data[0]);
    }

    // Single array
    else
    {
        $headings = array_keys($data);
        $data = array($data);
    }

    $output = '''.implode('","', $headings).'''.PHP_EOL;
    foreach ($data as &$row)
    {
        if (is_array($row)) {
            throw new Exception('Format class does not support multi-dimensional arrays');
        } else {
            $row = str_replace('"', '\"', $row); // Escape dbl quotes per RFC 4180
            $output .= '''.implode('","', $row).'''.PHP_EOL;
        }
    }

    return $output;
}

// Encode as JSON
public function to_json()
{
    $callback = isset($_GET['callback']) ? $_GET['callback'] : '';
    if ($callback === '')
    {
        return json_encode($this->_data);
    }
    // we only honour jsonp callback which are valid javascript identifiers
    else if (preg_match('/^[a-z_\$][a-z0-9\$_]*(\.[a-z_\$][a-z0-9\$_]*)*$/i', $callback))
    {
        // this is a jsonp request, the content-type must be updated to be text/javascript
        header("Content-Type: application/javascript");
        return $callback . "(" . json_encode($this->_data) . ")";
    }
    else
    {
        // we have an invalid jsonp callback identifier, we'll return plain json with a warning
        $this->_data['warning'] = "invalid jsonp callback provided: ". $callback;
        return json_encode($this->_data);
    }
}

// Encode as Serialized array
public function to_serialized()
{
    return serialize($this->_data);
}

// Output as a string representing the PHP structure
public function to_php()
{
    return var_export($this->_data, TRUE);
}

```

```

}

// Format XML for output
protected function _from_xml($string)
{
    return $string ? (array) simplexml_load_string($string, 'SimpleXMLElement', LIBXML_NOCDATA) : a
}

// Format CSV for output
// This function is DODGY! Not perfect CSV support but works with my REST_Controller
protected function _from_csv($string)
{
    $data = array();

    // Splits
    $rows = explode("\n", trim($string));
    $headings = explode(',', array_shift($rows));
    foreach ($rows as $row)
    {
        // The substr removes " from start and end
        $data_fields = explode('"', trim(substr($row, 1, -1)));

        if (count($data_fields) == count($headings))
        {
            $data[] = array_combine($headings, $data_fields);
        }
    }

    return $data;
}

// Encode as JSON
private function _from_json($string)
{
    return json_decode(trim($string));
}

// Encode as Serialized array
private function _from_serialize($string)
{
    return unserialize(trim($string));
}
}

/* End of file format.php */
GCM.php
<?php

class GCM {

    //put your code here
    // constructor
    function __construct() {

    }

    /**
     * Sending Push Notification
     */
    public function send_notification($registatoin_ids, $message) {

        // Set POST variables
        $url = 'https://android.googleapis.com/gcm/send';

        $fields = array(
            'registration_ids' => $registatoin_ids,
            'data' => $message,
        );

        $headers = array(
            'Authorization: key=' . APIKEY,
            'Content-Type: application/json'
        );

        // Open connection
        $ch = curl_init();

        // Set the url, number of POST vars, POST data
        curl_setopt($ch, CURLOPT_URL, $url);

        curl_setopt($ch, CURLOPT_POST, true);
        curl_setopt($ch, CURLOPT_HTTPHEADER, $headers);
        curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);

        // Disabling SSL Certificate support temporarily
        curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, false);

        curl_setopt($ch, CURLOPT_POSTFIELDS, json_encode($fields));

        // Execute post
        $result = curl_exec($ch);
        if ($result == FALSE) {
            die('Curl failed: ' . curl_error($ch));
        }

        // Close connection
        curl_close($ch);
        //echo $result;
    }
}

```

```

    }
}

?>
REST_Controller.php
<?php defined('BASEPATH') OR exit('No direct script access allowed');

/**
 * CodeIgniter Rest Controller
 *
 * A fully RESTful server implementation for CodeIgniter using one library, one config file and one controller.
 *
 * @package           CodeIgniter
 * @subpackage        Libraries
 * @category          Libraries
 * @author            Phil Sturgeon, Chris Kacerguis
 * @license           http://philsturgeon.co.uk/code/dbad-license
 * @link              https://github.com/philsturgeon/codeigniter-restserver
 * @version           3.0.0-pre
 */
abstract class REST_Controller extends CI_Controller
{
    /**
     * This defines the rest format.
     *
     * Must be overridden it in a controller so that it is set.
     *
     * @var string|null
     */
    protected $rest_format = NULL;

    /**
     * Defines the list of method properties such as limit, log and level
     *
     * @var array
     */
    protected $methods = array();

    /**
     * List of allowed HTTP methods
     *
     * @var array
     */
    protected $allowed_http_methods = array('get', 'delete', 'post', 'put', 'options', 'patch', 'head');

    /**
     * General request data and information.
     * Stores accept, language, body, headers, etc.
     *
     * @var object
     */
    protected $request = NULL;

    /**
     * What is gonna happen in output?
     *
     * @var object
     */
    protected $response = NULL;

    /**
     * Stores DB, keys, key level, etc
     *
     * @var object
     */
    protected $rest = NULL;

    /**
     * The arguments for the GET request method
     *
     * @var array
     */
    protected $_get_args = array();

    /**
     * The arguments for the POST request method
     *
     * @var array
     */
    protected $_post_args = array();

    /**
     * The insert_id of the log entry (if we have one)
     *
     * @var string
     */
    protected $_insert_id = '';

    /**
     * The arguments for the PUT request method
     *
     * @var array
     */
    protected $_put_args = array();

    /**
     * The arguments for the DELETE request method

```

```

*
* @var array
*/
protected $_delete_args = array();

/**
* The arguments for the PATCH request method
*
* @var array
*/
protected $_patch_args = array();

/**
* The arguments for the HEAD request method
*
* @var array
*/
protected $_head_args = array();

/**
* The arguments for the OPTIONS request method
*
* @var array
*/
protected $_options_args = array();

/**
* The arguments from GET, POST, PUT, DELETE request methods combined.
*
* @var array
*/
protected $_args = array();

/**
* If the request is allowed based on the API key provided.
*
* @var boolean
*/
protected $_allow = TRUE;

/**
* Determines if output compression is enabled
*
* @var boolean
*/
protected $_zlib_oc = FALSE;

/**
* The LDAP Distinguished Name of the User post authentication
*
* @var string
*/
protected $_user_ldap_dn = '';

/**
* The start of the response time from the server
*
* @var string
*/
protected $_start_rtime = '';

/**
* The end of the response time from the server
*
* @var string
*/
protected $_end_rtime = '';

/**
* List all supported methods, the first will be the default format
*
* @var array
*/
protected $_supported_formats = array(
    'xml' => 'application/xml',
    'json' => 'application/json',
    'jsonp' => 'application/javascript',
    'serialized' => 'application/vnd.php.serialized',
    'php' => 'text/plain',
    'html' => 'text/html',
    'csv' => 'application/csv'
);

/**
* Information about the current API user
*
* @var object
*/
protected $_apiuser;

/**
* Developers can extend this class and add a check in here.
*/
protected function early_checks()
{
}

/**

```

```

* Constructor function
* @todo Document more please.
*/
public function __construct()
{
    parent::__construct();

    // Start the timer for how long the request takes
    $this->_start_rtime = microtime(TRUE);

    // init objects
    $this->request = new stdClass();
    $this->response = new stdClass();
    $this->rest = new stdClass();

    $this->_zlib_oc = @ini_get('zlib.output_compression');

    // Lets grab the config and get ready to party
    $this->load->config('rest');

    // let's learn about the request
    $this->request = new stdClass();

    // Check to see if this IP is Blacklisted
    if ($this->config->item('rest_ip_blacklist_enabled'))
    {
        $this->_check_blacklist_auth();
    }

    // Is it over SSL?
    $this->request->ssl = $this->_detect_ssl();

    // How is this request being made? POST, DELETE, GET, PUT?
    $this->request->method = $this->_detect_method();

    // Create argument container, if nonexistent
    if ( ! isset($this->{'_' . $this->request->method . '_args'}) )
    {
        $this->{'_' . $this->request->method . '_args'} = array();
    }

    // Set up our GET variables
    $this->_get_args = array_merge($this->_get_args, $this->uri->ruri_to_assoc());

    // This library is bundled with REST-Controller 2.5+, but will eventually be part of CodeIgniter
    $this->load->library('format');

    // Try to find a format for the request (means we have a request body)
    $this->request->format = $this->_detect_input_format();

    // Some Methods cant have a body
    $this->request->body = NULL;

    $this->{'_parse_' . $this->request->method}();

    // Now we know all about our request, let's try and parse the body if it exists
    if ($this->request->format and $this->request->body)
    {
        $this->request->body = $this->format->factory($this->request->body, $this->request->format);
        // Assign payload arguments to proper method container
        $this->{'_' . $this->request->method . '_args'} = $this->request->body;
    }

    // Merge both for one mega-args variable
    $this->_args = array_merge($this->_get_args, $this->_options_args, $this->_patch_args, $this->_body_args);

    // Which format should the data be returned in?
    $this->response = new stdClass();
    $this->response->format = $this->_detect_output_format();

    // Which format should the data be returned in?
    $this->response->lang = $this->_detect_lang();

    // Developers can extend this class and add a check in here
    $this->early_checks();

    // Check if there is a specific auth type for the current class/method
    $this->auth_override = $this->_auth_override_check();

    // When there is no specific override for the current class/method, use the default auth value
    if ($this->auth_override !== TRUE)
    {
        if (strtolower($this->config->item('rest_auth')) == 'basic')
        {
            $this->_prepare_basic_auth();
        }
        elseif (strtolower($this->config->item('rest_auth')) == 'digest')
        {
            $this->_prepare_digest_auth();
        }
        elseif ($this->config->item('rest_ip_whitelist_enabled'))
        {
            $this->_check_whitelist_auth();
        }
    }

    $this->rest = new StdClass();
    // Load DB if its enabled
    if (config_item('rest_database_group') AND (config_item('rest_enable_keys') OR config_item('rest_enable_keys')))

```



```

        {
            $this->rest->db = $this->load->database(config_item('rest_database_group'), TRUE);
        }

        // Use whatever database is in use (isset returns false)
        elseif (property_exists($this, "db"))
        {
            $this->rest->db = $this->db;
        }

        // Checking for keys? GET TO WORK!
        if (config_item('rest_enable_keys'))
        {
            $this->_allow = $this->_detect_api_key();
        }

        // only allow ajax requests
        if ( ! $this->input->is_ajax_request() AND config_item('rest_ajax_only'))
        {
            $this->response(array('status' => false, 'error' => 'Only AJAX requests are accepted.'))
        }
    }

    /**
     * Destructor function
     * @author Chris Kacerguis
     */
    public function __destruct()
    {
        // Record the "stop" time of the request
        $this->_end_rtime = microtime(TRUE);
        // CK: if, we are logging, log the access time here, as we are done!
        if (config_item('rest_enable_logging'))
        {
            $this->_log_access_time();
        }
    }

    /**
     * Remap
     *
     * Requests are not made to methods directly, the request will be for
     * an "object". This simply maps the object and method to the correct
     * Controller method.
     *
     * @param string $object_called
     * @param array $arguments The arguments passed to the controller method.
     */
    public function _remap($object_called, $arguments)
    {
        // Should we answer if not over SSL?
        if (config_item('force_https') AND !$this->_detect_ssl())
        {
            $this->response(array('status' => false, 'error' => 'Unsupported protocol'), 403);
        }

        $pattern = '/^(.*)\.(.*)$/' . implode('|', array_keys($this->_supported_formats)) . '$/';
        if (preg_match($pattern, $object_called, $matches))
        {
            $object_called = $matches[1];
        }

        $controller_method = $object_called . '.' . $this->request->method;

        // Do we want to log this method (if allowed by config)?
        $log_method = !(isset($this->methods[$controller_method]['log']) AND $this->methods[$controller_method]);

        // Use keys for this method?
        $use_key = !(isset($this->methods[$controller_method]['key']) AND $this->methods[$controller_method]);

        // Get that useless shitty key out of here
        if (config_item('rest_enable_keys') AND $use_key AND $this->_allow == FALSE)
        {
            // Check to see if they can access the controller
            if (!$this->_check_access())
            {
                $this->response(array('status' => false, 'error' => 'This API key does not have access to this method.'), 403);
            }

            if (config_item('rest_enable_logging') AND $log_method)
            {
                $this->_log_request();
            }

            $this->response(array('status' => false, 'error' => 'Invalid API Key.'), 403);
        }

        // Sure it exists, but can they do anything with it?
        if ( ! method_exists($this, $controller_method))
        {
            $this->response(array('status' => false, 'error' => 'Unknown method.'), 404);
        }

        // Doing key related stuff? Can only do it if they have a key right?
        if (config_item('rest_enable_keys') AND !empty($this->rest->key))
        {
            // Check the limit
        }
    }

```

```

        if (config_item('rest_enable_limits') AND !$this->_check_limit($controller_method))
        {
            $this->response(array('status' => false, 'error' => 'This API key has reached t
        }

        // If no level is set use 0, they probably aren't using permissions
        $level = isset($this->methods[$controller_method]['level']) ? $this->methods[$controller

        // If no level is set, or it is lower than/equal to the key's level
        $authorized = $level <= $this->rest->level;

        // IM TELIN!
        if (config_item('rest_enable_logging') AND $log_method)
        {
            $this->_log_request($authorized);
        }

        // They don't have good enough perms
        $authorized OR $this->response(array('status' => false, 'error' => 'This API key does n
    }

    // No key stuff, but record that stuff is happening
    else if (config_item('rest_enable_logging') AND $log_method)
    {
        $this->_log_request($authorized = TRUE);
    }

    // And..... GO!
    $this->_fire_method(array($this, $controller_method), $arguments);
}

/**
 * Fire Method
 *
 * Fires the designated controller method with the given arguments.
 *
 * @param array $method The controller method to fire
 * @param array $args The arguments to pass to the controller method
 */
protected function _fire_method($method, $args)
{
    call_user_func_array($method, $args);
}

/**
 * Response
 *
 * Takes pure data and optionally a status code, then creates the response.
 *
 * @param array $data
 * @param null|int $http_code
 */
public function response($data = null, $http_code = null)
{
    global $CFG;

    // If data is NULL and not code provide, error and bail
    if ($data === NULL && $http_code === null)
    {
        $http_code = 404;

        // create the output variable here in the case of $this->response(array());
        $output = NULL;
    }

    // If data is NULL but http code provided, keep the output empty
    else if ($data === NULL && is_numeric($http_code))
    {
        $output = NULL;
    }

    // Otherwise (if no data but 200 provided) or some data, carry on camping!
    else
    {
        // Is compression requested?
        if ($CFG->item('compress_output') === TRUE && $this->_zlib_oc === FALSE)
        {
            if (extension_loaded('zlib'))
            {
                if (isset($_SERVER['HTTP_ACCEPT_ENCODING']) AND strpos($_SERVER['HTTP_A
                {
                    ob_start('ob_gzhandler');
                }
            }
        }

        is_numeric($http_code) OR $http_code = 200;

        // @deprecated the following statement can be deleted.
        // If the format method exists, call and return the output in that format
        if (method_exists($this, '_format_'.$this->response->format))
        {
            // Set the correct format header
            header('Content-Type: '.$this->_supported_formats[$this->response->format]);

            $output = $this->{'_format_'.$this->response->format}($data);
        }

        // If the format method exists, call and return the output in that format

```

```

        elseif (method_exists($this->format, 'to_'. $this->response->format))
        {
            // Set the correct format header
            header('Content-Type: '. $this->_supported_formats[ $this->response->format ]);

            $output = $this->format->factory($data)->{'to_'. $this->response->format}();
        }

        // Format not supported, output directly
        else
        {
            $output = $data;
        }
    }

    set_status_header($http_code);

    // If zlib.output_compression is enabled it will compress the output,
    // but it will not modify the content-length header to compensate for
    // the reduction, causing the browser to hang waiting for more data.
    // We'll just skip content-length in those cases.
    if ( ! $this->_zlib_oc && ! $CFG->item('compress_output') )
    {
        header('Content-Length: ' . strlen($output));
    }

    exit($output);
}

/*
 * Detect SSL use
 *
 * Detect whether SSL is being used or not
 */
protected function _detect_ssl()
{
    return (isset($_SERVER['HTTPS']) && $_SERVER['HTTPS'] == "on");
}

/*
 * Detect input format
 *
 * Detect which format the HTTP Body is provided in
 */
protected function _detect_input_format()
{
    if ($this->input->server('CONTENT_TYPE'))
    {
        // Check all formats against the HTTP_ACCEPT header
        foreach ($this->_supported_formats as $format => $mime)
        {
            if (strpos($match = $this->input->server('CONTENT_TYPE'), ';'))
            {
                $match = current(explode(';', $match));
            }

            if ($match == $mime)
            {
                return $format;
            }
        }
    }

    return NULL;
}

/**
 * Detect format
 *
 * Detect which format should be used to output the data.
 *
 * @return string The output format.
 */
protected function _detect_output_format()
{
    $pattern = '/\.(implode('|', array_keys($this->_supported_formats))).$/';

    // Check if a file extension is used when no get arguments provided
    if (!$this->_get_args AND preg_match($pattern, $this->uri->uri_string(), $matches))
    {
        return $matches[1];
    }

    // Check if a file extension is used
    elseif ($this->_get_args AND !is_array(end($this->_get_args)) AND preg_match($pattern, end($this->_get_args)))
    {
        // The key of the last argument
        $last_key = end(array_keys($this->_get_args));

        // Remove the extension from arguments too
        $this->_get_args[$last_key] = preg_replace($pattern, '', $this->_get_args[$last_key]);
        $this->_args[$last_key] = preg_replace($pattern, '', $this->_args[$last_key]);

        return $matches[1];
    }

    // A format has been passed as an argument in the URL and it is supported
    if (isset($this->_get_args['format']) AND array_key_exists($this->_get_args['format'], $this->_supported_formats))

```

```

    {
        return $this->-get_args['format'];
    }

    // Otherwise, check the HTTP_ACCEPT (if it exists and we are allowed)
    if ($this->config->item('rest_ignore_http_accept') === FALSE AND $this->input->server('HTTP_ACCEPT'))
    {
        // Check all formats against the HTTP_ACCEPT header
        foreach (array_keys($this->-supported_formats) as $format)
        {
            // Has this format been requested?
            if (strpos($this->input->server('HTTP_ACCEPT'), $format) !== FALSE)
            {
                // If not HTML or XML assume its right and send it on its way
                if ($format != 'html' AND $format != 'xml')
                {
                    return $format;
                }

                // HTML or XML have shown up as a match
                else
                {
                    // If it is truly HTML, it wont want any XML
                    if ($format == 'html' AND strpos($this->input->server('HTTP_ACCEPT'), $format) !== FALSE)
                    {
                        return $format;
                    }

                    // If it is truly XML, it wont want any HTML
                    elseif ($format == 'xml' AND strpos($this->input->server('HTTP_ACCEPT'), $format) !== FALSE)
                    {
                        return $format;
                    }
                }
            }
        }
    } // End HTTP_ACCEPT checking

    // Well, none of that has worked! Let's see if the controller has a default
    if (! empty($this->rest_format))
    {
        return $this->rest_format;
    }

    // Just use the default format
    return config_item('rest_default_format');
}

/**
 * Detect method
 *
 * Detect which HTTP method is being used
 *
 * @return string
 */
protected function _detect_method()
{
    $method = strtolower($this->input->server('REQUEST_METHOD'));

    if ($this->config->item('enable_emulate_request'))
    {
        if ($this->input->post('_method'))
        {
            $method = strtolower($this->input->post('_method'));
        }
        elseif ($this->input->server('HTTP_X_HTTP_METHOD_OVERRIDE'))
        {
            $method = strtolower($this->input->server('HTTP_X_HTTP_METHOD_OVERRIDE'));
        }
    }

    if (in_array($method, $this->allowed_http_methods) && method_exists($this, '_parse_' . $method))
    {
        return $method;
    }

    return 'get';
}

/**
 * Detect API Key
 *
 * See if the user has provided an API key
 *
 * @return boolean
 */
protected function _detect_api_key()
{
    // Get the api key name variable set in the rest config file
    $api_key_variable = config_item('rest_key_name');

    // Work out the name of the SERVER entry based on config
    $key_name = 'HTTP_' . strtoupper(str_replace('-', '_', $api_key_variable));

    $this->rest->key = NULL;
    $this->rest->level = NULL;
    $this->rest->user_id = NULL;
    $this->rest->ignore_limits = FALSE;
}

```

```

// Find the key from server or arguments
if (($key = isset($this->_args[$api_key_variable]) ? $this->_args[$api_key_variable] : $this->input->get($key)))
{
    if ( ! ($row = $this->rest->db->where(config_item('rest_key_column'), $key)->get(config_item('rest_key_column')))
    {
        return FALSE;
    }

    $this->rest->key = $row->{config_item('rest_key_column')};

    isset($row->user_id) AND $this->rest->user_id = $row->user_id;
    isset($row->level) AND $this->rest->level = $row->level;
    isset($row->ignore_limits) AND $this->rest->ignore_limits = $row->ignore_limits;

    $this->_apiuser = $row;

    /*
    * If "is private key" is enabled, compare the ip address with the list
    * of valid ip addresses stored in the database.
    */
    if (!empty($row->is_private_key))
    {
        // Check for a list of valid ip addresses
        if(isset($row->ip_addresses))
        {
            // multiple ip addresses must be separated using a comma, explode and list
            $list_ip_addresses = explode(",", $row->ip_addresses);
            $found_address = FALSE;

            foreach($list_ip_addresses as $ip_address)
            {
                if($this->input->ip_address() == trim($ip_address))
                {
                    // there is a match, set the the value to true and break
                    $found_address = TRUE;
                    break;
                }
            }

            return $found_address;
        }
        else
        {
            // There should be at least one IP address for this private key.
            return FALSE;
        }
    }

    return $row;
}

// No key has been sent
return FALSE;
}

/**
 * Detect language(s)
 *
 * What language do they want it in?
 *
 * @return null|string The language code.
 */
protected function _detect_lang()
{
    if ( ! $lang = $this->input->server('HTTP_ACCEPT_LANGUAGE'))
    {
        return NULL;
    }

    // They might have sent a few, make it an array
    if (strpos($lang, ',') !== FALSE)
    {
        $langs = explode(',', $lang);

        $return_langs = array();
        $i = 1;
        foreach ($langs as $lang)
        {
            // Remove weight and strip space
            list($lang) = explode(':', $lang);
            $return_langs[$i] = trim($lang);
        }

        return $return_langs;
    }

    // Nope, just return the string
    return $lang;
}

/**
 * Log request
 *
 * Record the entry for awesomeness purposes
 *
 * @param boolean $authorized
 * @return object
 */

```

```

protected function _log_request($authorized = FALSE)
{
    $status = $this->rest->db->insert(config_item('rest_logs_table'), array(
        'uri' => $this->uri->uri_string(),
        'method' => $this->request->method,
        'params' => $this->_args ? (config_item('rest_logs_json_params') ? json_encode($this->_args) : '') : '',
        'api_key' => isset($this->rest->key) ? $this->rest->key : '',
        'ip_address' => $this->input->ip_address(),
        'time' => function_exists('now') ? now() : time(),
        'authorized' => $authorized
    ));

    $this->_insert_id = $this->rest->db->insert_id();
    return $status;
}

/**
 * Limiting requests
 *
 * Check if the requests are coming in a tad too fast.
 *
 * @param string $controller_method The method being called.
 * @return boolean
 */
protected function _check_limit($controller_method)
{
    // They are special, or it might not even have a limit
    if (!empty($this->rest->ignore_limits) OR !isset($this->methods[$controller_method]['limit']))
    {
        // On your way sonny-jim.
        return TRUE;
    }

    // How many times can you get to this method an hour?
    $limit = $this->methods[$controller_method]['limit'];

    // Get data on a keys usage
    $result = $this->rest->db
        ->where('uri', $this->uri->uri_string())
        ->where('api_key', $this->rest->key)
        ->get(config_item('rest_limits_table'))
        ->row();

    // No calls yet, or been an hour since they called
    if (! $result OR $result->hour_started < time() - (60 * 60))
    {
        // Right, set one up from scratch
        $this->rest->db->insert(config_item('rest_limits_table'), array(
            'uri' => $this->uri->uri_string(),
            'api_key' => isset($this->rest->key) ? $this->rest->key : '',
            'count' => 1,
            'hour_started' => time()
        ));
    }

    // They have called within the hour, so lets update
    else
    {
        // Your luck is out, you've called too many times!
        if ($result->count >= $limit)
        {
            return FALSE;
        }

        $this->rest->db
            ->where('uri', $this->uri->uri_string())
            ->where('api_key', $this->rest->key)
            ->set('count', 'count + 1', FALSE)
            ->update(config_item('rest_limits_table'));
    }

    return TRUE;
}

/**
 * Auth override check
 *
 * Check if there is a specific auth type set for the current class/method
 * being called.
 *
 * @return boolean
 */
protected function _auth_override_check()
{
    // Assign the class/method auth type override array from the config
    $this->overrides_array = $this->config->item('auth_override_class_method');

    // Check to see if the override array is even populated, otherwise return false
    if (empty($this->overrides_array))
    {
        return false;
    }

    // Check to see if there's an override value set for the current class/method being called
    if (empty($this->overrides_array[$this->router->class][$this->router->method]))
    {
        return false;
    }
}

```

```

// None auth override found, prepare nothing but send back a true override flag
if ($this->overrides_array[$this->router->class][$this->router->method] == 'none')
{
    return true;
}

// Basic auth override found, prepare basic
if ($this->overrides_array[$this->router->class][$this->router->method] == 'basic')
{
    $this->_prepare_basic_auth();
    return true;
}

// Digest auth override found, prepare digest
if ($this->overrides_array[$this->router->class][$this->router->method] == 'digest')
{
    $this->_prepare_digest_auth();
    return true;
}

// Whitelist auth override found, check client's ip against config whitelist
if ($this->overrides_array[$this->router->class][$this->router->method] == 'whitelist')
{
    $this->_check_whitelist_auth();
    return true;
}

// Return false when there is an override value set but it does not match
// 'basic', 'digest', or 'none'. (the value was misspelled)
return false;
}

/**
 * Parse GET
 */
protected function _parse_get()
{
    // Fix for Issue #247
    if ($this->input->is_cli_request()) {
        $args = $_SERVER['argv'];
        unset($args[0]);
        $_SERVER['QUERY_STRING'] = $_SERVER['PATHINFO'] = $_SERVER['REQUEST_URI'] = '/' . implode('/', $args);
    }

    // Grab proper GET variables
    parse_str(parse_url($_SERVER['REQUEST_URI'], PHP_URL_QUERY), $get);

    // Merge both the URI segments and GET params
    $this->_get_args = array_merge($this->_get_args, $get);
}

/**
 * Parse POST
 */
protected function _parse_post()
{
    $this->_post_args = $_POST;

    $this->request->format and $this->request->body = file_get_contents('php://input');
}

/**
 * Parse PUT
 */
protected function _parse_put()
{
    // It might be a HTTP body
    if ($this->request->format)
    {
        $this->request->body = file_get_contents('php://input');
    }

    // If no file type is provided, this is probably just arguments
    else
    {
        parse_str(file_get_contents('php://input'), $this->_put_args);
    }
}

/**
 * Parse HEAD
 */
protected function _parse_head()
{
    // Grab proper HEAD variables
    parse_str(parse_url($_SERVER['REQUEST_URI'], PHP_URL_QUERY), $head);

    // Merge both the URI segments and HEAD params
    $this->_head_args = array_merge($this->_head_args, $head);
}

/**
 * Parse OPTIONS
 */
protected function _parse_options()
{
    // Grab proper OPTIONS variables

```

```

        parse_str(parse_url($_SERVER['REQUEST_URI'], PHP_URL_QUERY), $options);

        // Merge both the URI segments and OPTIONS params
        $this->options_args = array_merge($this->options_args, $options);
    }

    /**
     * Parse PATCH
     */
    protected function _parse_patch()
    {
        // It might be a HTTP body
        if ($this->request->format)
        {
            $this->request->body = file_get_contents('php://input');
        }

        // If no file type is provided, this is probably just arguments
        else
        {
            parse_str(file_get_contents('php://input'), $this->_patch_args);
        }
    }

    /**
     * Parse DELETE
     */
    protected function _parse_delete()
    {
        // Set up our DELETE variables (which shouldn't really exist, but ssh!)
        parse_str(file_get_contents('php://input'), $this->_delete_args);
    }

    // INPUT FUNCTION -----

    /**
     * Retrieve a value from the GET request arguments.
     *
     * @param string $key The key for the GET request argument to retrieve
     * @param boolean $xss_clean Whether the value should be XSS cleaned or not.
     * @return string The GET argument value.
     */
    public function get($key = NULL, $xss_clean = TRUE)
    {
        if ($key == NULL)
        {
            return $this->_get_args;
        }

        return array_key_exists($key, $this->_get_args) ? $this->_xss_clean($this->_get_args[$key], $xss_clean) : NULL;
    }

    /**
     * This function retrieves a values from the OPTIONS request arguments
     *
     * @param string $key The OPTIONS/GET argument key
     * @param boolean $xss_clean Whether the value should be XSS cleaned or not
     * @return string The OPTIONS/GET argument value
     */
    public function options($key = NULL, $xss_clean = TRUE)
    {
        if ($key == NULL) {
            return $this->_options_args;
        }

        return array_key_exists($key, $this->_options_args) ? $this->_xss_clean($this->_options_args[$key], $xss_clean) : NULL;
    }

    /**
     * This function retrieves a values from the HEAD request arguments
     *
     * @param string $key The HEAD/GET argument key
     * @param boolean $xss_clean Whether the value should be XSS cleaned or not
     * @return string The HEAD/GET argument value
     */
    public function head($key = NULL, $xss_clean = TRUE)
    {
        if ($key == NULL) {
            return $this->_head_args;
        }

        return array_key_exists($key, $this->_head_args) ? $this->_xss_clean($this->_head_args[$key], $xss_clean) : NULL;
    }

    /**
     * Retrieve a value from the POST request arguments.
     *
     * @param string $key The key for the POST request argument to retrieve
     * @param boolean $xss_clean Whether the value should be XSS cleaned or not.
     * @return string The POST argument value.
     */
    public function post($key = NULL, $xss_clean = TRUE)
    {
        if ($key == NULL)
        {
            return $this->_post_args;
        }

        return array_key_exists($key, $this->_post_args) ? $this->_xss_clean($this->_post_args[$key], $xss_clean) : NULL;
    }

```



```

}

/**
 * Retrieve a value from the PUT request arguments.
 *
 * @param string $key The key for the PUT request argument to retrieve
 * @param boolean $xss_clean Whether the value should be XSS cleaned or not.
 * @return string The PUT argument value.
 */
public function put($key = NULL, $xss_clean = TRUE)
{
    if ($key == NULL)
    {
        return $this->_put_args;
    }

    return array_key_exists($key, $this->_put_args) ? $this->_xss_clean($this->_put_args[$key], $xss_clean) : NULL;
}

/**
 * Retrieve a value from the DELETE request arguments.
 *
 * @param string $key The key for the DELETE request argument to retrieve
 * @param boolean $xss_clean Whether the value should be XSS cleaned or not.
 * @return string The DELETE argument value.
 */
public function delete($key = NULL, $xss_clean = TRUE)
{
    if ($key == NULL)
    {
        return $this->_delete_args;
    }

    return array_key_exists($key, $this->_delete_args) ? $this->_xss_clean($this->_delete_args[$key], $xss_clean) : NULL;
}

/**
 * Retrieve a value from the PATCH request arguments.
 *
 * @param string $key The key for the PATCH request argument to retrieve
 * @param boolean $xss_clean Whether the value should be XSS cleaned or not.
 * @return string The PATCH argument value.
 */
public function patch($key = NULL, $xss_clean = TRUE)
{
    if ($key == NULL)
    {
        return $this->_patch_args;
    }

    return array_key_exists($key, $this->_patch_args) ? $this->_xss_clean($this->_patch_args[$key], $xss_clean) : NULL;
}

/**
 * Process to protect from XSS attacks.
 *
 * @param string $val The input.
 * @param boolean $process Do clean or note the input.
 * @return string
 */
protected function _xss_clean($val, $process)
{
    if (CL_VERSION < 2)
    {
        return $process ? $this->input->xss_clean($val) : $val;
    }

    return $process ? $this->security->xss_clean($val) : $val;
}

/**
 * Retrieve the validation errors.
 *
 * @return array
 */
public function validation_errors()
{
    $string = strip_tags($this->form_validation->error_string());

    return explode("\n", trim($string, "\n"));
}

// SECURITY FUNCTIONS -----

/**
 * Perform LDAP Authentication
 *
 * @param string $username The username to validate
 * @param string $password The password to validate
 * @return boolean
 */
protected function _perform_ldap_auth($username = '', $password = NULL)
{
    if (empty($username))
    {
        log_message('debug', 'LDAP Auth: failure, empty username');
        return false;
    }
}

```

```

log_message('debug', 'LDAP Auth: Loading Config');

$this->config->load('ldap.php', true);

$daptimeout = $this->config->item('timeout', 'ldap');
$daphost = $this->config->item('server', 'ldap');
$dapport = $this->config->item('port', 'ldap');
$daprdn = $this->config->item('binduser', 'ldap');
$dappass = $this->config->item('bindpw', 'ldap');
$dapbasedn = $this->config->item('basedn', 'ldap');

log_message('debug', 'LDAP Auth: Connect to ' . $daphost);

$dapconfig['authrealm'] = $this->config->item('domain', 'ldap');

// connect to ldap server
$dapconn = ldap_connect($daphost, $dapport);

if ($dapconn) {

    log_message('debug', 'Setting timeout to ' . $daptimeout . ' seconds');

    ldap_set_option($dapconn, LDAP_OPT_NETWORK_TIMEOUT, $daptimeout);

    log_message('debug', 'LDAP Auth: Binding to ' . $daphost . ' with dn ' . $daprdn);

    // binding to ldap server
    $dapbind = ldap_bind($dapconn, $daprdn, $dappass);

    // verify binding
    if ($dapbind) {
        log_message('debug', 'LDAP Auth: bind successful');
    } else {
        log_message('error', 'LDAP Auth: bind unsuccessful');
        return false;
    }
}

// search for user
if (($res_id = ldap_search($dapconn, $dapbasedn, "uid=$username")) == false) {
    log_message('error', 'LDAP Auth: User ' . $username . ' not found in search');
    return false;
}

if (ldap_count_entries($dapconn, $res_id) != 1) {
    log_message('error', 'LDAP Auth: failure , username ' . $username . ' found more than once');
    return false;
}

if (($entry_id = ldap_first_entry($dapconn, $res_id)) == false) {
    log_message('error', 'LDAP Auth: failure , entry of searchresult could not be fetched');
    return false;
}

if (($user_dn = ldap_get_dn($dapconn, $entry_id)) == false) {
    log_message('error', 'LDAP Auth: failure , user-dn could not be fetched');
    return false;
}

// User found, could not authenticate as user
if (($link_id = ldap_bind($dapconn, $user_dn, $password)) == false) {
    log_message('error', 'LDAP Auth: failure , username/password did not match: ' . $user_dn);
    return false;
}

log_message('debug', 'LDAP Auth: Success ' . $user_dn . ' authenticated successfully');

$this->user_ldap_dn = $user_dn;
ldap_close($dapconn);
return true;
}

/**
 * Perform Library Authentication - Override this function to change the way the library is called
 *
 * @param string $username The username to validate
 * @param string $password The password to validate
 * @return boolean
 */
protected function _perform_library_auth($username = '', $password = NULL)
{
    if (empty($username))
    {
        log_message('debug', 'Library Auth: failure , empty username');
        return false;
    }

    $auth_library_class = strtolower($this->config->item('auth_library_class'));
    $auth_library_function = strtolower($this->config->item('auth_library_function'));

    if (empty($auth_library_class))
    {
        log_message('debug', 'Library Auth: failure , empty auth_library_class');
        return false;
    }

    if (empty($auth_library_function))
    {

```

```

        log_message('debug', 'Library Auth: failure , empty auth_library_function ');
        return false;
    }

    $this->load->library($auth_library_class);
    return $this->$auth_library_class->$auth_library_function($username, $password);
}

/**
 * Check if the user is logged in.
 *
 * @param string $username The user's name
 * @param string $password The user's password
 * @return boolean
 */
protected function _check_login($username = '', $password = NULL)
{
    if (empty($username))
    {
        return FALSE;
    }

    $auth_source = strtolower($this->config->item('auth_source'));

    if ($auth_source == 'ldap')
    {
        log_message('debug', 'performing LDAP authentication for $username');
        return $this->_perform_ldap_auth($username, $password);
    }

    if ($auth_source == 'library')
    {
        log_message('debug', 'performing Library authentication for $username');
        return $this->_perform_library_auth($username, $password);
    }

    $valid_logins = $this->config->item('rest_valid_logins');

    if (! array_key_exists($username, $valid_logins))
    {
        return FALSE;
    }

    // If actually NULL (not empty string) then do not check it
    if ($password != NULL AND $valid_logins[$username] != $password)
    {
        return FALSE;
    }

    return TRUE;
}

/**
 * @todo document this.
 */
protected function _prepare_basic_auth()
{
    // If whitelist is enabled it has the first chance to kick them out
    if (config_item('rest_ip_whitelist_enabled'))
    {
        $this->_check_whitelist_auth();
    }

    $username = NULL;
    $password = NULL;

    // mod_php
    if ($this->input->server('PHP_AUTH_USER'))
    {
        $username = $this->input->server('PHP_AUTH_USER');
        $password = $this->input->server('PHP_AUTH_PW');
    }

    // most other servers
    elseif ($this->input->server('HTTP_AUTHENTICATION'))
    {
        if (strpos(strtolower($this->input->server('HTTP_AUTHENTICATION')), 'basic') == 0)
        {
            list($username, $password) = explode(':', base64_decode(substr($this->input->server('HTTP_AUTHENTICATION'), 6)));
        }
    }

    if (! $this->_check_login($username, $password))
    {
        $this->_force_login();
    }
}

/**
 * @todo Document this.
 */
protected function _prepare_digest_auth()
{
    // If whitelist is enabled it has the first chance to kick them out
    if (config_item('rest_ip_whitelist_enabled'))
    {
        $this->_check_whitelist_auth();
    }
}

```

```

$uniqid = uniqid(""); // Empty argument for backward compatibility
// We need to test which server authentication variable to use
// because the PHP ISAPI module in IIS acts different from CGI
if ($this->input->server('PHP_AUTH_DIGEST'))
{
    $digest_string = $this->input->server('PHP_AUTH_DIGEST');
}
elseif ($this->input->server('HTTP_AUTHORIZATION'))
{
    $digest_string = $this->input->server('HTTP_AUTHORIZATION');
}
else
{
    $digest_string = "";
}

// The $_SESSION['error-prompted'] variable is used to ask the password
// again if none given or if the user enters wrong auth information.
if (empty($digest_string))
{
    $this->_force_login($uniqid);
}

// We need to retrieve authentication informations from the $auth_data variable
preg_match_all('@(username|nonce|uri|nc|cnonce|qop|response)=[\']?([\']?)+@', $digest_string, $matches);
$digest = (empty($matches[1]) || empty($matches[2])) ? array() : array_combine($matches[1], $matches[2]);

if ( ! array_key_exists('username', $digest) OR !$this->_check_login($digest['username']))
{
    $this->_force_login($uniqid);
}

$valid_logins = $this->config->item('rest_valid_logins');
$valid_pass = $valid_logins[$digest['username']];

// This is the valid response expected
$A1 = md5($digest['username'].':'.$this->config->item('rest_realm').':'.$valid_pass);
$A2 = md5(strtoupper($this->request->method).':'.$digest['uri']);
$valid_response = md5($A1.':'.$digest['nonce'].':'.$digest['nc'].':'.$digest['cnonce'].':'.$digest['response']);

if ($digest['response'] != $valid_response)
{
    set_status_header(401);
    exit;
}
}

/**
 * Check if the client's ip is in the 'rest_ip_blacklist' config
 */
protected function _check_blacklist_auth()
{
    $blacklist = explode(',', config_item('rest_ip_blacklist'));

    foreach ($blacklist AS &$ip)
    {
        $ip = trim($ip);
    }

    if (!in_array($this->input->ip_address(), $blacklist))
    {
        $this->response(array('status' => false, 'error' => 'IP Denied'), 401);
    }
}

/**
 * Check if the client's ip is in the 'rest_ip_whitelist' config
 */
protected function _check_whitelist_auth()
{
    $whitelist = explode(',', config_item('rest_ip_whitelist'));
    array_push($whitelist, '127.0.0.1', '0.0.0.0');

    foreach ($whitelist AS &$ip)
    {
        $ip = trim($ip);
    }

    if ( ! in_array($this->input->ip_address(), $whitelist))
    {
        $this->response(array('status' => false, 'error' => 'Not authorized'), 401);
    }
}

/**
 * @todo Document this.
 *
 * @param string $nonce
 */
protected function _force_login($nonce = '')
{
    if (strtolower($this->config->item('rest_auth')) == 'basic')
    {
        header('WWW-Authenticate: Basic realm="'.$this->config->item('rest_realm').'"');
    }
    elseif (strtolower($this->config->item('rest_auth')) == 'digest')
    {
        header('WWW-Authenticate: Digest realm="'.$this->config->item('rest_realm').'", qop="');
    }
}

```

```

    }

    $this->response(array('status' => false, 'error' => 'Not authorized'), 401);
}

/**
 * Force it into an array
 *
 * @param object|array $data
 * @return array
 */
protected function _force_loopable($data)
{
    // Force it to be something useful
    if ( ! is_array($data) AND !is_object($data) )
    {
        $data = (array) $data;
    }

    return $data;
}

/**
 * updates the log with the access time
 *
 * @author Chris Kacerguis
 * @return boolean
 */
protected function _log_access_time()
{
    $payload['rtime'] = $this->_end_rtime - $this->_start_rtime;
    return $this->rest->db->update(config_item('rest_logs_table'), $payload, array('id' => $this->_
}

/**
 * Check to see if the API key has access to the controller and methods
 *
 * @return boolean
 */
protected function _check_access()
{
    // if we don't want to check access, just return TRUE
    if (config_item('rest_enable_access') == FALSE)
    {
        return TRUE;
    }

    $controller = explode('/', $this->uri->uri_string());

    $this->rest->db->select();
    $this->rest->db->where('key', $this->rest->key);
    $this->rest->db->where('controller', $controller[0]);

    $query = $this->rest->db->get(config_item('rest_access_table'));

    if ($query->num_rows > 0)
    {
        return TRUE;
    }

    return FALSE;
}
}

```

The following codes refer to the CSS used for the Web-based Mobi-MAPP

signin.css

```

body {
    padding-top: 30px;
    padding-bottom: 40px;
    background-color: #eee;
}

.form-signin {
    max-width: 330px;
    padding: 15px;
    margin: 0 auto;
}

.form-signin .form-signin-heading,
.form-signin .checkbox {
    margin-bottom: 10px;
}

.form-signin .checkbox {
    font-weight: normal;
}

.form-signin .form-control {
    position: relative;
    height: auto;
    -webkit-box-sizing: border-box;
    -moz-box-sizing: border-box;
    box-sizing: border-box;
    padding: 10px;
    font-size: 16px;
}

.form-signin .form-control:focus {
    z-index: 2;
}

```

```

}
.form-signin input[type="email"] {
    margin-bottom: -1px;
    border-bottom-right-radius: 0;
    border-bottom-left-radius: 0;
}
.form-signin input[type="password"] {
    margin-bottom: 10px;
    border-top-left-radius: 0;
    border-top-right-radius: 0;
}

#mobimapp-logo-cont
{
    position: relative;
    width: 100%;
    text-align: center;
    opacity: 0.6;
}

#mobimapp-logo-img
{
    position: relative;
    width: 35%;
}

utils.css-----
#reportFrame{
    overflow: hidden;
    height: 370px;
}

#disasterFrame{
    overflow: hidden;
    height: 370px;
}

#gmapcanvas{
    width: 100%;
    height: 500px;
}

.messageboard{
    font-size: 13px;
    overflow-y: auto;
    height: 350px;
    width: 100%;
}

.messagebox{
    margin-top: 20px;
    height: 225px;
    width: 100%;
}

.sidelinks{
    float: left;
    width: 23%;
}

.li-border{
    border: 1px solid #ddd;
    margin-bottom: -1px;
    border-radius: 5px;
}

.mobi-container{
    float: left;
    width: 75%;
    min-height: 500px;
    border: 1px solid;
    border-color: #ddd;
    background-color: #f8f8f8;
    border-radius: 4px 4px 0 0;
}

.mobi-login{
    float: center;
    width: 30%;
    height: 500px;
}

#geocomplete { width: 200px}

.map_canvas {
    width: 300px;
    height: 300px;
    margin: 10px 20px 10px 0;
    color: #0470a1;
    border: 1px solid;
}

.disasContainer{
    float: left;
    width: 30%;
    margin: 10px 30px 10px 0px
}

```

```

.image-preview {
    width:300px;
    height:300px;
    margin:13px 0px 15px 0px;
    background-color: #f8f8f8;
}

#multiple li {
    cursor: pointer;
    text-decoration: underline;
}

.whitecells {
    background-color: #f8f8f8;
}

.listcells {
    border-right: 1px solid;
    border-color: #ddd;
    float: left;
    text-align: left;
    font-family: Helvetica;
    white-space: nowrap;
    overflow: hidden;
    text-overflow: ellipsis;
    font-size: 13px;
    font-weight: normal;
    padding: 7px 12px;
    margin-bottom: 0;
    line-height: 1.42857143;
}

.listcells:btn {
}

.bluecells {
}

.cellshd {
    font-weight: bold;
    border-top: 1px solid;
    border-color: #ddd;
    color: #0470a1;
    padding: 4px;
    border-bottom: 1px solid;
}

.results-add {
    margin-bottom:20px;
    padding:5px;
    text-align: center;
    vertical-align: text-bottom;
}

.simpleLine {
    color: #0470a1;
    border-top: 1px solid;
}

.groupHeader {
    color: #0470a1;
    border-bottom: 1px solid;
    margin-bottom: 25px;
}

.text-aright {
    text-align: right;
}

.text-aleft {
    text-align: left;
}

.float-left {
    float: left;
}

.float-right {
    float: right;
}

.margin5 {
    margin: 5px;
}

.margin10 {
    margin: 10px;
}

.width5pc {
    width:5%;
}

.width10pc {
    width:10%;
}

.width15pc {

```

```

        width:15%;
    }

    .width20pc{
        width:20%;
    }

    .width25pc{
        width:25%;
    }

    .width30pc{
        width:30%;
    }
    .width35pc{
        width:35%;
    }
    .width40pc{
        width:40%;
    }

    .width75pc{
        width:75%;
    }

    .height400{
        height:400px
    }

    /*for footer.html*/
    #building-bg-cont
    {
        position:fixed;
        bottom:0;
        width:100%;
        z-index:-1;
        opacity:0.4;
    }

    #building-bg-img
    {
        width:50%;
        float:left;
        user-drag:none;
        -moz-user-select:none;
        -webkit-user-drag:none;
    }
    /*-----*/

```

The following codes refer to the JavaScripts used for the Web-based Mobi-MAPP

disasters.js

```

$( document ).ready(function() {

    var disaster_data;

    if (!window.location.origin) {
        window.location.origin = window.location.protocol + "://" + window.location.hostname + (window.location
    }

    $('#pushViewDisaster').click(function(){

        $('#pushDone').attr('disabled', true);
        $('#pushUpdate').attr('disabled', true);

        var essay_id = $(this).attr('id');
        $('#viewDisasterBody').html('Loading..');
        $.ajax({
            type : 'post',
            url : window.location.origin+"/mobimapp/disasters/view/"+essay_id, // in here you sh
            // data : 'post.id='+ essay_id, // here you pass your id via ajax .
            // in php you should use $_POST['post.id'] to get this value
            cache: false ,
            success : function(r)
            {
                // now you can show output in your modal
                $('#viewDisasterBody').html(r);
                disaster_data = r;
                $('#pushDone').attr('disabled', false);
                $('#pushUpdate').attr('disabled', false);
            },

            error: function(r)
            {
                // now you can show output in your modal
                $('#viewDisasterBody').html(r);
            }
        });

        $('#operations-link').attr("href",window.location.origin+"/mobimapp/drrm/operations/"+e

        /*
        var frame = "<iframe frameBorder='0' id='disasterFrame' width='100%' align='left' src="
        $('#viewDisasterBody').html(frame);
        data_to_validate = frame;
        */
    });

```



```

$( '#pushDone' ). click ( function () {
    $( '#doneDisasterBody' ). html ( disaster_data );
    $( '#viewDisasterModal' ). modal ( 'toggle' );
    $( '#confirmDone' ). attr ( 'disabled', false );
});

$( '#pushUpdate' ). click ( function () {
    if ( !window.location.origin ) {
        window.location.origin = window.location.protocol + "://" + window.location.hostname +
    }
    $( "#disasterViewForm" ). attr ( "action", window.location.origin + '/mobimapp/disasters/edit' );
    $( '#disaster_view_submit' ). trigger ( 'click' );
});

$( '#confirmDone' ). click ( function () {
    if ( !window.location.origin ) {
        window.location.origin = window.location.protocol + "://" + window.location.hostname +
    }
    $( "#disasterViewForm" ). attr ( "action", window.location.origin + '/mobimapp/disasters/done' );
    $( '#disaster_view_submit' ). trigger ( 'click' );
});

$( '#cancelDone' ). click ( function () {
    $( '#doneDisasterModal' ). modal ( 'hide' );
    $( 'body' ). removeClass ( 'modal-open' );
    $( '.modal-backdrop' ). remove ();
});

var ctr=0;

$( '#addShelter' ). click ( function () {
    $( '#shelters' ). removeClass ( 'hidden' );
    var shelterField = "<div class='form-inline' id='divshelter"+(++ctr)+">" +
        "<input type='text' required class='form-control margin5' st" +
        "<button type='button' class='btn btn-link' id='closeShelter' onClick=\"remo" +
        "</div>";
    $( '#shelters' ). append ( shelterField );
    $( '#shelter'+ctr ). geocomplete ();
});

});

function removeInput(id){
    $(id).remove();
}

-----
drmm.js
$(function(){

    $( '.assignBtn' ). click ( function (e){
        var disaster_id = $(this).attr('id');
        $( '#disasterInput' ). attr ( 'value', disaster_id );
    });

});

-----
geocomplete.js
$(document).ready(function() {
    var options = {
        map: ".map-canvas",
        details: "#formAdd"
    };
    $( "#geocomplete" ). geocomplete ( options );
    $( "#find" ). click ( function () {
        $( "#geocomplete" ). trigger ( "geocode" );
    });
});

$(window).load(function() {
    var options = {
        map: ".map-canvas",
        details: "#formEdit"
    };
    $( "#edit-geocomplete" ). geocomplete ( options );
    $( "#edit-geocomplete" ). trigger ( "geocode" );
});

-----
mydisastermap.js
$(document).ready(function() {
    map =new GMaps({
        div: '#gmapcanvas',
        lat: 14.043333,
        lng: 121.028333,
        zoom: 6,
        mapTypeId: google.maps.MapTypeId.HYBRID

```

```

});
var count = $('#disasCount').val();

var x=14;
var y=121;
var loc, dtype, remarks;
for (var i=0;i<count;i++){
    x = $('#disLocX'+i).val();
    y = $('#disLocY'+i).val();
    loc = $('#disLoc'+i).val();
    dtype = $('#disType'+i).val();
    remarks = $('#disRemarks'+i).val();
    map.addMarker({
        //alert(">>>"+'#disLocY'+i).val());
        lat: x, // $('#disLocX'+i).val(),
        lng: y, // $('#disLocX'+i).val(),
        title: loc,
        infoWindow: {
            content: '<b>Type</b>: <p>'+dtype+'</p>' +
                    '<b>Location:</b> <p>'+loc+'</p>' +
                    '<b>Remarks:</b> <p>'+remarks+'</p>'
        }
    });
}
});
});
}

reports.js
-----
$( document ).ready(function() {

    var data_to_validate;
    if (!window.location.origin) {
        window.location.origin = window.location.protocol + "://" + window.location.hostname + (window.location
    }

    $('#pushValidate').click(function(e){

        e.preventDefault();
        $('#pushDelete').attr('disabled', true);
        $('#pushValid').attr('disabled', true);
        var essay_id = $(this).attr('id');
        $('#validateModalBody').html('Loading..');
        $.ajax({
            type : 'get',
            url : window.location.origin+'/mobimapp/reports/view/'+essay_id, // in here you should
            // data : 'post_id='+ essay_id, // here you pass your id via ajax .
            // in php you should use $_POST['post_id'] to get this value

            cache: false,
            success : function(r)
            {
                // now you can show output in your modal
                $('#validateModalBody').html(r);
                data_to_validate = r;
                $('#pushDelete').attr('disabled', false);
                $('#pushValid').attr('disabled', false);
            },

            error: function(r)
            {
                $('#pushDelete').attr('disabled', true);
                $('#pushValid').attr('disabled', true);
                // now you can show output in your modal
                $('#validateModalBody').html(r);
            }
        });

        /*
        if (!window.location.origin) {
            window.location.origin = window.location.protocol + "://" + window.location.hostname +
        }
        var frame = "<iframe frameBorder='0' id='reportFrame' width='100%' align='left' src='"+
        $('#validateModalBody').html(frame);
        data_to_validate = frame;
        */
    });

    $('#pushDelete').click(function(){
        $('#deleteModalBody').html(data_to_validate);
        $('#validateModal').modal('toggle');
        $('#confirmDelete').attr('disabled', false);
    });

    $('#pushValid').click(function(){
        $('#validModalBody').html(data_to_validate);
        $('#validateModal').modal('toggle');
        $('#confirmValid').attr('disabled', false);
    });

    $('#pushInvalid').click(function(){
        $('#invalidModalBody').html(data_to_validate);
        $('#validateModal').modal('toggle');
    });

    $('#confirmDelete').click(function(){
        if (!window.location.origin) {
            window.location.origin = window.location.protocol + "://" + window.location.hostname +
        }
        $('#reportValidation').attr("action",window.location.origin+'/mobimapp/reports/delete/%2B3');
        $('#report-validation-submit').trigger('click');
    });
}

```

```

});
$('#confirmValid').click(function(){
    if (!window.location.origin) {
        window.location.origin = window.location.protocol + "://" + window.location.hostname +
    }
    $('#reportValidation').attr("action",window.location.origin+'/mobimapp/reports/valid/%2B10');
    $('#report_validation_submit').trigger('click');
});
$('#confirmInvalid').click(function(){
    if (!window.location.origin) {
        window.location.origin = window.location.protocol + "://" + window.location.hostname +
    }
    $('#reportValidation').attr("action",window.location.origin+'/mobimapp/reports/delete/%2D20');
    $('#report_validation_submit').trigger('click');
});
$('.cancelDelete').click(function(){
    $('#deleteModal').modal('hide');
    $('body').removeClass('modal-open');
    $('.modal-backdrop').remove();
});
$('.cancelValid').click(function(){
    $('#validModal').modal('hide');
    $('body').removeClass('modal-open');
    $('.modal-backdrop').remove();
});
// alert($('.listcells').css('height'));
// $('.pushValidate').css('height', $('.listcells').css('height'));
});
}
usermgmt.js
$(function(){
    var regionSelect = $('#placename').html();
    $('#drrmc_type').on('change', function(){
        //onloadOfficeType();
        var provinceSelect = '<label style="margin-top: 20px;">Province</label>'+
            '<select id="officeProvince" name="province"
        if($('#drrmc_type').val()== 'rdrrmc'){
            $('#placename').html(regionSelect);
        }
        else if($('#drrmc_type').val()== 'pdrrmc'){
            $('#placename').html(regionSelect+provinceSelect);
            loadProvince();
        }
        else if($('#drrmc_type').val()== 'ldrrmc'){
            $('#placename').html(regionSelect+provinceSelect + '<label style="margin-top: 20px">Lo
            '<input type="text" name="locality" class="form
            loadProvince();
        }
    });
    //$('#officeRegion').on('change', function(){
    //    if($('#officeRegion').val()== 'National Capital Region'){
    //        $('#officeProvince').attr('value', 'Metro Manila')
    //    }
    //});
    if (!window.location.origin) {
        window.location.origin = window.location.protocol + "://" + window.location.hostname + (window.location
    }
    $('#pushViewMobileUser').click(function(e){
        $('#pushUnban').addClass("hidden");
        $('#pushUnban').addClass("disabled");
        e.preventDefault();
        $('#pushBan').attr('disabled', true);
        var essay_id = $(this).attr('id');
        $('#viewMobileUserBody').html('Loading..');
        $.ajax({
            type : 'get',
            url : window.location.origin+'/mobimapp/usermgmt/mobileview/'+essay_id, // in here you
            // data : 'post_id='+ essay_id, // here you pass your id via ajax .
            // in php you should use $_POST['post_id'] to get this value
            cache: false ,
            success : function(r)
            {
                // now you can show output in your modal
                $('#viewMobileUserBody').html(r);
                if($('#isBanned').attr('value')== "yes"){
                    $('#pushBan').attr('disabled', true);
                    $('#pushBan').hide();
                    $('#pushUnban').removeClass("hidden");
                    $('#pushUnban').removeClass("disabled");
                }
                else{
                    $('#pushBan').attr('disabled', false);
                    $('#pushBan').show();
                }
            }
        },
    ),

```

```

        error: function(r)
        {
            $('#pushBan').attr('disabled', true);
            // now you can show output in your modal
            $('#viewMobileUserBody').html(r);
        }
    });
});

$("#pushBan").click(function(){
    if (!window.location.origin) {
        window.location.origin = window.location.protocol + "://" + window.location.hostname +
    }
    $('#mobileUserViewForm').attr("action",window.location.origin+'/mobimapp/usermgmt/banmobile/');
    $('#mobile_user_view_submit').trigger('click');
});

$("#pushUnban").click(function(){
    if (!window.location.origin) {
        window.location.origin = window.location.protocol + "://" + window.location.hostname +
    }
    $('#mobileUserViewForm').attr("action",window.location.origin+'/mobimapp/usermgmt/unbanmobile/');
    $('#mobile_user_view_submit').trigger('click');
});

});

function onloadOfficeType(){

    var regionSelect = $('#placename').html();

    var provinceSelect = '<label style="margin-top: 20px">Province</label>'+
        '<input type="text" id="officeProvince" name='

    if($('#drmmc_type').val()=='rdrmmc'){
        $('#placename').html(regionSelect);
    }
    else if($('#drmmc_type').val()=='pdrmmc'){
        $('#placename').html(regionSelect+provinceSelect);
    }
    else if($('#drmmc_type').val()=='ldrrmc'){
        $('#placename').html(regionSelect+provinceSelect + '<label style="margin-top: 20px">Lo
        '<input type="text" name="locality" class="for

    }

}

function loadProvince(){
    $('#officeRegion').change(function(){
        if($('#officeRegion').val()=="Ilocos Region"){
            $('#officeProvince').load().html("<option value='Ilocos Norte'>Ilocos N
        }
        if($('#officeRegion').val()=="Cagayan Valley"){
            $('#officeProvince').load().html("<option value='Batanes'>Batanes</optio
        }
        if($('#officeRegion').val()=="Central Luzon"){
            $('#officeProvince').load().html("<option value='Aurora'>Aurora</option>
        }
        if($('#officeRegion').val()=="CALABARZON"){
            $('#officeProvince').load().html("<option value='Batangas'>Batangas</op
        }
        if($('#officeRegion').val()=="MIMAROPA"){
            $('#officeProvince').load().html("<option value='Marinduque'>Marinduque
        }
        if($('#officeRegion').val()=="Bicol Region"){
            $('#officeProvince').load().html("<option value='Albay'>Albay</option><
        }
        if($('#officeRegion').val()=="Western Visayas"){
            $('#officeProvince').load().html("<option value='Aklan'>Aklan</option><
        }
        if($('#officeRegion').val()=="Central Visayas"){
            $('#officeProvince').load().html("<option value='Bohol'>Bohol</option><
        }
        if($('#officeRegion').val()=="Eastern Visayas"){
            $('#officeProvince').load().html("<option value='Biliran'>Biliran</optio
        }
        if($('#officeRegion').val()=="Zamboanga Peninsula"){
            $('#officeProvince').load().html("<option value='Zamboanga del Norte'>Z
        }
        if($('#officeRegion').val()=="Northern Mindanao"){
            $('#officeProvince').load().html("<option value='Bukidnon'>Bukidnon</op
        }
        if($('#officeRegion').val()=="Davao Region"){
            $('#officeProvince').load().html("<option value='Compostella Valley'>Co
        }
        if($('#officeRegion').val()=="SOCCSKSARGEN"){
            $('#officeProvince').load().html("<option value='Cotabato'>Cotabato</op
        }
        if($('#officeRegion').val()=="Caraga"){
            $('#officeProvince').load().html("<option value='Agusan del Norte'>Agus
        }
        if($('#officeRegion').val()=="Autonomous Region in Muslim Mindanao"){
            $('#officeProvince').load().html("<option value='Basilan'>Basilan</optio
        }
        if($('#officeRegion').val()=="Cordillera Administrative Region"){
            $('#officeProvince').load().html("<option value='Abra'>Abra</option><op
        }
        if($('#officeRegion').val()=="National Capital Region"){
            $('#officeProvince').load().html("<option value='Metro Manila'>Metro Ma
        }
    }
}

```

```

    });
}

utils.js

function readURL(input) {
    if (input.files && input.files[0]) {
        var reader = new FileReader();
        reader.onload = function(e) {
            $('#previewHolder').attr('src', e.target.result);
            $('#previewHolder').removeClass('hidden');
        }
        reader.readAsDataURL(input.files[0]);
    }
}

$("#filePhoto").change(function() {
    readURL(this);
});

$("#removeFile").on("click", function() {
    var control = $("#filePhoto");
    control.replaceWith( control.clone( true ) );
    $('#previewHolder').removeAttr("src");
    $('#previewHolder').addClass('hidden');
});

$("#removeUploadNew").on("click", function(){
    $('#previewHolder').attr("src","");
    $('#previewHolder').addClass("hidden");
    $('#uploadNewPhoto').addClass("hidden");
    $('#uploadDisasterPhoto').removeClass("hidden");
    $('#prev-image').attr("value","");
});

function resizeIframe(obj)
{
    {
        {
            obj.style.height = 0;
        }
        ;
        {
            obj.style.height = obj.contentWindow.document.body.scrollHeight + 'px';
        }
    }
}

```

XII. Acknowledgement

Una sa lahat, pinasasalamat ko ang Panginoon, sa lahat ng kanyang binigay na blessings at sa lahat ng mga pagsubok na hinarap ko. Kung totoo nga na sinagot ng Nine Mornings ang kahilingan ko, maraming salamat!

Sa aking pamilyang inspirasyon ko, na laging andiyan para sakin. Para sa aking Mader, na halos isang dekadang sa pagiging OFW at pagiging madrama tuwing kausap ko sa telepono, sa aking Pader, na parang kuya ko na rin kasi astang bagets pa din at sa mga nautangan niya, maraming salamat. Sa aking mga kapatid, Kuya Tok, graduate na tayo! Si Lance na walang sawang katabi ko sa pagcocomputer. At sa aking lola, lolo, tita liway, tito mhel, tito lotlot, tita lizas at sa aking mga pinsan, maraming salamat!

Sa aking adviser, Maam Sheila, salamat po sa pasensya at suporta kahit nahuhuli lagi ako sa mga consultations ko po.

Ang aking mga kabarkada, Team Smile! Kim, Jerson, Jaye, Jayrell, Colin. Alam na kung saan sunod nito! Rak na ituuuuu!

Kay Jayvee at Zee, Sa mga overnights sa pag-aaral na nauuwi sa movie marathon at pagtulong sakin sa layout ng Mobi-MAPP.

Sa aking mga blockmates na nagging kasama ko sa apat na taon, at sa mga guro ko, maraming salamat!

Pinasasalamat ko rin ng sobra ang mahal kong mga brods sa Kapatirang Sigma Kappa Pi, sa paghubog sakin, sa kulitan, sa hirap at saya ng ating pagsasama, sa pangaasar sakin na Best Thesis ako, pasensya brods. At sa mga mahal kong EDIT sisses, maraming salamat!

Sa aking mga barkada mula elementary at high school, sa mga barkada kong dotaboys, Carlo, Marvin, Kim, Levi, Jayson, RR, Bontype, Jose, at kay Yurie nan amiss ko ang binyag dahil sa SP ko, maraming salamat!

At syempre, sa pinakamamahal ko, sa yo Gee, na naging inspirasyon ko, na laging andyan para sakin, sa pagsama sa puyatan days ko sa SP, na kaduet ko lagi sa mga sabaw moments (karamihan sa Frozen). Sa utang kong oras at panahon, hayaan mo kong bumawi pagtapos nito. Maraming salamat Geebabes! Love you. Awtsuu!

At sa lahat ng mga tumulong sakin na hindi ko nabanggit, Maraming Salamat!

XIII. Bibliography

- [1] “World disaster 2012: Focus on forced migration and displacement,” October 2012. International Federation of Red Cross and Red Crescent Societies.
- [2] D. West and E. Valentini, “How mobile devices are transforming disaster relief and public safety,” *Center for Technology Innovation at Brookings*, 2013.
- [3] K. M. Rahman, T. Alam, and M. Chowdhury, “Location based early disaster warning and evacuation system on mobile phones using openstreetmap,” *IEEE Conference on Open Systems ICOS*, 2012.
- [4] J. Russel, “Philippines named social networking capital of the world,” May 2011.
- [5] “Project noah.” Available from <http://noah.dost.gov.ph/>.
- [6] A. Carrara and F. Guzzetti, “Geographical information systems in assessing natural hazards,” in *Gis and Natural Hazards: An overview from a Gis Perspective*, pp. 21–34, Kluwer Academic Publishers, 1995.
- [7] J. Ye, G. Lin, M. Zhang, Z. Lin, and J. Chen, “Gis-based study of natural disaster vulnerability for xiamen city,” *18th International Conference on Geoinformatics*, 2010.
- [8] A. Pan, M. Yang, and B. Chen, “Flood hazard evaluation and gis in guangzhou,” *International Conference on Multimedia Technology (ICMT)*, 2010.
- [9] J. Santillan, E. Paringit, R. Ramos, R. Mendoza, N. Espanola, and J. Alconis, “Near real-time flood extent monitoring in marikina river, philippines: Model parameterization using remotely-sensed data and field measurements,” in *Proceedings of the 33rd Asian Conference on Remote Sensing (ACRS)*, 2012.

- [10] K. K. Lwin and Y. Muruyama, "Web-based gis system for real-time field data collection using a personal mobile phone," *Journal of Geographic Information System*, 2011.
- [11] J. Ortmann, M. Limbu, D. Wang, and T. Kauppinen, "Crowdsourcing linked open data for disaster management," in *Proceedings of Terra Cognita 2011, The 10th International Semantic Web Conference*, 2011.
- [12] H. Gao, "Harnessing the crowdsourcing power of social media for disaster relief," *IEEE Intelligent Systems*, 2011.
- [13] R. Fontugne, K. Cho, Y. Won, and K. Fukuda, "Disasters seen through flickr cameras,," in *Proceedings of Special Workshop on Internet and Disasters (SWID)*, ACM, 2011.
- [14] C. Hodapp, M. Robbins, J. Gray, and A. Graettinger, "Damage tracker - a cloud and mobile system for collecting damage information after natural disasters," in *Proceedings of the 51st ACM Southeast Conference*, 2013.
- [15] "Openstreetmap." Available from <http://wiki.openstreetmap.org/>.
- [16] "Cloudmade." Available from <http://cloudmade.com/technologies/hybrid>.
- [17] B. Gamboa, "Mobile disaster communication system." Undergraduate Special Problem, Dept. Phys. and Math., UP Manila, 2013.
- [18] J. Alimanza, "Mobile disaster reporting system." Undergraduate Special Problem, Dept. Phys. and Math., UP Manila, 2011.
- [19] "Definition of disaster." Retrieved from Merriam-Webster (<http://www.merriamwebster.com/dictionary/disaster>).
- [20] "Definition of disaster." Retrieved from International Federation of Red Cross (<http://ifrc.org>).

- [21] “Definition of disaster risk reduction.” Retrieved from The United Nations Office of Disaster Risk Reduction (UNISDR).
- [22] “National disaster risk reduction and management framework,” 2012. Framework from NDRRMC.
- [23] “Republic act 10121 an act strengthening the ndrrmf and institutionalizing the national disaster risk reduction and management plan, appropriating funds therefor and for other purposes.” from Philippine Consitution passed 2009.
- [24] “Definition of smartphone.” Available from PhoneScoop(<http://www.phonescoop.com/glossary/term.php?gid=131>).
- [25] “Geographic information systems as an integrating technology: Context, concepts, and definitions.” Retrieved from <http://www.colorado.edu/geography/gcraft/notes/intro/intro.html>.
- [26] “Definition of web services.” Available from <http://searchsoa.techtarget.com/definition/Web-services>.
- [27] “Google play services.” Available from <http://developer.android.com/google/play-services/index.html>.
- [28] “Google maps.” Available from <https://developers.google.com/maps/>.
- [29] “Google maps api.” Available from <https://developers.google.com/maps/>.
- [30] “Google locations api.” Available from <https://developer.android.com/google/play-services/location.html>.
- [31] “Google cloud messaging.” Available from <http://developer.android.com/google/gcm/index.html>.