

I. Introduction

A. Background of the Study

Pets, either professionally trained or average domestic have various benefits to humans. Guide for the blind, ease loneliness, reduce stress, promote social interaction, encourage exercise and playfulness are some benefits of having a pet. Caring for a pet may also help a person live longer. [1]

Human health is inextricably linked to animal health and production. [2] One of the most enriching and delightful experience in life is to own a pet. [3] Pets help a person to become more responsible, when they get a pet there is no doubt they will take care of them. The life and death of another being on their hands makes them become a more responsible person. They learn to manage their time, since it will surely take a significant amount of their time taking care of pets. They will become a more nurturing and caring person. The emotional link between them and their pet could both last for a lifetime. And surely, they'll get a better health.

However, there are some cases that a person is not allowed to have a pet, or if they do they must know the potential risk for diseases that can be passed from animals to humans. Also during these cases, the service of veterinary clinic is important. Veterinarians could give a person all information they need on infections that they might get from their pets. Veterinarians could also examine all their pets and give their pet vaccinations if necessary.

In Philippines, availability of information about location of veterinary clinics, specialization and services of veterinarians is very limited. This information would be beneficial in helping the companion animals for protection as well as for being able to protect people from diseases that could be transferred from animals to human.

From an interview with one of the veterinarians of UP Veterinary Hospital located inside UP Diliman, veterinary clinic/hospital operations are basically the same as operations of human hospitals aside from the difference of whom to be treated. The same procedures are followed on treating animal's diseases, confinement and operation. Veterinarian's specialization is almost the same as normal doctor's specialization. But even though the facilities and technology used for treatment of animal disease are modern, there are no applications or automated systems used for maintaining animal data records. Specifically in UP Veterinary Hospital, manual input of animal records are done by hospital secretary, transactions from registration to payment are done manually as well.

B. Statement of the Problem

The essential role of animals in the transmission of infectious diseases has long been recognized. Not only were animals known to be responsible for the maintenance of infections in nature by harboring and enabling pathogens to survive; animals were also found to be an active cause of pathogen spread in the environment and to other animals, including man. [4]

Availability of information about veterinary clinics that could help prevent the transmission of infectious diseases from animals is very limited. It is hard to find the nearest veterinary clinic that could cater the needs of pet owners with the current available systems on the internet.

Information on the internet sometimes lack details on services and specialization of veterinary clinics thus leading pet owners on wrong clinic. This causes loss in time, effort and even money for both the pet owner and the clinic.

C. Objectives

The project aims to create a Web and Android-based application that would provide necessary information to pet owners of the location, and specialization of the nearest veterinary clinic. It aims to help them find or search the most suitable clinic to cater their pet's needs.

General Features/Functionalities:

- Allow users to issue queries on the nearest veterinary clinic available to their location.
The system will use an online interactive map service (Google Maps) to show the nearest search results.
- Allow users to view information on each clinic available on the system.
- Allow users to view information about veterinarians and their schedule.
- Allow veterinarians to enter consultation information, along with the laboratory results and medication
Allow users to schedule appointment with a veterinarian on the clinic.

Specific functionalities for each access levels:

Web-based application:

1. To allow registered users with the following roles to:

Pet Owner:

- a. View own profile
- b. Update personal information
- c. View clinics' information
- d. Request schedule for consultation

- e. View own pet information
- f. Add own pet information
- g. Update own pet information

Clinic Owner:

- a. Request for clinic account
- b. View clinic information
- c. Update clinic information
- d. View own veterinarian information
- e. Add veterinarian to system and clinic
- f. View own veterinarian schedule

Veterinarian:

- a. View own information
- b. Update own information
- c. View pet information
- d. View servicing clinic information
- e. View personal schedule
- f. Update personal schedule
- g. Add pet information result from consultation
- h. Approve appointment request from client

- 2. To allow the System Administrator who maintains the overall system's database to:
 - a. Approve new clinic request
 - b. Deactivate or activate an existing clinic account

Android application:

To allow all users to:

- a. Search available veterinary clinic based on location
- b. View clinic information
- c. View veterinarian information

D. Significance of the Study

Looking for a good veterinarian to cater pet owner needs with less time taken than usual, this system would create a way to connect more pet lovers to veterinarians and the community.

Animal diseases that human can catch are called zoonoses. Pets are no exception with possibility to carry worms, viruses, and bacteria causing disease that is transferrable to human [5], thus regular consultation on veterinarians is important. By providing the necessary information on location, specialization and services on veterinary clinics, pet owners could protect their beloved companion and themselves from this kind of disease.

It helps reduce anxiety for pet owners or users looking for clinics for their pet to be checked or cured. Scheduling visit and consultation will also not be a problem. Pet records and medical history are maintained and keep secured by the system.

E. Scope and Limitations

1. Only information of registered clinics is available.
2. Accuracy of information, data or location is based from the data provided by clinic and not automatically obtained by the system.

3. Speed of queries is based on internet connection speed.
4. Only covers veterinary clinics located in the Philippines.
5. Veterinarian account can only be created by registered clinic
6. Pet owner account can only be created by registered veterinarian
7. Only legally allowed pets are considered as type of pet and veterinarian's specialization
as of this development

II. Review of Related Literature

Medical informatics is the discipline concerned with the theoretical foundations and application of information science, engineering, and computer technology in medical teaching, research, and practice. [6] Medical informatics areas are an integral part of veterinary practice – for example, digital radiology and patient information software. Though information systems for practice management and health are common, there is only minimal system that caters the need for veterinary medicine. A greater understanding of the informatics environment of veterinary practice is necessary to promote the integration of knowledge base resource in veterinary practice. [7]

Web-Based Integrated Information System of Phoenix Veterinary Clinic in Cagayan de Oro City, Philippines is a system designed for quick and easy retrieval of pet records. It is used to provide reliable and secured data preservation for the clinic. With the use of the system, the clinic can operate more efficiently because every information is managed electronically. [8]

There are also available software systems to control, help and improve the current process implemented. One example of this is CryoWEB. CryoWeb is web software used for the documentation of cryo-preserved material in animal gene banks. The software is generally applicable to all livestock breeds, independent of donors' species and type of preserved animals. It also provides a way to trace the usage of the sample gene and the sample vessel within the storage facilities. [9]

A Geographic Information System (GIS) is an integrated collection of computer software and data used to view and manage information connected with specific locations, analyze spatial relationships, and model spatial processes. It can be used to provide information on affected

location that is critical to decision making associated with large outbreaks, whether animal or human. Animal health authorities stated that GIS is one of the most powerful weapons they have to control diseases and minimize economic loss. [10]

Nearest Hospital Search and Inquiry System (NHSIS) is a Special Problem made by Meybel R. Martirez on 2004. NHSIS is a location based system designed to locate the nearest hospitals from user's location. It is a web-based application that generates clinical data provided by the registered hospitals. It allows both registered and general users to issue their queries on the hospital nearest to their location, doctors' schedule of activities and the list of available facilities and specialties. [11]

Later on during 2010, another Special Problem was made, Telehealth's Geographic Information System (TGIS) by Alexis Angelo Sadie. One of its objectives is to provide data analysis to map and track origin of outbreak for diseases. It includes maps and charts in presenting analyzed data. It uses Geographic Information system for a better visualization of the geographical location associated with a specific study. It also integrates the result of NHSIS as support for locating nearest hospitals. [12]

Web-based geographic search engine for location-aware search in Singapore – is a WEB-based search engine that enables user to search local specific location in Singapore and return the distance from the current user's location. The user can view the searched location on a satellite map instead of the traditional two dimensional map currently use in street view directories. It uses Haversine's formula to calculate the distance between two coordinates. Whereas, the user is required to input their current coordinate, then this coordinate is used during search to be able to return the distance from user's current location and the searched location. [13]

Geospatial health resource data and improved software analysis enable development of digital “health maps”. The information derived from GIS-based descriptive maps provides an operational tool for planning, monitoring and managing control programs for infectious diseases. [14]

GIS-Enabled Cancer Atlas is a client-server application designed to provide information related to cancer-control activities. The system is designed primarily to enable web distribution of maps to support public health activities related to cancer. It uses GIS as a tool to integrate, analyze, and present geo-referenced data about cancer and associated covariates and risk factors. The Pennsylvania Cancer Atlas (PA CA) for example, is designed to serve not as a product for public use but as a model for the kind of atlases that state departments of public health, state cancer registries, and related organizations can implement to disseminate the data they generate and to help users understand these data. [15]

Web Based Bicycle Trip Planning for Broward County, Florida is a GIS application designed to promote bicycle as mode of transportation for increasing urban traffic congestion. It uses ArcGIS Sever framework with ArcGIS JavaScript Extension for Google Maps API and the Google Local Search Control for Maps API that allows user to access local search results from businesses, addresses, road and intersections, political entities to geographic coordinates. [16]

Implementation of Location based Services in Android using GPS and Web Services uses Android’s build in support for location base services that allows mobile user to exploit their knowledge of the geographical position and use that information to provide different kind of services like advising clients of current traffic conditions, providing routing information, helping them find nearby hotels. [17]

Restaurant Finder is an Android application that helps users to find the restaurants according to their preferences. It provides easy access to nearby restaurants and its location on map. It allows user to search by category such as type of cuisine, rating of restaurants and distance from current location. The application also provides opportunity for customers to write reviews and ratings for the restaurant's service. With the use of Google Maps API it allows users to view top five (5) restaurants present in the current location and it also provides route to the restaurants. The application also makes it easy for the user to send out invitations to his/her friends/colleagues for meetings and also mark the events in their calendar. [18]

Women Veterans' Experience with a Web-Based Diabetes Prevention Program: A Qualitative Study to Inform Future Practice, is a Web-based DPD intervention that aims to qualitatively explore women veterans' early experiences with diabetes, estimate weight loss, participation and engagement to provide context for its findings. Participants perceived the DFD program as an appealing way of initiating lifestyle changes and made them feel accountable in achieving their daily goals. The online program was convenient because it could be accessed at any time, and many found that it integrated well into daily life. [27]

Exploring the Far Side of Mobile Health: Information Security and Privacy of Mobile Health Apps on iOS and Android establish an overview of Mobile health (mHealth) apps offered on iOS and Android with a special focus on potential damage to users through information security and privacy infringements. Through this study, it is found that various kinds of mHealth apps collect and offer critical, sensitive, private medical information, calling for a special focus on information security and privacy of mHealth apps. In order to foster user acceptance and trust, appropriate security measures and processes need to be devised and employed so that users can

benefit from seamlessly accessible, tailored mHealth apps without exposing themselves to the serious repercussions of information security and privacy infringements. [28]

III. Theoretical Framework

A. Pet Medical Record

Many veterinary medical offices have changed their system of recordkeeping from a handwritten one to electronic system. Whether a veterinary practice keeps medical records in a handwritten format or in an electronic format, the same rules applies; there is no single standard of completeness for each record. However, medical records should contain the following key elements; client complaints, vaccination or preventive health update, assessments, full prescription information, and one of the 3 R's (recheck/recall/reminder).

This SP serves as an information system for pet medical records. It stores data from appointment to medication and will be available from the time they used the system. The information stored may be used as reference if needed, be it future use or for practice of veterinary medicine.

B. Android OS

Android or Android OS is the operating system that powers more than one billion smartphones and tablets. [19] It is based on Linux kernel and currently developed by Google and primarily designed for touchscreen devices. On top of the Linux kernel, there are the middleware, libraries and APIs written in C, and application software running on an application framework which includes Java-compatible libraries. This enables any android device the ability to evolve, and open the possibility of new android application to be developed.

Android applications are usually developed in the Java programming language using the Android Software Development Kit (SDK). The Android SDK provides a fast and reliable

development environment for building, testing, and debugging Android applications in Windows, Mac or Linux. It gives mobile developers the chance to play around with the new platform. It contains everything you need to start building applications, as well as a fully functional device emulator for testing out when you've finished.

C. Geographic Location Systems

Geographic location systems or coordinate systems were developed as a tool to describe specific geographic locations and are used for navigation and mapping. Most of these systems use coordinate values, which are expressed as numeric or alphanumeric characters, to define geographic locations. Coordinate systems are based on reference points from which position measurements are made. [20]

Coordinate systems enable geographic datasets to use common locations for integration. A coordinate system is a reference system used to represent the locations of geographic features, imagery, and observations, such as Global Positioning System (GPS) locations, within a common geographic framework. It is a measurement framework, which either two geographic or planimetric coordinates of the earth and is projected onto a two-dimensional planar surface. It uses feet or meter for projected coordinate systems or decimal degrees for latitude-longitude. [21]

D. Google Maps

Google Maps is a Web-based service that provides detailed information about geographical regions and sites around the world. In addition to conventional road maps, Google Maps offers aerial and satellite views of many places. It serves as a route planner that offers

directions on uses of public transport. Google Maps API makes it possible for Web site administrators or designers to embed Google Maps on their applications. Google Street View enables user to view and navigate through the horizontal and vertical panoramic street level images. [22]

E. JavaScript Object Notation (JSON)

JavaScript Object Notation (JSON) is a lightweight data-interchange format. It is easy for humans to read and write. It is easy for machines to parse and generate. JSON is a text format that is completely language independent but uses conventions that are familiar to programmers of the C-family of languages, including C, C++, C#, Java, JavaScript, Perl, Python, and many others. These properties make JSON an ideal data-interchange language. [23]

JSON is built on two structures:

1. A collection of name/value pairs. In various languages, this is realized as an object, record, struct, dictionary, hash table, keyed list, or associative array.

```
{ string 1 ; value 1, string 2 ; value 2, string ... ; value ... } <- object
```

Figure 1. JSON representation of an object

2. An ordered collection of values. In most languages, this is realized as an array, vector, list, or sequence.

```
[ value 1, value 2, value ... ] <- array
```

Figure 2. JSON representation of a list

An Android application and a remote MySQL database cannot communicate with each other directly. By utilizing PHP to send query and return it in JSON format, it is possible for an Android application to connect and obtain data from a database.

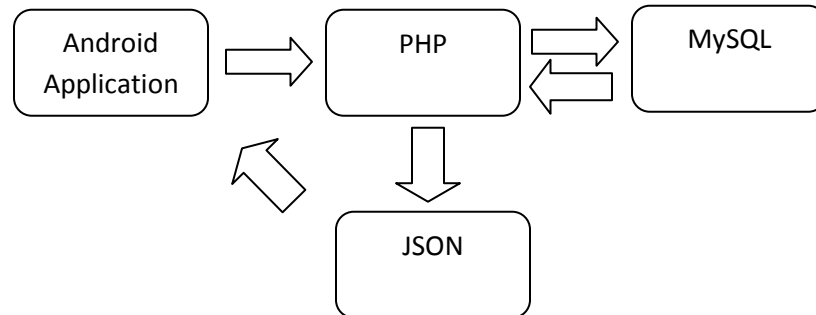


Figure 3. Process in communicating Android application and a remote MySQL database

As of PHP 5.2.0, the JSON extension is bundled and compiled into PHP by default.

```
$arr = array('a' => 1, 'b' => 2, 'c' => 3, 'd' => 4, 'e' => 5);
echo json_encode($arr);
```

Figure 4. Sample code snippet on converting arrays into JSON using PHP

PHP When executed, this will produce the following result:

```
{"a":1,"b":2,"c":3,"d":4,"e":5}
```

Figure 5. JSON result for PHP code in Figure 4

F. The Google Geocoding API

Geocoding is the process of converting addresses (like "1600 Amphitheatre Parkway, Mountain View, CA") into geographic coordinates (like latitude 37.423021 and longitude -122.083739), which you can use to place markers or position the map. [24]

A Geocoding API request must be of the following form:

<https://maps.googleapis.com/maps/api/geocode/output?parameters>

where output may be either of the following values:

- json (recommended) indicates output in JavaScript Object Notation (JSON)
- xml indicates output as XML

```
{
  "results" : [
    {
      "address_components" : [
        {
          "long_name" : "Manila",
          "short_name" : "Manila",
          "types" : [ "locality", "political" ]
        },
        {
          "long_name" : "Metro Manila",
          "short_name" : "NCR",
          "types" : [ "administrative_area_level_1", "political" ]
        },
        {
          "long_name" : "Philippines",
          "short_name" : "PH",
          "types" : [ "country", "political" ]
        }
      ],
      "formatted_address" : "Manila, Metro Manila, Philippines",
      "geometry" : {
        "bounds" : {
          "northeast" : {
            "lat" : 14.6390175,
            "lng" : 121.0261686
          },
          "southwest" : {
            "lat" : 14.5571256,
            "lng" : 120.9331339
          }
        },
        "location" : {
          "lat" : 14.5995124,
          "lng" : 120.9842195
        },
        "location_type" : "APPROXIMATE",
        "viewport" : {
          "northeast" : {
            "lat" : 14.6390175,
            "lng" : 121.0261686
          }
        }
      }
    }
  ]
}
```



```
    },
    "southwest" : {
      "lat" : 14.5571256,
      "lng" : 120.9331339
    }
  },
  "place_id" : "ChIJi8MeVwPKlzMRH8FpEHXV0Wk",
  "types" : [ "locality", "political" ]
},
],
"status" : "OK"
}
```

Figure 6. JSON output for

<https://maps.googleapis.com/maps/api/geocode/json?address=manila>

The values on lat and lng under “location” string will be used for Haversine computation. Whenever possible the value to be used will be based on location type result base on the following indicator:

- "ROOFTOP" restricts the results to addresses for which we have location information accurate down to street address precision.
- "RANGE_INTERPOLATED" restricts the results to those that reflect an approximation (usually on a road) interpolated between two precise points (such as intersections). An interpolated range generally indicates that rooftop geocodes are unavailable for a street address.
- "GEOMETRIC_CENTER" restricts the results to geometric centers of a location such as a polyline (for example, a street) or polygon (region).
- "APPROXIMATE" restricts the results to those that are characterized as approximate.

[24]

G. Haversine

The Haversine formula is an equation important in navigation, giving great-circle distances between two points on a sphere from their longitudes and latitudes. [25]

Haversine formula:

$$a = \sin^2(\Delta\phi/2) + \cos(\phi_1) * \cos(\phi_2) * \sin^2(\Delta\lambda/2)$$

$$c = 2 * \text{atan2}(\sqrt{a}, \sqrt{1-a})$$

$$d = R * c$$

note: where ϕ is latitude, λ is longitude, R is earth's radius (mean radius = 6,371km)

(that angles need to be in radians to pass to trig functions) [26]

By using the latitude and longitude value we get on Google Geocoding API and this formula we could determine the nearest veterinary clinic within the area.

IV. Design and Implementation

A. Context Diagram

Overall interaction of all external entities that have direct use and access to the application is shown in Figure 7. SpotAVet have five main role types namely: Clinic Owner, Veterinarian, Pet Owner, System Administrator, and the non-registered users or guests.

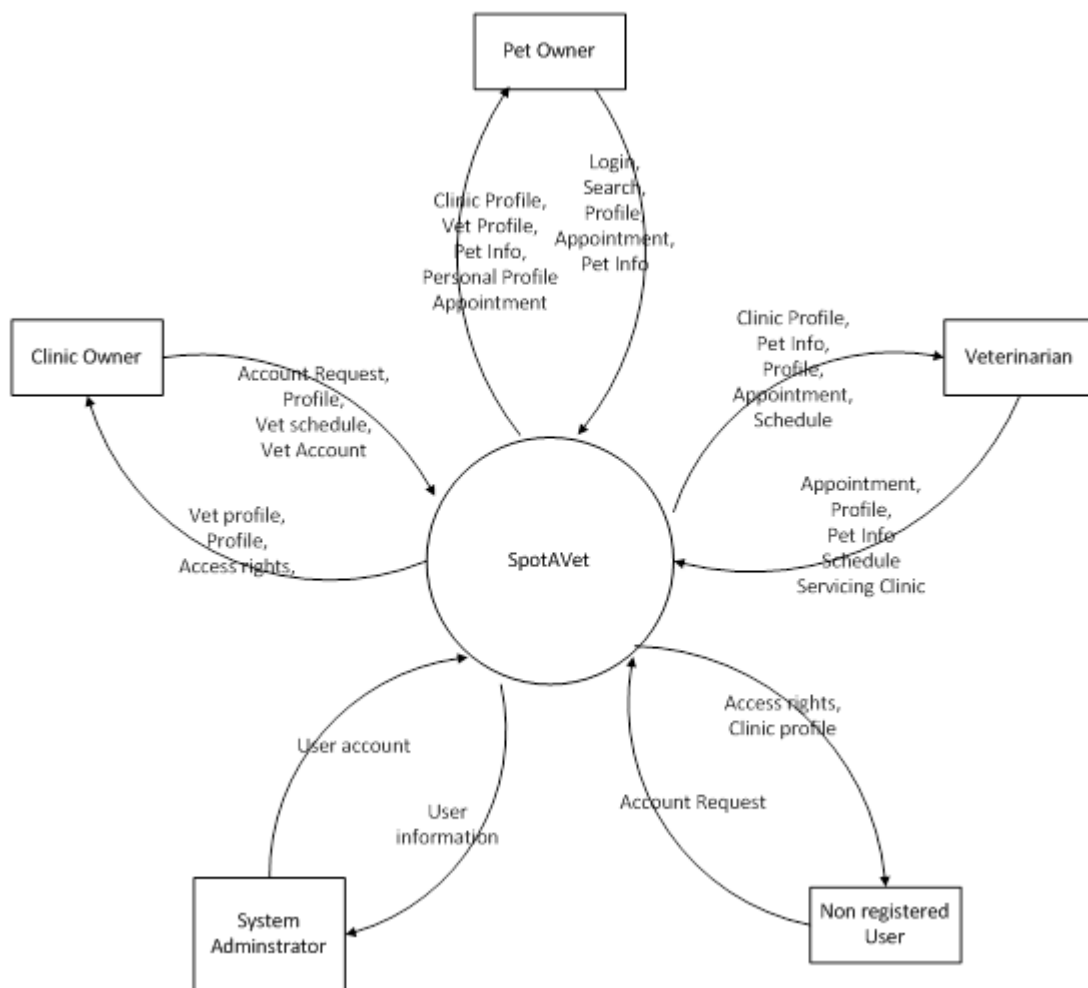


Figure 7. Context Diagram of SpotAVet

B. Data Flow Diagram

The top-level data flow diagram is shown in Figure 8. It represents the relationships between the various components and entities that are included in the system

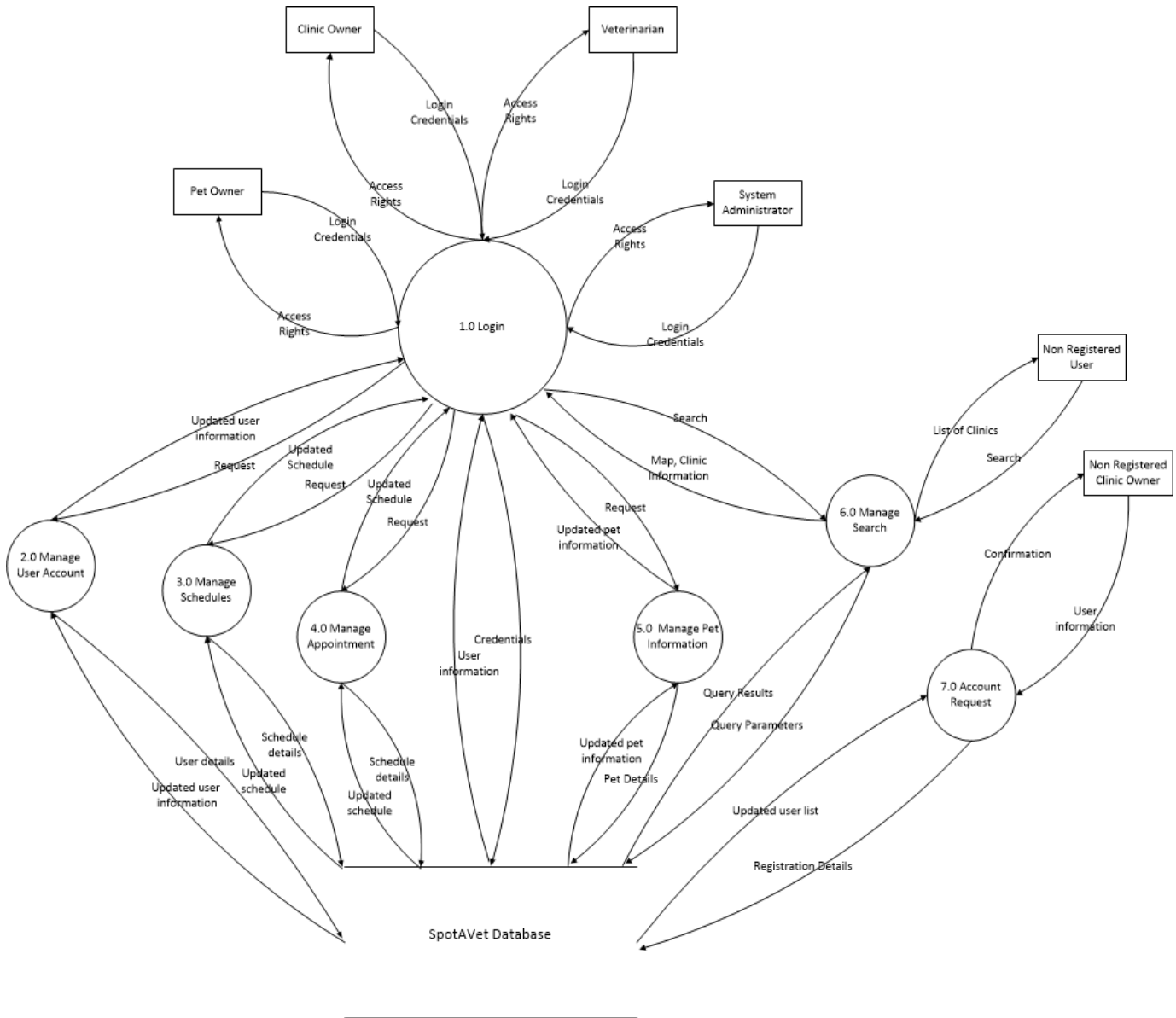


Figure 8. Top Level Data Flow Diagram of SpotAVet

All users who intend to use the system go through the login process (refer to Figure 9: Login Process). Credentials (username/password) assigned for each user are used during login process. It is then check if same matches the data from database. The system notifies the user for both successful and failed login attempts.

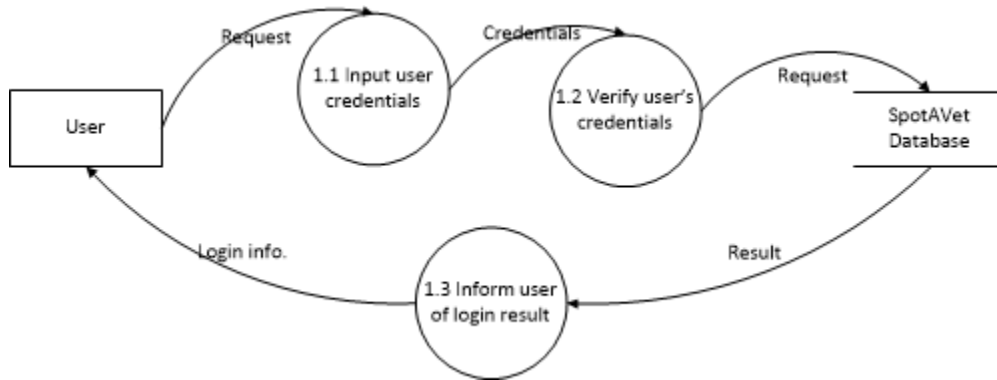


Figure 9. Sub-explosion Level 1 – 1.0 Login

System administrator has full control on all accounts used to access the application. Approval of new clinic account request, as well as activating and deactivating it are roles that the system admin is capable of.

Other users of the application (Clinic Secretary, Pet Owner and Veterinarian) are able to update information associated to them. Editing information is only allowed for fields that is not needed as unique (i.e., address). All changes on user information are subject to system verification.

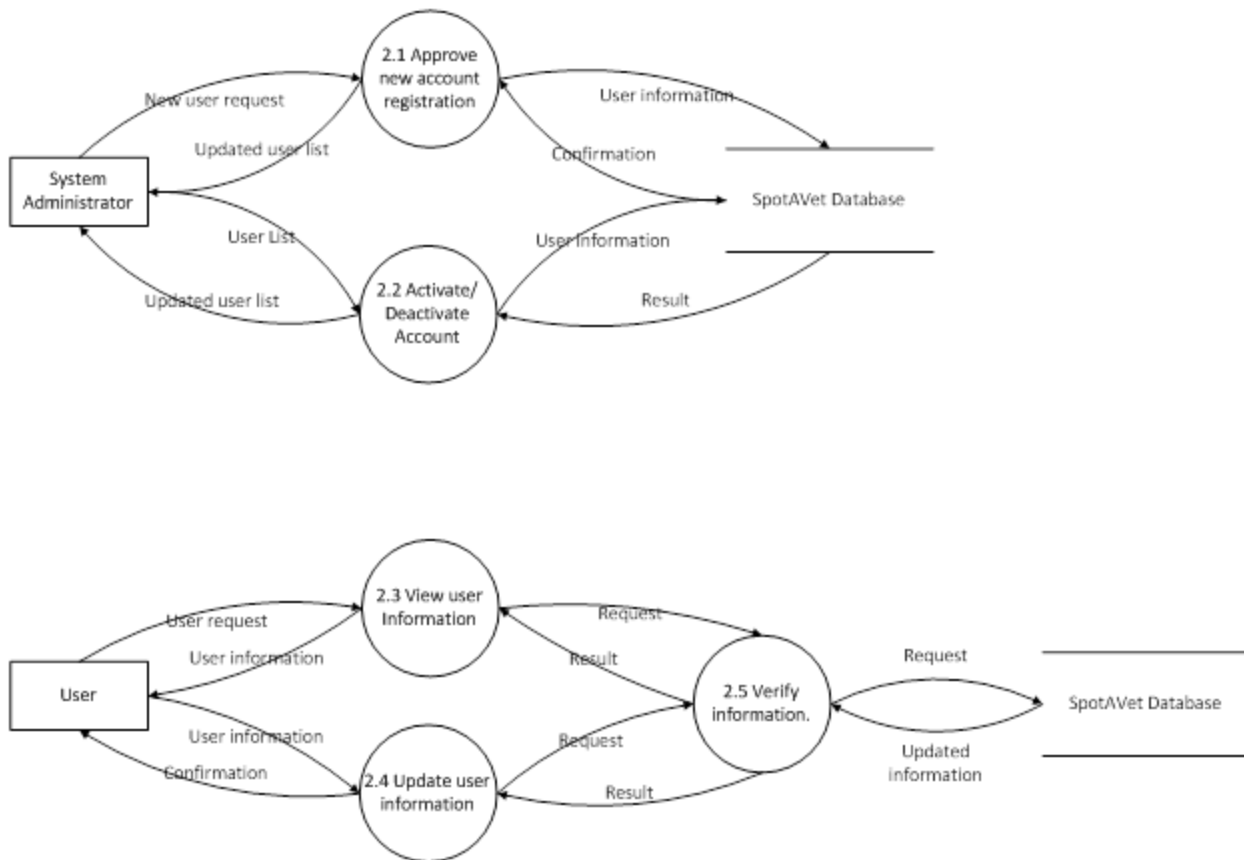


Figure 10. Sub-explosion Level 1 – 2.0 Manage User Account

Veterinarian’s schedule is controlled by the veterinarian itself. Veterinarians are only allowed to make changes on its own schedule per servicing clinic. This is applicable to clinic schedule or operating hours, only the clinic owner is allowed such change in schedule.

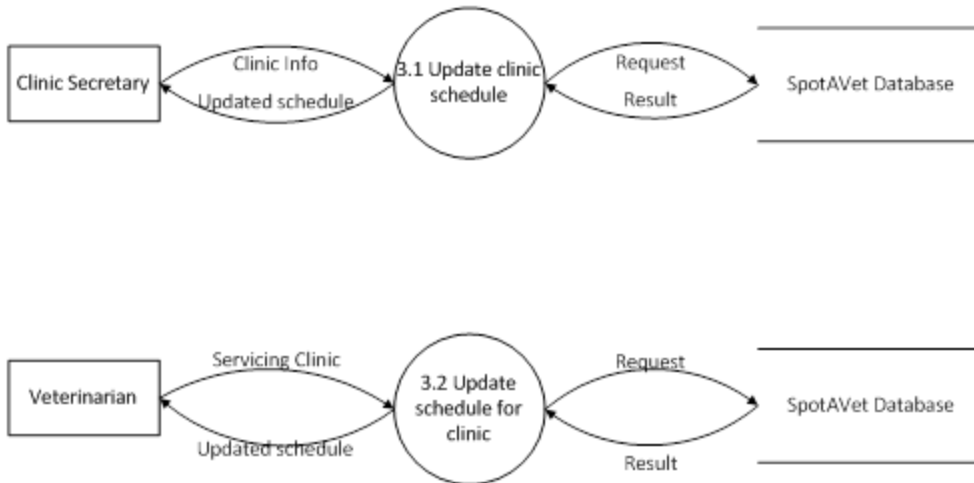


Figure 11. Sub-explosion Level 1 – 3.0 Manage Schedule

When the pet owner decides to make an appointment, it is required to do the search as first step (refer to Sub-explosion Level 1 of Manage Search). At the end of the search, pet owner chooses the veterinarian to set up schedule with, fill up all necessary information and make a request.

When the veterinarian received an appointment request, the vet reviews the request and decides whether to accept or decline the request. If a request is declined, the veterinarian needs to provide information on why the request is declined and the appointment is updated.

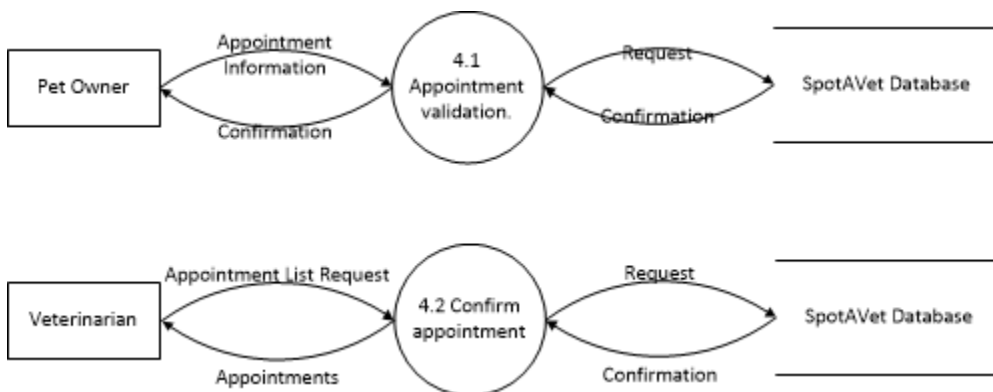


Figure 12. Sub-explosion Level 1 – 4.0 Manage Appointments

Before a pet owner can set appointment it is required to register the pet that will undergo examination/checkup. In the occurrence that pet information must be updated, pet owners have the right to update its pet information.

Every after examination, veterinarians are required to update the pet information for the result, this will serve as the medical history for the pet. In case of walk-in patient and unregistered owner, veterinarians can create account for the owner and initial profile for the pet, as well as the result of the consultation.

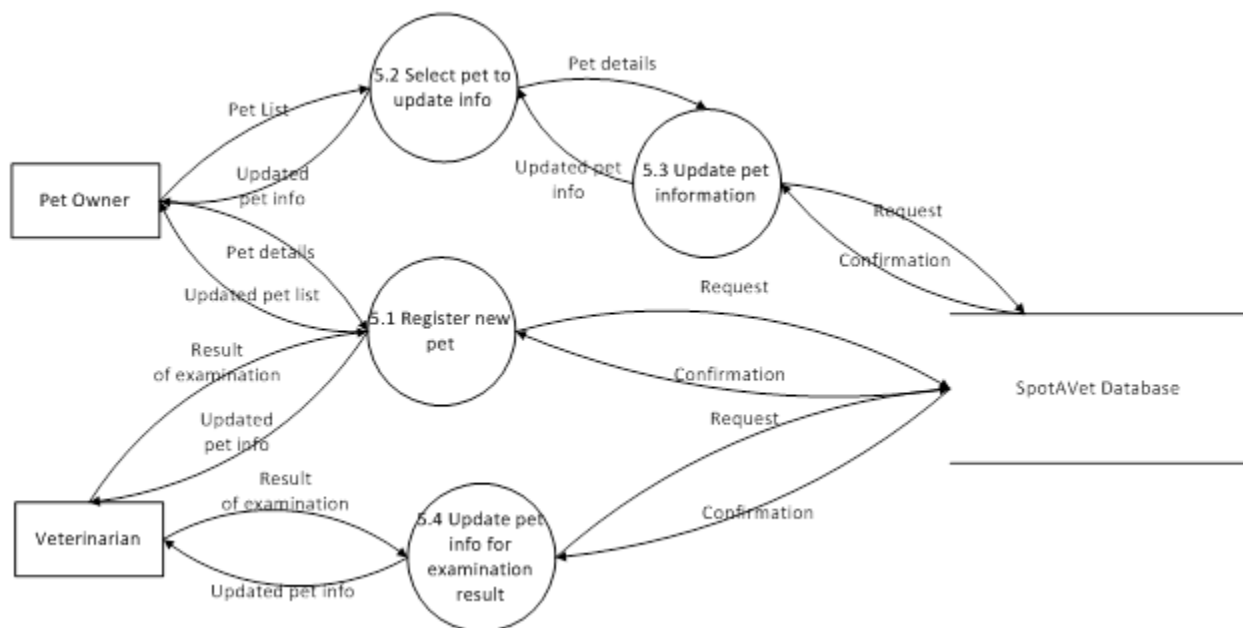


Figure 13. Sub-explosion Level 1 – 5.0 Manage Pet Information

Pet owner is allowed to filter search conditions such as veterinarian name and clinic name. This will then show list of clinic that have match on the set conditions. From the list, select the clinic and the list of veterinarian and schedule is shown. Select the veterinarian and from there you can set an appointment (refer to 4.0 Manage Appointments).

Non registered users can search clinic nearest to current location, but is not able to set appointment.

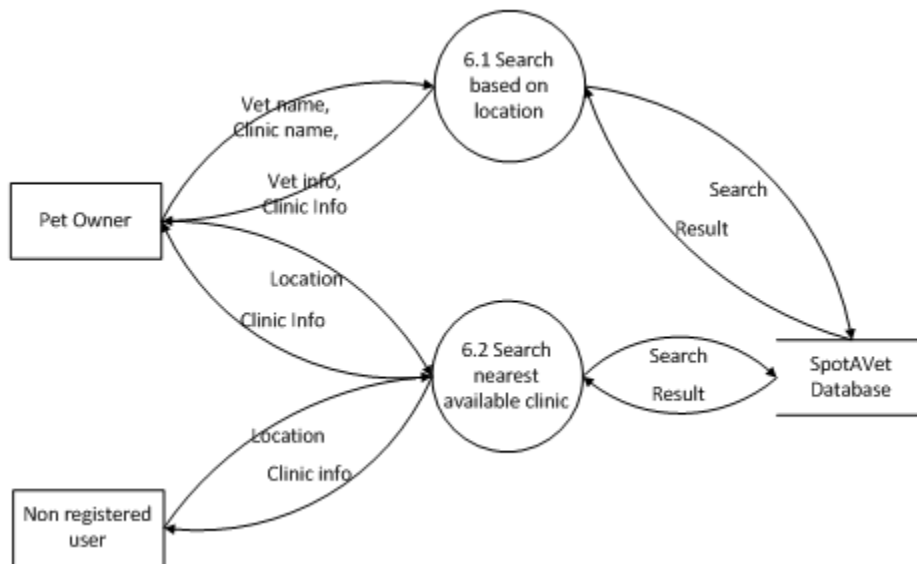


Figure 14. Sub-explosion Level 1 – 6.0 Manage Search

Non-registered clinic owners can fill up registration form with desired account information and then submit to the system. It is then queued to System Admin for approval (refer to 2.0 Manage User Account). Upon approval the account can already be used to access the application.



Figure 15. Sub-explosion Level 1 – 7.0 Account Request

C. Entity Relationship Diagram

The system is following the specific design for database shown in Figure 16.

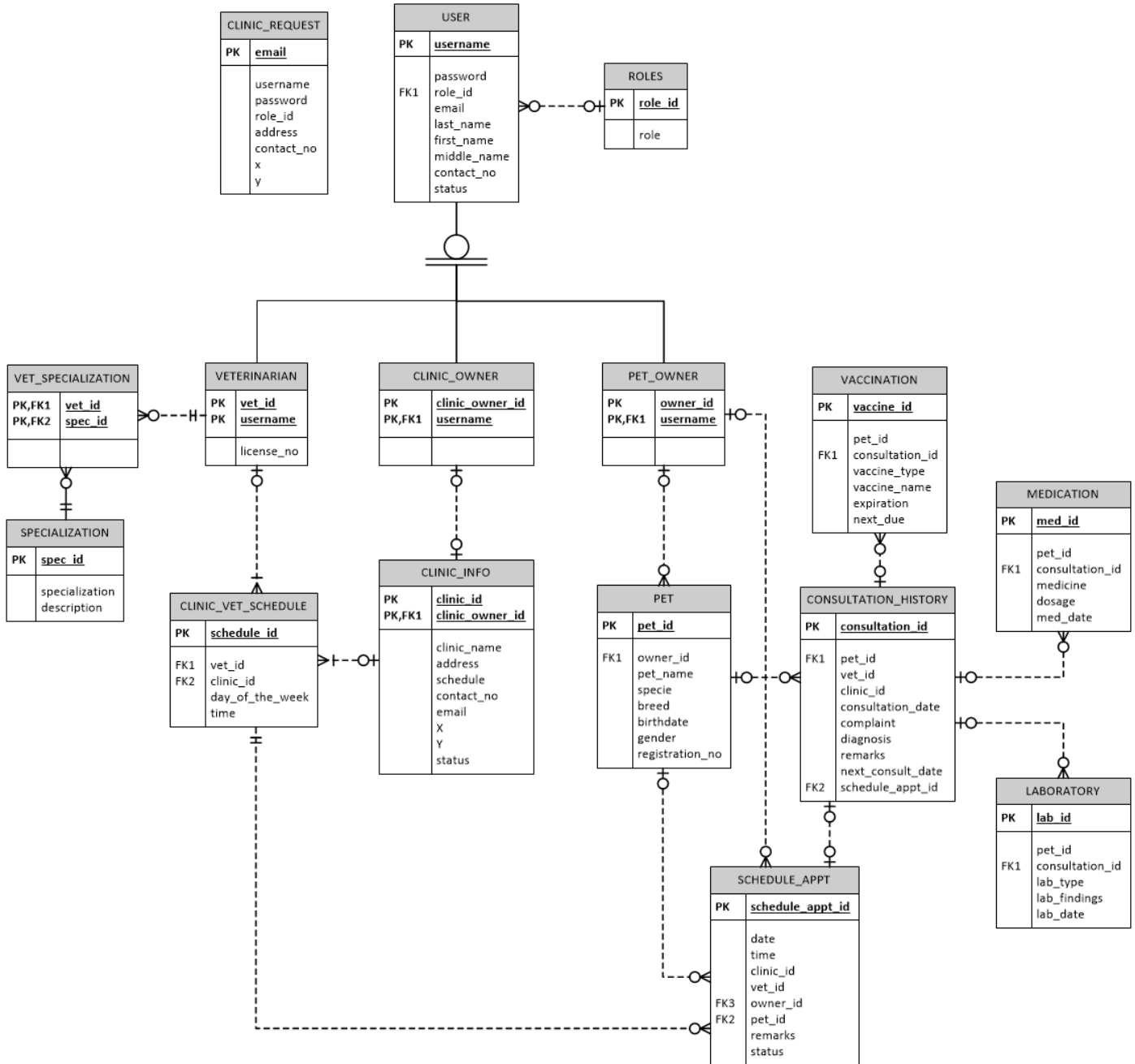


Figure 16. Entity Relationship Diagram of SpotAVet.

D. Database Table

The database consists of tables containing all information necessary for the system to function such as CLINIC_REQUEST, USER, ROLES, VETERINARIAN, CLINIC_OWNER, PET_OWNER, VET_SPECIALIZATION, SPECIALIZATION, CLINIC_INFO, CLINIC_VET_SCHEDULE, PET, SCHEDULE_APPT, CONSULTATION_HISTORY, MEDICATION, LABORATORY and VACCINATION. Primary Key = (pk) and Foreign Key = (fk).

Data Field	Data Type	Description
username	VARCHAR(50)	Username for system login.
password	VARCHAR(50)	Password for system login. Must be a minimum of six (6) alpha numeric characters.
email (pk)	VARCHAR(100)	Email ID for registration confirmation.
role_id	INTEGER(5)	Integer value that defines the type of user to be registered.
address	VARCHAR(200)	Clinic's street name. May include house no etc.
contact_no	VARCHAR(50)	Clinic's contact number(s).
x	FLOAT(11, 7)	X – longitude. Clinic's address location on map.
y	FLOAT(11, 7)	Y – latitude. Clinic's address location on map.

TABLE 1: CLINIC_REQUEST contains new clinic information that requests access to the system

Data Field	Data Type	Description
username (pk)	VARCHAR(50)	Username for system login.
role_id	INTEGER(10)	1 – ADMINISTRATOR, 2 – CLINIC OWNER, 3 – VETERINARIAN, 4 - PET OWNER
password	VARCHAR(50)	Password for system login. Must be a minimum of six (6) alpha numeric characters.
last_name	VARCHAR(50)	User's last name.
first_name	VARCHAR(100)	User's first name.
middle_name	VARCHAR(50)	User's middle name.
contact_no	VARCHAR(50)	User's contact number.
email	VARCHAR(100)	Email ID used upon registration.
status	INTEGER(1)	0 – non active users, 1 – active users, 2 – pending

approval

TABLE 2: USER contains all registered users of the system

Data Field	Data Type	Description
role_id	INTEGER(5)	Role identification number
role	VARCHAR(20)	Roles per user in the system. 1 – ADMINISTRATOR, 2 – CLINIC OWNER, 3 – VETERINARIAN, 4 - PET OWNER

TABLE 3: ROLES contains all registered users of the system

Data Field	Data Type	Description
vet_id (pk)	INTEGER(5)	Veterinarian's Identification number (system generated)
username (pk, fk)	VARCHAR(50)	Username associated with the client.
license_no	VARCHAR(50)	Veterinarian's license number.

TABLE 4: VETERINARIAN contains all registered users with type VETERINARIAN in the system

Data Field	Data Type	Description
clinic_owner_id (pk)	INTEGER(10)	Clinic Owner's Identification number (system generated)
username (pk, fk)	VARCHAR(20)	Username associated with the client.

TABLE 5: CLINIC_OWNER contains all information about clinic owner

Data Field	Data Type	Description
owner_id (pk)	INTEGER(10)	Pet owner Identification number (system generated)
username (pk, fk)	VARCHAR(50)	Username associated with the client.

TABLE 6: PET_OWNER contains all registered user information with type PET OWNER in the system

Data Field	Data Type	Description
vet_id (pk, fk)	INTEGER(5)	Veterinarian's Identification number (system generated)
spec_id (pk, fk)	INTEGER(5)	Specialization identification number.

TABLE 7: VET_SPECIALIZATION contains details of all VETERINARIAN specialization

Data Field	Data Type	Description
spec_id (pk)	INTEGER(5)	Veterinarian's Identification number (system generated)
specialization	VARCHAR(50)	Types of veterinarian's specialization.
description	VARCHAR(50)	Description for each specialization.

TABLE 8: SPECIALIZATION contains details of all SPECIALIZATION available on veterinary medicine

Data Field	Data Type	Description
clinic_owner_id (pk, fk)	INTEGER(10)	Clinic secretary Identification Number (system generated)
clinic_id (pk)	INTEGER(10)	Clinic Identification Number (system generated)
clinic_name	VARCHAR(100)	Clinic's name or company name.
address	VARCHAR(200)	Clinic's complete address.
contact_no	VARCHAR(50)	Clinic's contact number(s).
X	FLOAT	X – longitude. Clinic's address location on map.
Y	FLOAT	Y – latitude. Clinic's address location on map.

TABLE 9: CLINIC_INFO contains all information about the registered clinics

Data Field	Data Type	Description
schedule_id (pk)	INTEGER(10)	Schedule ID
vet_id (fk)	INTEGER(5)	Veterinarian's Identification number (system generated)
clinic_id (fk)	INTEGER(5)	Clinic Identification Number (system generated)
schedule	VARCHAR(200)	Complete schedule of veterinarian for each clinic

TABLE 10: CLINIC_VET_SCHEDULE contains schedule of all veterinarian in different clinic.

Data Field	Data Type	Description
pet_id (pk)	INTEGER(5)	Pet's ID.
owner_id (fk)	INTEGER(5)	Pet owner's ID.
registration_no	VARCHAR(50)	Registration number of the pet.
specie	VARCHAR(50)	Type of pet.

breed	VARCHAR(50)	Specific breed of pet.
name	VARCHAR(50)	Name of the pet.
age	INTEGER(5)	Age of the pet.
gender	VARCHAR(20)	Pet's gender.
birthdate	VARCHAR(50)	Pet's birthday (if applicable)

TABLE 11: PET contains all necessary pet information.

Data Field	Data Type	Description
schedule_appt_id (pk)	INTEGER(5)	Schedule ID.
date	VARCHAR(50)	Date of consultation/schedule.
time	VARCHAR(50)	Time of consultation/schedule.
clinic_id (fk)	INTEGER(5)	Clinic's ID.
vet_id (fk)	INTEGER(5)	Veterinarian's ID.
owner_id	INTEGER(5)	Pet owner's ID.
pet_id	INTEGER(5)	Pet's ID
remarks	VARCHAR(200)	States the reason for visit and reason for disapproved request.
confirmed	INTEGER(5)	Status of request: 1 – Approved, 0 – Disapproved.

TABLE 12: SCHEDULE_APPT contains requested and approved schedule of pet owner in different clinic.

Data Field	Data Type	Description
consultation_id (pk)	INTEGER(5)	Consultation ID.
pet_id (fk)	INTEGER(5)	Pet's ID.
consultation_date	VARCHAR(50)	Date the consultation was done.
diagnosis	INTEGER(5)	Diagnosis for the pet.
remarks	VARCHAR(50)	Other remarks for the pet consultation result.
next_consult_date	VARCHAR(50)	Advised next consultation date for the pet(if necessary)
schedule_appt_id (fk)	INTEGER(5)	Reference to schedule ID.

TABLE 13: CONSULTATION_HISTORY contains all previous consultation record of pet.

Data Field	Data Type	Description
med_id (pk)	INTEGER(5)	Medication control number.
consultation_id (fk)	INTEGER(5)	Reference number from CONSULTATION.
pet_id	INTEGER(5)	Reference for pet receiving the medication.

medicine	VARCHAR(50)	Medicine information.
dosage	VARCHAR(50)	Dosage of medicine.
date	VARCHAR(50)	Date the medication was prescribed.

TABLE 14: MEDICATION contains information for necessary medication for the pet.

Data Field	Data Type	Description
lab_id (pk)	INTEGER(5)	Laboratory control number.
consultation_id (fk)	INTEGER(5)	Reference number from CONSULTATION.
pet_id	INTEGER(5)	Reference for pet undergone laboratory examination.
lab_type	VARCHAR(50)	Type of laboratory examination performed.
lab_findings	VARCHAR(200)	Laboratory findings.
lab_date	VARCHAR(50)	Date the laboratory was performed.

TABLE 15: LABORATORY contains information for laboratory results for the pet.

Data Field	Data Type	Description
vaccine_id (pk)	INTEGER(5)	Vaccine id
consultation_id (fk)	INTEGER(5)	Reference number from CONSULTATION.
pet_id	INTEGER(5)	Reference for pet undergone vaccination.
vaccine_type	VARCHAR(50)	Type of vaccination e.g. for rabies, heartworm
vaccine_name	VARCHAR(200)	Name of vaccine
expiration	DATE	Expiration of vaccine
next_due	DATE	Next due date of vaccine

TABLE 16: VACCINATION contains information for vaccination info for the pet.

E. Technical Architecture

The system is implemented using various technologies available today. The following are list of technologies used in development of this system:

1. Programming Language:
 - a. PHP 5.5.11
 - b. jQuery 1.11.0

- c. Java
2. Web Server:
- a. XAMPP 1.8.3
 - b. Apache 2.4.7 or later
 - c. JSON
3. Database:
- a. MySQL 5.0.11 or later

The system is available on Web and Android devices with OS v4.x, and requires internet connection. The system is working and tested using Google Chrome as web browser.

V. Results

A. Web-based application: User Interface

The home page of SpotAVet is shown in Figure 17. In this page, clinic owners may choose to register their clinic to the system, and registered users may choose to login. Initially, only clinic owners could request for a user account. It is then approved by the system administrator.

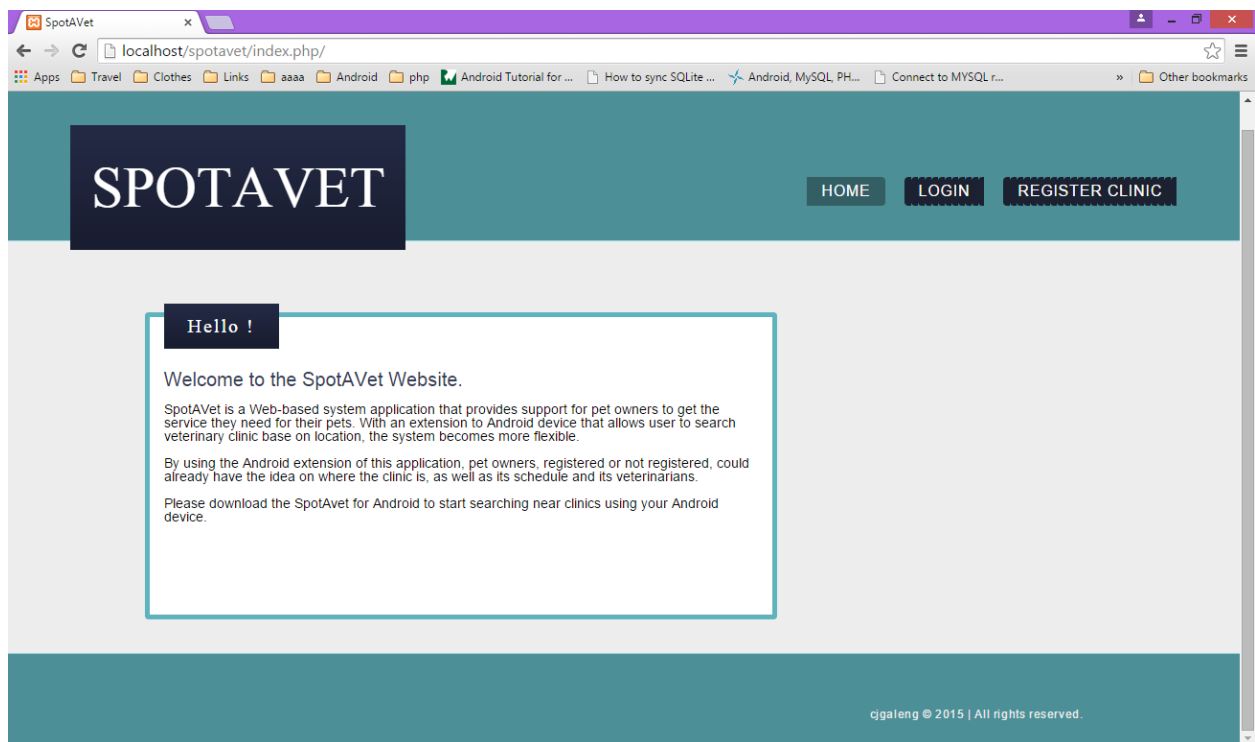


Figure 17. SpotAVet homepage

Upon successful login to the system, users are directed to their specific page based on their roles.

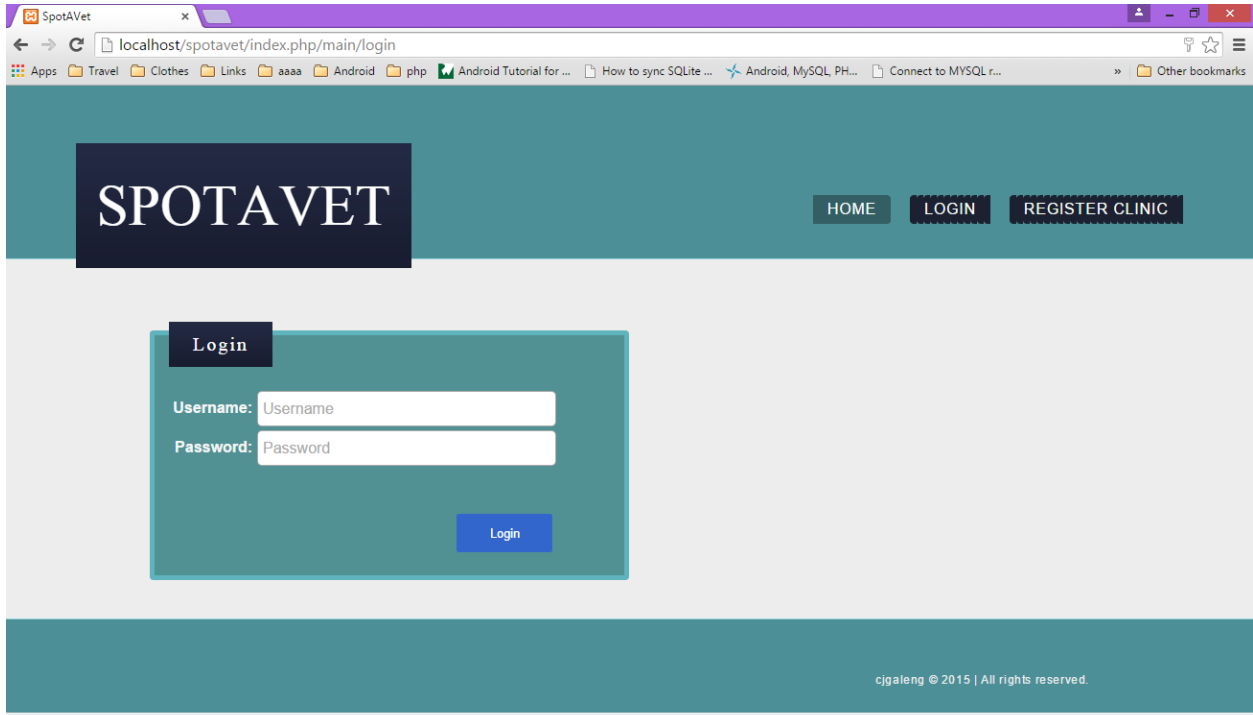


Figure 18. SpotAVet login page

If the user is an administrator, the page for administrator is displayed, as shown in Fig 19

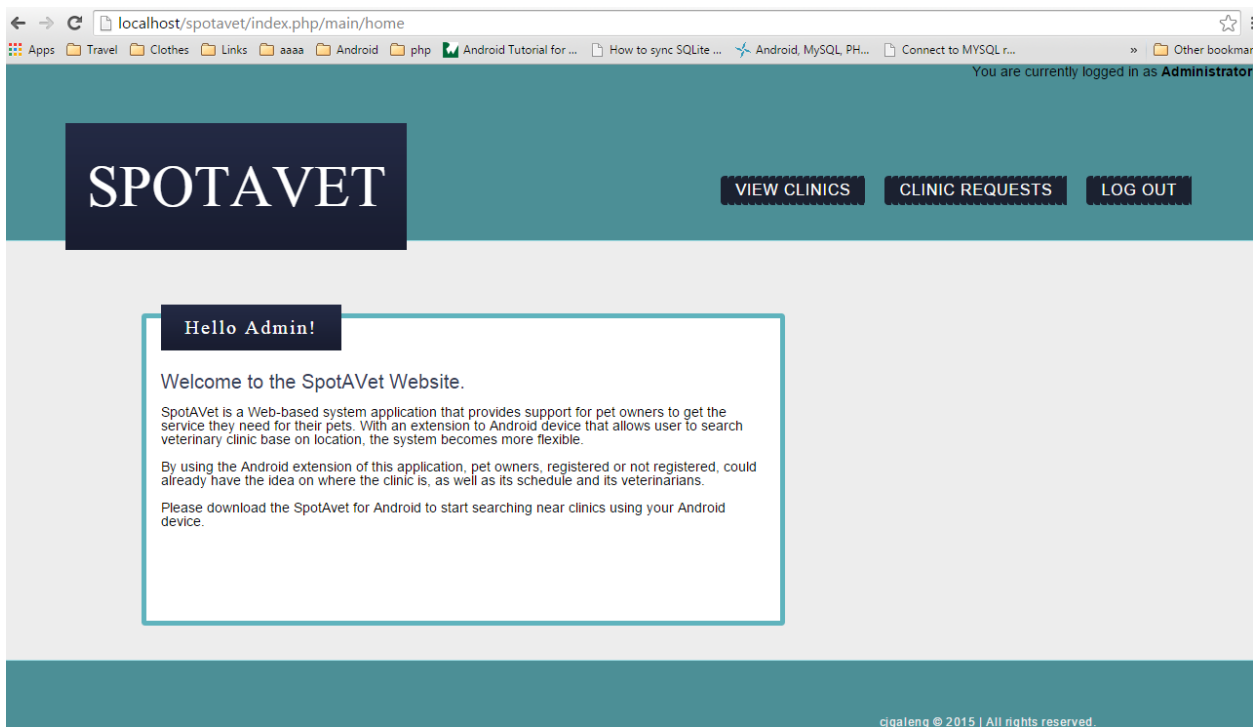



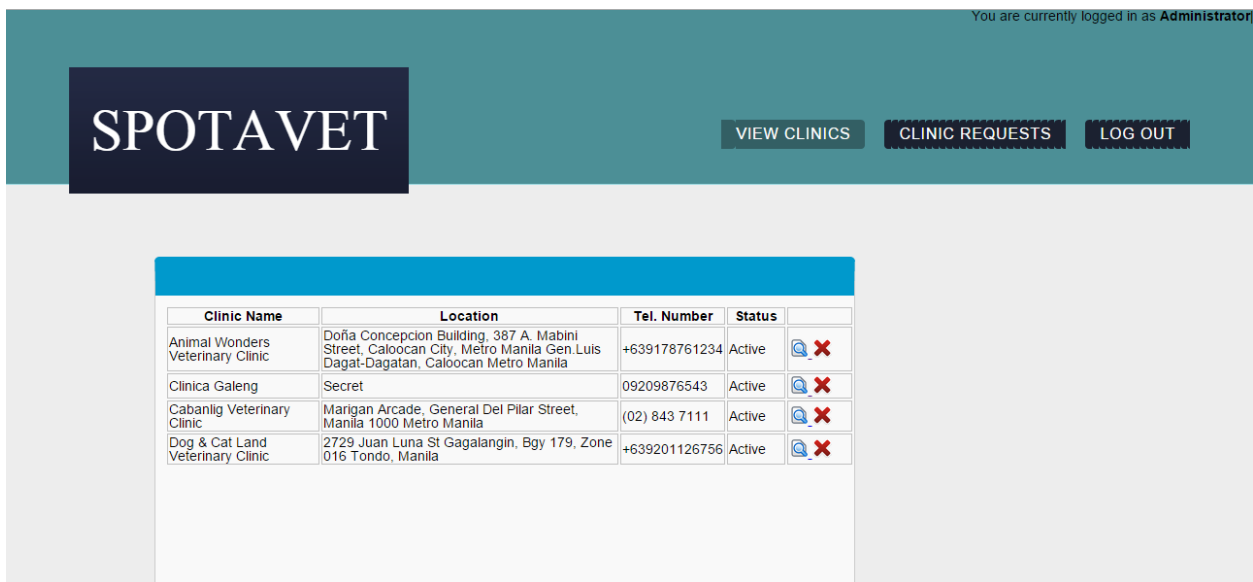


Figure 19. Administrator homepage

Administrator is responsible for approval, activation and deactivation of clinic account.

By Viewing Clinic account the activation and deactivation menu will be available.

In this page, if  is clicked, clinic information is displayed. If either  or  is clicked, the clinic account is activated or deactivated respectively. See Figure 20 as reference.



The screenshot shows the SPOTAVET Administrator interface. At the top right, it says "You are currently logged in as Administrator". The main header features the SPOTAVET logo and three buttons: "VIEW CLINICS", "CLINIC REQUESTS", and "LOG OUT". Below the header is a table with the following data:











Clinic Name	Location	Tel. Number	Status	
Animal Wonders Veterinary Clinic	Dofia Concepcion Building, 387 A. Mabini Street, Caloocan City, Metro Manila Gen.Luis Dagat-Dagatan, Caloocan Metro Manila	+639178761234	Active	 
Clinica Galeng	Secret	09209876543	Active	 
Cabanlig Veterinary Clinic	Marigan Arcade, General Del Pilar Street, Manila 1000 Metro Manila	(02) 843 7111	Active	 
Dog & Cat Land Veterinary Clinic	2729 Juan Luna St Gagalangin, Bgy 179, Zone 016 Tondo, Manila	+639201126756	Active	 

Figure 20. Administrator – View Clinics

When Clinic Requests button is clicked, list of clinics requesting account activation are shown. See Figure 21. Clicking  or  icon will approve or disapprove the request respectively.

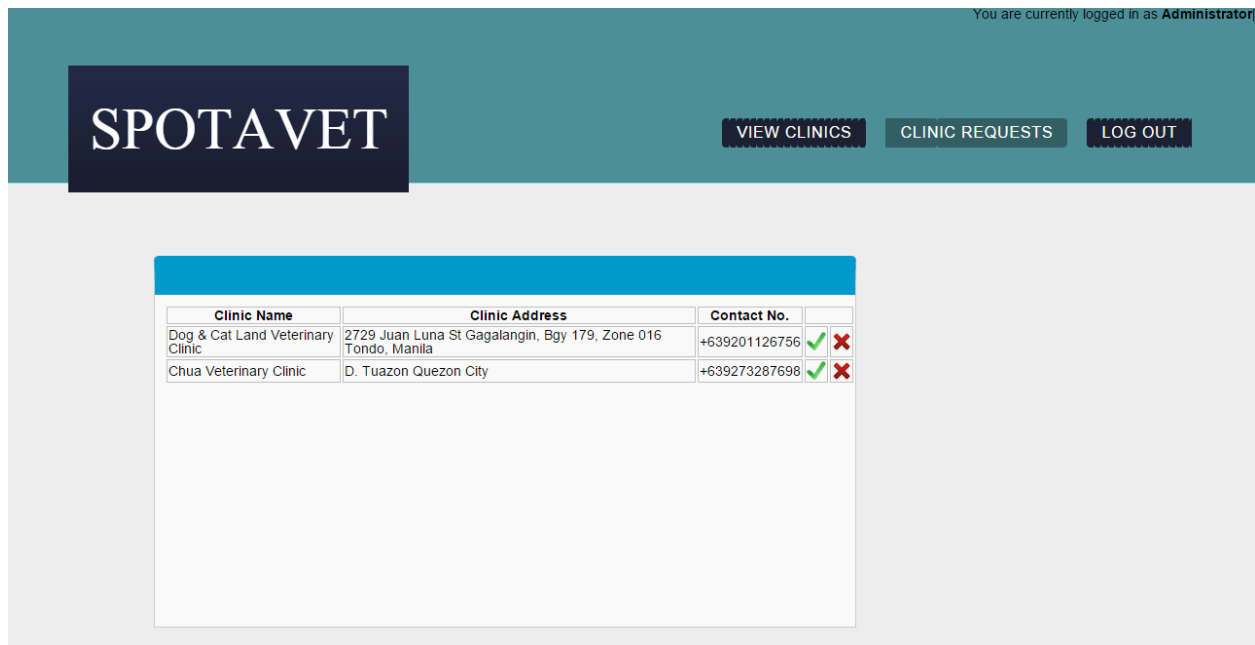


Figure 21. Administrator – Clinic Requests page.

If the user is a Clinic Owner, the Clinic Owner homepage is displayed, as shown in Figure 22

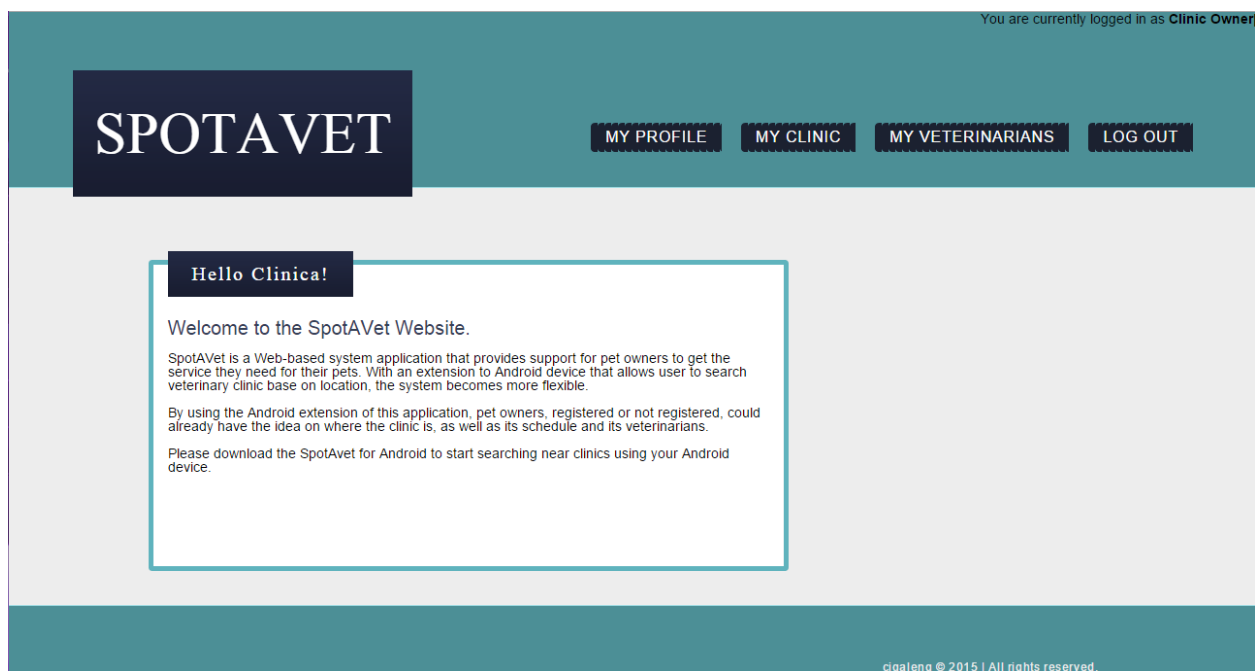


Figure 22. Clinic Owner Homepage

Clinic owners could navigate from personal profile, clinic profile and veterinarians within the clinic. The clinic owner is the one responsible in creating account for every veterinarian working at the clinic.

If the user role is a Veterinarian the vet homepage is displayed as shown in Figure 23.

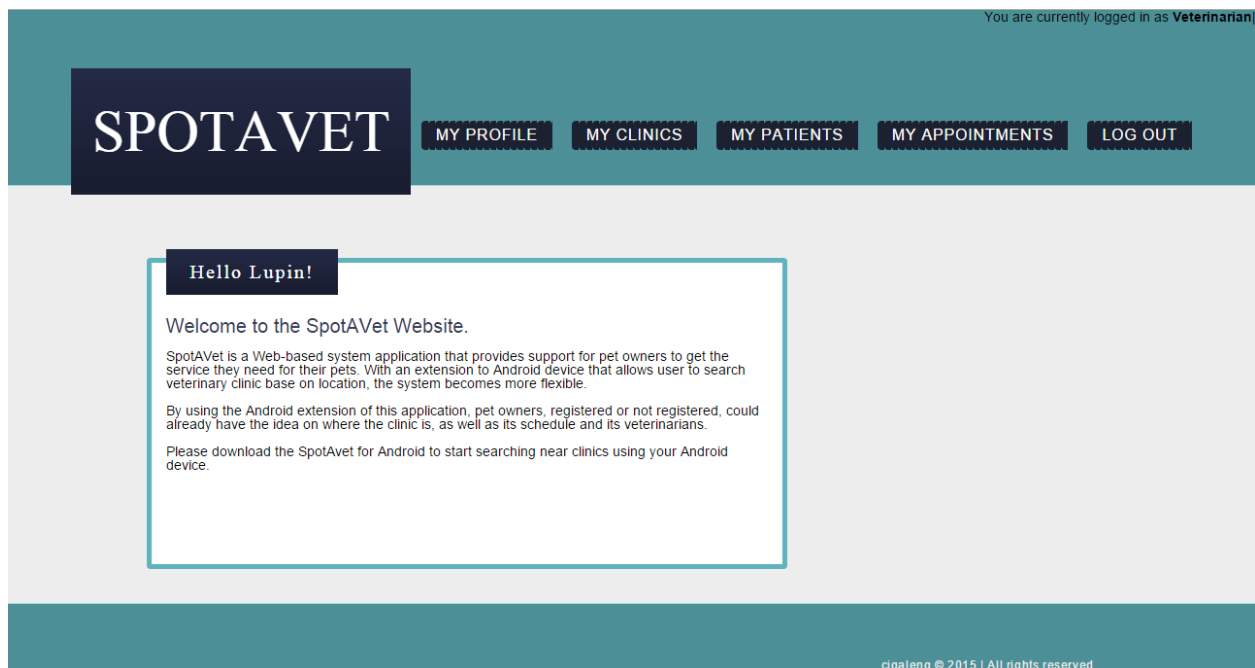


Figure 23. Veterinarian homepage

Just like the clinic owner, Veterinarian could update personal profile and view clinic information, maintains medical record for his/her patient and approve appointment request. Also, Veterinarians are the one responsible in creating account for pet owners and their pets having consultation for the first time. Both pet owners and pet details are added being when creating

account for pet owners. Appointment request coming from registered users can be viewed, approve and decline by the veterinarian.

Lastly, if the user is a Pet Owner, they will be able to see their homepage as shown in Figure 24.

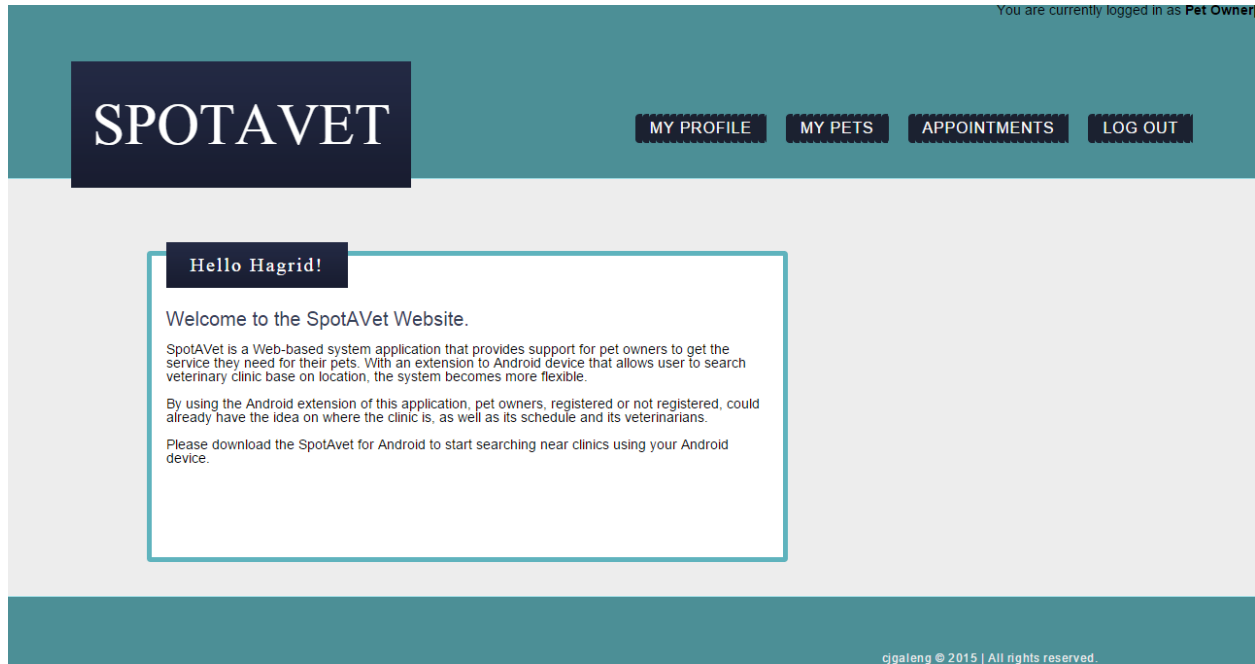


Figure 24. Pet owner homepage

View and update of own profile and pet information are the main functionality the owner can do. Once account as pet owner was created, scheduling appointment via web will now be available for use.

B. Web-based application: System Functionality

Only clinic owners can apply for account for the system. By using the form shown in Figure 25, any clinic owner could request for account to the system.

Registration Form


Clinic Owner Information	
Register As:	Clinic Owner
*Username:	Username
*Password:	Password
*Confirm Password:	Confirm Password
*Email:	Email
*Last Name:	Last Name
*First Name:	First Name
*Middle Name:	Middle Name
*Contact #:	Contact
Clinic Information	
*Clinic Name:	Clinic Name
*Clinic Address:	Clinic Address
*Operating Hours:	(Mon ~ Fri :: 8:00 AM ~ 5:00 PM)
Clinic Location Details	
*San Guillermo Street, Ciudad	
	
Lat: 14.688242	Lng: 120.9789202
<input type="button" value="Register!"/>	

Figure 25. Clinic Request Form

It will then need to be approved by system administrator. Once approved, veterinarian account could now be created by the registered clinic. Only registered clinic can create new veterinarian account with the use of the form shown in Figure 26. Accounts created for veterinarians are pre-approved by the system and can already be used. It is advised to change password after initial login to the system. Any user can change password by editing his/her own profile.

The screenshot displays a web browser window with the URL `localhost/spotavet/index.php/main/add_veterinarian`. The page features a teal header with the **SPOTAVET** logo on the left and navigation buttons for **MY PROFILE**, **MY CLINIC**, **MY VETERINARIANS**, and **LOG OUT** on the right. The main content area contains a form titled "Veterinarian Info" with a blue header bar. The form includes two radio buttons: "Search from existing veterinarian" (unselected) and "Add new veterinarian account" (selected). Below this is a "Register As:" dropdown menu set to "Veterinarian". The form contains several required text input fields: *Username, *Password, *Confirm Password, *Email Address, *Last Name, *First Name, *Middle Name, *Contact No., and *License No. A section for animal types includes checkboxes for Canine, Avian, Equine, Feline, Reptiles/Amphibians, and Lagomorph. At the bottom, there is a *Schedule field and a blue "Add Vet" button.

Figure 26. Add New Veterinarian Account form

In cases where the veterinarian has already an account and would only join a new clinic to offer their services, clinic owner could search veterinary name or email, set up the schedule and add the vet on clinic's vet list. See Figure 27.

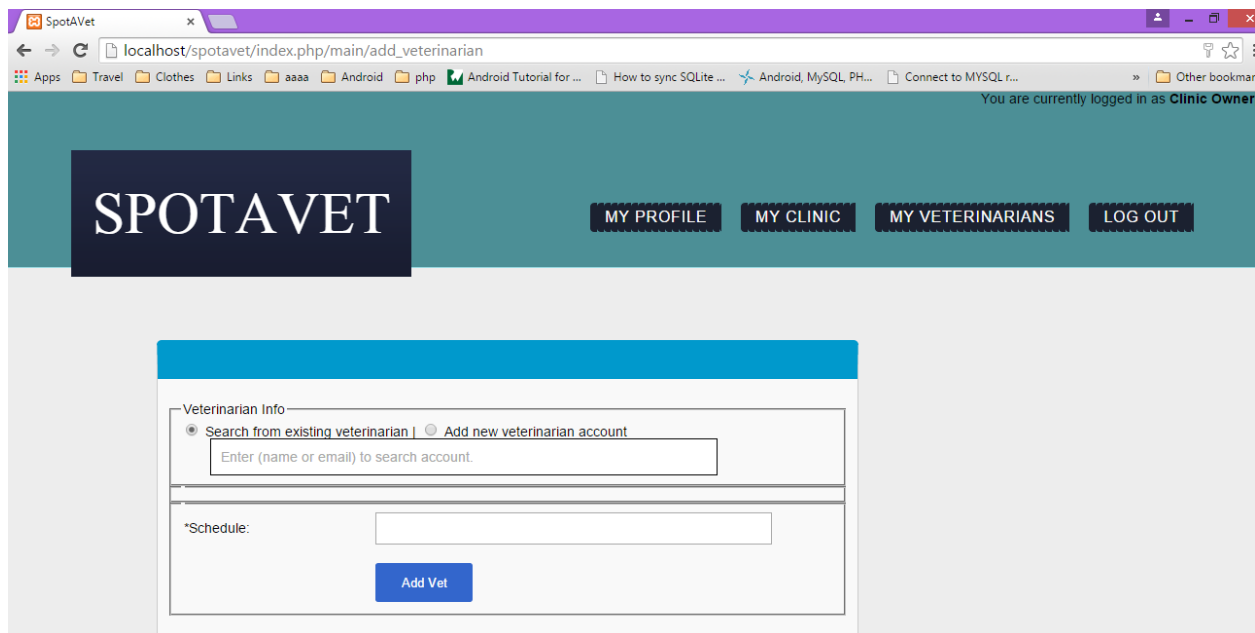


Figure 27. Add Veterinarian from list of existing veterinarians

As stated, veterinarian account is a pre-approved account, thus after successful registration, veterinarian could already use the system. As a veterinarian, they are responsible in maintaining the pet medical record for their patients. Every after consultation, veterinarians will add the result of consultation to the system, result of laboratory if there is any and medication details as well as vaccination history. In cases where the client/pet owner does not yet have the

account for the system veterinarians can create account for owner and profile for the pet. Refer to Figure 28 and Figure 19 respectively;

The screenshot shows the SPOTAVET web application interface. At the top, there is a navigation bar with the logo 'SPOTAVET' and four menu items: 'MY PROFILE', 'MY CLINICS', 'MY PATIENTS', and 'MY APPOINTMENTS'. Below the navigation bar, the main content area features a form titled 'New Patient Record'. The form has two radio buttons: 'Select from registered pet list |' (which is selected) and 'Add new patient's owner account'. Below these buttons is a text input field containing the name 'fluffy'. Underneath the name field is a dropdown menu labeled '*Clinic:' with 'Animal Wonders Veterinary Clinic' selected. The 'Consultation Details' section contains several input fields: '*Consultation Date:' with a placeholder 'mm/dd/yyyy', '*Chief Complaints:', '*Diagnosis/Findings:', 'Other Remarks:', and 'Next Consultation Date:' with a placeholder 'mm/dd/yyyy'. At the bottom of the form, there are four blue buttons: 'Medication', 'Vaccination', 'Laboratory', and 'Submit'.

Figure 28. Adding Consultation Details for existing patients

New Patient Record

Select from registered pet list | Add new patient's owner account

Pet Owner Information

Register As:

*Username:

*Password:

*Confirm Password:

*Email Address:

*Last Name:

*First Name:

*Middle Name:

*Contact No:

Pet Information

*Pet Name:

*Specie:

*Breed:

*Birth Date:

*Gender:

Registration No.:

*Clinic:

Consultation Details

*Consultation Date:

*Chief Complaints:

*Diagnosis/Findings:

Other Remarks:

Next Consultation Date:

[Medication](#)[Vaccination](#)[Laboratory](#)[Submit](#)

Figure 29. Add new patient form

Pet owners that already have access to the system should register all his pets to the system for the veterinarian. Pet owners are responsible on updating personal info as well as pet information. When adding and updating the pet information, the form to be used is shown in Figure 30.

The screenshot shows a web browser window with the URL localhost/spotavet/index.php/main/add_pet. The page features a teal header with the 'SPOTAVET' logo and navigation buttons for 'MY PROFILE', 'MY PETS', 'APPOINTMENTS', and 'LOG OUT'. The main content area contains a white form titled 'Pet Information' with the following fields:

- *Name:
- *Specie:
- *Breed:
- *Birth date:
- *Gender:
- Registration No.:

A blue 'Save' button is located at the bottom of the form.

Figure 30. Add Pet form.

Pet owners can also utilize the set appointment feature of the system. This will set up appointment schedule for the desired clinic or veterinarian. Only the veterinarian chosen from the selection will see the details of the request. See Figure 31 to have an idea how appointment is requested.

The screenshot shows a web browser window with the URL `localhost/spotavet/index.php/main/appointment_request`. The page features a teal header with the **SPOTAVET** logo on the left and navigation buttons for **MY PROFILE**, **MY PETS**, **APPOINTMENTS**, and **LOG OUT** on the right. A status message at the top right indicates the user is logged in as a **Pet Owner**. The main content area contains an **APPOINTMENT REQUEST** form with the following fields:

- A search bar with the placeholder text "Enter name to begin search." and a dropdown menu labeled "Clinic".
- A dropdown menu labeled "VETERINARIAN".
- Input fields for "PET NAME" (containing "fluffy"), "DATE" (containing "06/15/2015"), and "TIME" (containing "01:44:02 PM").
- A text area for "REASON FOR VISIT" with the placeholder text "State reason for appointment."
- A blue "Submit Request" button.

Figure 31. Request appointment form

C. Android-based application: Functionality

One of the unique features of mobile application is location awareness. SpotAVet for android starts with the current location of the user. The search could be triggered at start to check if any registered clinic is nearby. If none, you could either zoom out to widen the search perimeter or touch in the map the location you want to start the search. See Figure 32 for reference on how the UI looks like.

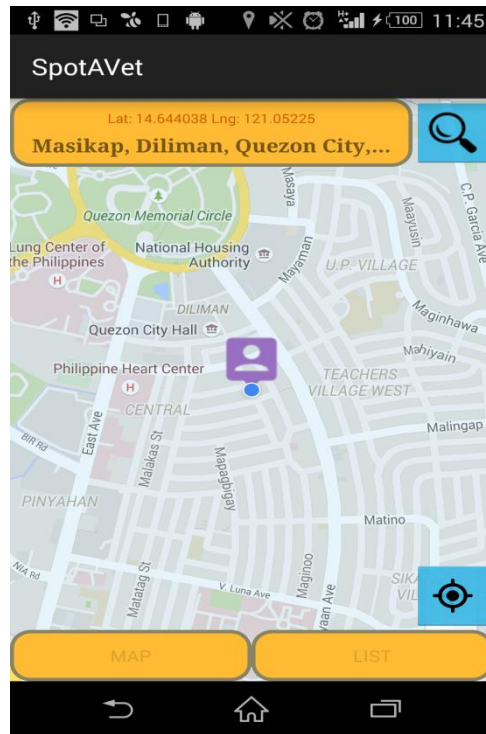


Figure 32. SpotAVet android

The application would notify the user if a clinic has been found or not. By clicking the marker on the map, the name of the clinic would pop up. See figure 33 for reference.

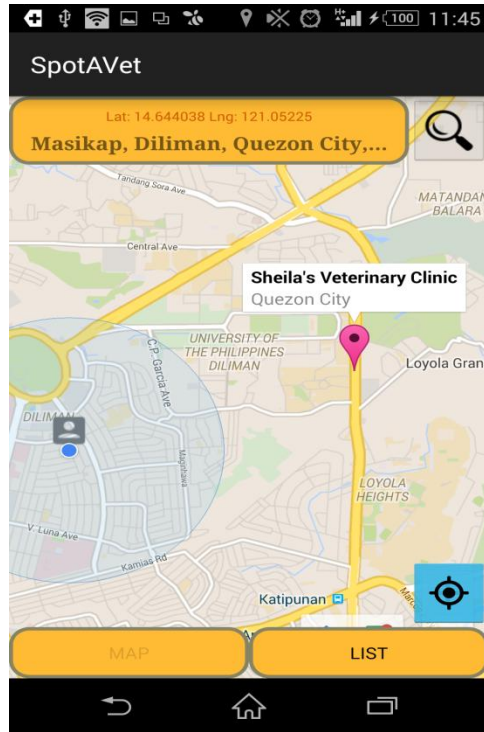


Figure 33. SpotAVet Android: Clinic found

Tapping on the name of the clinic would give you the complete information about the clinic. If no list of veterinarian is shown it means that the clinic has not added any veterinarians on clinic profile. Tapping on the name of a veterinarian on list would give the user the complete profile of the vet. See Figure 48 and Figure 49 for example.

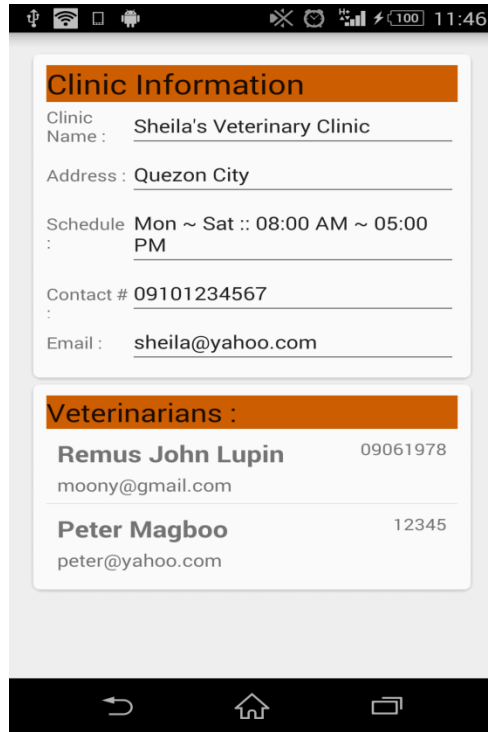


Figure 34. SpotAVet android: Clinic Information

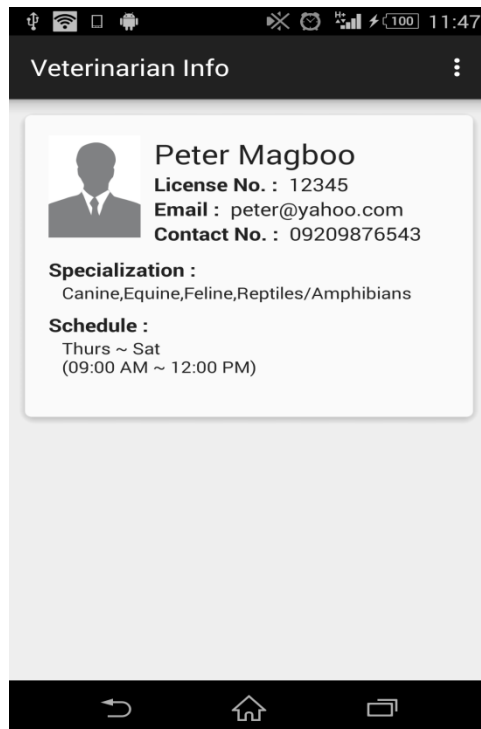


Figure 35. SpotAVet android: Veterinarian Information

VI. Discussion

SpotAVet is a Web-based system application that provides support for pet owners to get the service they need for their pets. With an extension to Android device that allows user to search veterinary clinic base on location, the system becomes more flexible. Even though the main functionality such as registration, appointment request and storage of pet medical record are only available on Web, by using the Android extension, pet owners, registered or not registered, could already have the idea on where the clinic is, as well as its schedule and its veterinarians. Furthermore, information about veterinarians schedule and specialty are also provided.

Aside from the information about the clinic and veterinarians, the system also provides the capability to store pet medical records. Consultation result, medication given, and laboratory results are examples of data the system stores as part of pet medical record. But, this information is only available to both pet owners and veterinarian for confidentiality.

Interested clinic could request for account to gain access to the system, this is done so that administrator of the system have time to investigate legitimacy of the clinic applying for system use. After the clinic account has been approved by system administrator, the clinic owner could now add veterinarian account for veterinarians working in their clinic. Veterinarians could later on create account for pet owners which would later on use the system for adding pet information and appointment request.

To ensure that all clinic data are correct, System Administrator was given the privilege to activate or deactivate a clinic account. In cases for example, if the clinic closes or stopped their operation the system admin could immediately deactivate its clinic account.

VII. Conclusion

Utilizing the features provided by SpotAVet, with support from various veterinary clinics, veterinarians and pet owners could improve the way we deal with some of the problems associated with animals and pets. Getting the help we need when we need them, and the information when we need them can be a key in solving problems associated with pet and animals on our community.

Also, SpotAVet, when enhanced, could greatly improve the way we handle pet medical record. When this records are properly handled, it could soon be reused by veterinarians and improve the way to deal with various diseases that pets or animals are suffering.

Furthermore, SpotAVet can also be used as template or reference for other similar applications. As a location based application that works both in Web and Android platform it can be considered as framework when developing similar applications. With the use of this kind of application we could revolutionize the way we search and explore things.

Finally, with the use of current technology where fast and reliable information are greatly in demand, use of mobile devices to access this information is crucial.

VIII. Recommendations

SpotAVet allows pet owner to search by clinic name or veterinarian name and email address whenever they want to set appointment. Improving the search option to more advance, like by specialization or clinic services, would greatly improve the benefit users could attain by using the system. Also, enabling the Android extension to allow setting-up appointment would be a great feature in the future.

The current scope of the system is limited to veterinary clinic that is servicing outpatient request, adding option for veterinary hospitals and inpatients services would broaden the scope. This would give users more diverse selection and information when using the system.

Having a standard or more comprehensive form of capturing pet information and record would greatly improve the information the system could provide its users. Identifying sensitive information and doing encryption would secure users information and could boost confidence in using the system.

IX. Bibliography

- [1] *The Therapeutic & Health Benefits of Pets*. (n.d.). Retrieved March 23, 2014, from Helpguide.org: A Trusted Non-Profit Resource: <http://www.helpguide.org/life/pets.htm>
- [2] *Veterinary Public Health*. (n.d.). Retrieved March 23, 2014, from WHO | World Health Organization: <http://www.who.int/zoonoses/vph/en/>
- [3] *The Importance of Pets – 6 Benefits of Owning a Pet*. (n.d.). Retrieved March 23, 2014, from Entirely For You: <http://entirely4you.com/2010/05/11/the-importance-of-pets-6-benefits-of-owning-a-pet/>
- [4] Mantovani, A. (1992). Collaborating Centre for Research and Training in Veterinary Public Health. *Revue Scientifique et Technique (International Office of Epizootics)* , 205-218.
- [5] Dr. Ronald Hines, D. (n.d.). *Diseases We Catch From Our Pets - Zoonotic Illnesses Of Dogs Cats And Other Pets*. Retrieved March 23, 2014, from <http://www.2ndchance.info/zoonoses.htm>
- [6] Smith, R. D., & Williams, M. (2000, January). Applications of informatics in veterinary medicine. *Bulletin of the Medical Library Association* , 49–51.
- [7] Alpi, K. M., & Kenan, W. J. (2009). Exploring the State of Veterinary Informatics. *Library of Veterinary Medicine* , 1-15.
- [8] Amarille, K. C., Seblós, C. R., Bustilla, N. F., Billones, J. C., & Asañero, J. P. (2010). Web-Based Integrated Information System of Phoenix Veterinary Clinic. *Advancing Information Technology Research* , 127–143.

- [9] Ducheve, Z., Cong, T. V., & Groeneveld, E. (2010). CryoWEB: Web software for the documentation of the cryo-preserved material in animal gene banks. *Bioinformation* , 219-220.
- [10] ESRI. (n.d.). *Geographic Information Systems and Pandemic Influenza Planning and Response*. Retrieved March 23, 2014, from <http://www.esri.com/library/whitepapers/pdfs/gis-and-pandemic-planning.pdf>
- [11] Martirez, M. R. (2004). "Nearest Hospital Search and Inquiry System ". University of the Philippines, Manila.
- [12] Sadie, A. A. (2010). "Telehealth's Geographic Information System". University of the Philippines, Manila.
- [13] Tsai, F. S. (2011). Web-based geographic search engine for location-aware search in Singapore. *Expert Systems with Applications* , 1011–1016.
- [14] Rinaldi, L. G. (2011). Geographical information systems as a tool in the control of heartworm infections in dogs and cats. *Veterinary Parasitology* (176), 286-290.
- [15] MacEachren, A. M., Crawford, S., Akella, M., & Lengerich, G. (2008). Design and Implementation of a Model, Web-based, GISEnabled Cancer Atlas. *The Cartographic journal* , 45 (4), 246-260.
- [16] Hochmair, H. H., & Fu, Z. (2013). Web Based Bicycle Trip Planning for Broward County, Florida. *Geographic Information Sciences Commons*.
- [17] Singhal, M., & Shukla, A. (2012). Implementation of Location based Services in Android using GPS and Web Services. *International Journal of Computer Science Issues*.

- [18] Parne, S. R. (2010). Android Application of Restaurant Finder. <https://krex.k-state.edu/dspace/bitstream/handle/2097/14061/SpoorthyReddyParne2012.pdf>
- [19] *The Android Story*. (n.d.). Retrieved April 21, 2015, from https://www.android.com/intl/en_us/history/
- [20] National Wildlife Coordinating Group. (2007, June). *Chapter 3 : Geographic Location Systems*. Retrieved March 23, 2014, from Basic Land Navigation: http://www.nwcg.gov/pms/pubs/475/PMS475_chap3.pdf
- [21] *Coordinate systems, map projections, and geographic (datum) transformations*. (n.d.). Retrieved March 23, 2014, from ArcGIS Resources: <http://resources.esri.com/help/9.3/arcgisengine/dotnet/89b720a5-7339-44b0-8b58-0f5bf2843393.htm>
- [22] *Google Maps*. (n.d.). Retrieved April 21, 2015, from <http://whatis.techtarget.com/definition/Google-Maps>
- [23] *Introducing JSON*. (n.d.). Retrieved April 21, 2015, from <http://json.org/>
- [24] *The Google Geocoding API*. (n.d.). Retrieved March 23, 2014, from Google Developers: <https://developers.google.com/maps/documentation/geocoding/>
- [25] *Latitude & Longitude Haversine Formula*. (n.d.). Retrieved March 23, 2014, from <http://www.longitudestore.com/haversine-formula.html>
- [26] *Calculate distance, bearing and more between Latitude/Longitude points*. (n.d.). Retrieved March 23, 2014, from Movable Type - Information Design & Management: <http://www.movable-type.co.uk/scripts/latlong.html>

[27] Dehling T, G. F. (2015). Exploring the Far Side of Mobile Health: Information Security and Privacy of Mobile Health Apps on iOS and Android. *JMIR mHealth and uHealth*.

Retrieved June 8, 2015, from <http://www.jmir.org/2015/5/e127/>

[28] Moin T, E. K. (2015). Women Veterans' Experience With a Web-Based Diabetes Prevention Program: A Qualitative Study to Inform Future Practice. *Journal of Medical*

Internet Research. Retrieved June 8, 2015, from <http://mhealth.jmir.org/2015/1/e8/>

X. Appendix

A. Source Code - PHP

Listing 1: controllers/main.php

```
<?php
defined('BASEPATH') OR exit('No direct
script access allowed');

class Main extends CI_Controller {

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

        $this->load->helper('url');
        $this->load->helper('form');
        $this->load-
>library('form_validation');
        $this->load-
>model('model_home');
    }

    /**
     * Index Page for this controller.
     *
     * Maps to the following URL
     *
     * http://example.com/index.php/main
     * - or -
     *
     * http://example.com/index.php/main/ind
ex
     * - or -
     * Since this controller is set as
the default controller in
     * config/routes.php, it's displayed
at http://example.com/
     *
     * So any other public methods not
prefixed with an underscore will
     * map to
     * /index.php/welcome/<method_name>
     * @see
     * http://codeigniter.com/user_guide/general/ur
ls.html
     */
    public function index()
    {
        $this->load-
>view('panels/header');
        $this->load-
>view('panels/sidebar');
        $this->load->view('home');
        $this->load-
>view('panels/footer');
    }

    public function login()
    {
        $data['title'] = 'Login';
        $data['status'] = $this-
>session->userdata('error_msg');

        if($this->session-
>userdata('isLoggedIn') == FALSE){
            $this->load-
>view('panels/header');
            $this->load-
>view('login', $data);
            $this->load-
>view('panels/footer');
            $this->session-
>set_userdata('error_msg', "");
        } else {
            $this->home();
        }
    }

    public function verify_login()
    {
        $username = $this->input-
>post('username');
        $password = $this->input-
>post('password');

        if($this->model_home-
>isUser($username, $password)){
            redirect('/main/home');
        } else {
            if($this->session-
>userdata('isLoggedIn') == FALSE){
                redirect('/main/login');
            } else {
                redirect('/main/home');
            }
        }
    }

    public function delete_session()
    {
        $sess_data = array(
            'username',
            'user_role',
            'email',
            'last_name',
            'first_name',
            'middle_name',
            'contact_no',
            'error_msg',
            'clinic_owner_id',
            'clinic_id',
            'vet_id',
            'owner_id'
        );

        $this->session-
>unset_userdata($sess_data);
    }
}
```

```

        $this->session-
>set_userdata('isLoggedIn', FALSE);
    }

    public function request_account()
    {
        $data['title'] = 'Request
Clinic Account';
        $this->load-
>view('panels/header', $data);
        $this->load-
>view('request_account', $data);
        $this->load-
>view('panels/footer');
    }

    public function submit_request()
    {
        $this->form_validation-
>set_rules('username', 'user name',
'required|callback_check_username');
        $this->form_validation-
>set_rules('password', 'password',
'required');
        $this->form_validation-
>set_rules('confirm_password', 'confirm
password', 'required|matches[password]');
        $this->form_validation-
>set_rules('email', 'e-mail address',
'required|valid_email|callback_check_email')
;
        $this->form_validation-
>set_rules('last_name', 'last name',
'required');
        $this->form_validation-
>set_rules('first_name', 'first name',
'required');
        $this->form_validation-
>set_rules('middle_name', 'middle name',
'required');
        $this->form_validation-
>set_rules('contact_no', 'contact number',
'required');
        $this->form_validation-
>set_rules('clinic_name', 'clinic name',
'required');
        $this->form_validation-
>set_rules('clinic_addr', 'clinic address',
'required');
        $this->form_validation-
>set_rules('clinic_hours', 'clinic
schedule',
'required|callback_check_schedule');
        $this->form_validation-
>set_rules('placepicker', 'clinic location',
'required');

        if ($this->form_validation-
>run() == FALSE) {
            $this-
>request_account();
        } else {

            $userRequest['username']    = $this-
>input->post('username');

            $userRequest['password']    = $this-
>input->post('password');
            $userRequest['email']
= $this->input->post('email');

            $userRequest['last_name']    = $this-
>input->post('last_name');

            $userRequest['first_name']   = $this-
>input->post('first_name');

            $userRequest['middle_name']  = $this-
>input->post('middle_name');

            $userRequest['contact_no']   = $this-
>input->post('contact_no');

            $userRequest['clinic_name']  = $this-
>input->post('clinic_name');

            $userRequest['clinic_addr']  = $this-
>input->post('clinic_addr');
            $userRequest['lat']
= $this->input->post('lat');
            $userRequest['lng']
= $this->input->post('lng');

            $userRequest['schedule']     = $this-
>input->post('clinic_hours');

            $userRequest['role_id']      = 2;
            $userRequest['status']
= 3;

            $this->model_home-
>addUser($userRequest);
            $this->model_home-
>addClinicRequest($userRequest);

            $data['title'] =
"Login";
            $data['status'] =
"Request has been submitted";
            $this->load-
>view('panels/header', $data);
            $this->load-
>view('login', $data);
            $this->load-
>view('panels/footer');
        }

        public function check_username($str)
        {
            if ($this->model_home-
>isUsernameUsed($str)) {
                $this-
>form_validation-
>set_message('check_username', $str . ' is
already used! Please choose another.');
```

```

        }
        else {
            return TRUE;
        }
    }

    public function check_schedule($str)
    {
        if ((preg_match("/[a-zA-Z]{3}\s*\s*[a-zA-Z]{3}\s*:\s*\d{1,2}:\d{1,2}\s*[ampAMP]{2}\s*~\s*\d{1,2}:\d{1,2}\s*[ampAMP]{2}\s*/", $str, $matches)
            || (preg_match("/[a-zA-Z]{3}\s*:\s*\d{1,2}:\d{1,2}\s*[ampAMP]{2}\s*~\s*\d{1,2}:\d{1,2}\s*[ampAMP]{2}\s*/", $str, $matches))) {
            return TRUE;
        } else {
            $this->form_validation->set_message('check_schedule', 'Please follow the format (WED ~ FRI :: 09:00 AM ~ 05:00 PM)');
            return FALSE;
        }
    }

    /*****
    *****
    *****
    *****
    *****
    *****
    *****
    *****
    *****
    *****
    *****
    *****
    *****
    *****
    *****/
    public function home()
    {
        if ( $this->session->userdata('isLoggedIn') == FALSE) {
            redirect('/');
        }

        $this->load->view('panels/header');
        $this->load->view('panels/sidebar');
        $this->load->view('home');
        $this->load->view('panels/footer');
    }

    public function view_profile()
    {
        if ( $this->session->userdata('isLoggedIn') == FALSE) {
            redirect('/');
        }

        $data['profile'] = $this->model_home->get_profile($_SESSION['username']);

        $this->load->view('panels/header');
        $this->load->view('panels/sidebar-profile');
        $this->load->view('content/my-profile', $data);
        $this->load->view('panels/footer');
    }
}

}

public function edit_profile()
{
    if ( $this->session->userdata('isLoggedIn') == FALSE) {
        redirect('/');
    }

    $data['profile'] = $this->model_home->get_profile($this->session->userdata('username'));
    $data['specializations'] = $this->model_home->get_specializationlist();
    $data['license_no'] = $this->model_home->get_vetlicense($this->session->userdata('username'));

    $data['vet_special'] = $this->model_home->get_specializations($this->session->userdata('vet_id'));

    $this->load->view('panels/header');
    $this->load->view('panels/sidebar');
    $this->load->view('content/edit-profile', $data);
    $this->load->view('panels/footer');
}

public function editing_profile()
{
    if ( $this->session->userdata('isLoggedIn') == FALSE) {
        redirect('/');
    }

    if ($_SESSION['user_role'] == 3) {
        $this->form_validation->set_rules('license_no', 'license number', 'required');
        $this->form_validation->set_rules('specialize[]', 'specialization', 'required');
    }

    $this->form_validation->set_rules('password', 'password', 'required');
    $this->form_validation->set_rules('confirm_password', 'confirm password', 'required|matches[password]');
    $this->form_validation->set_rules('last_name', 'last name', 'required');
    $this->form_validation->set_rules('first_name', 'first name', 'required');
    $this->form_validation->set_rules('middle_name', 'middle name', 'required');
    $this->form_validation->set_rules('contact_no', 'contact number', 'required');

    if ($this->form_validation->run() == FALSE)
    {

```

```

                $this->edit_profile();
            }
            else
            {
                $userRequest['username'] = $this->input->post('username');
                $userRequest['password'] = $this->input->post('password');
                $userRequest['last_name'] = $this->input->post('last_name');
                $userRequest['first_name'] = $this->input->post('first_name');
                $userRequest['middle_name'] = $this->input->post('middle_name');
                $userRequest['contact_no'] = $this->input->post('contact_no');

                // call db2 edit
                routine
                if
                ($SESSION['user_role'] == 3) {
                    //
                    veterinarian

                    $userRequest['license_no'] = $this->input->post('license_no');
                    $this->model_home->update_vetlicense($userRequest['username'],
                    $userRequest['license_no']);
                    $this->model_home->add_vetspecialization($this->session->userdata('vet_id'),
                    $_POST['specialize']);
                }

                $this->model_home->update_userprofile($userRequest);
                $this->view_profile();
            }
        }

        public function
        view_clinic($clinic_id)
        {
            if ( $this->session->userdata('isLoggedIn') == FALSE) {
                redirect('/');
            }

            $data['clinic'] = $this->model_home->get_clinic_info($clinic_id);

            $this->load->view('panels/header');
            $this->load->view('panels/sidebar-clinic', $data);
            $this->load->view('content/my-clinic', $data);
            $this->load->view('panels/footer');
        }

        public function logout()
        {
            $this->delete_session();
            redirect('/');
        }

        /*****
        *****
        *****
        *       Admin exclusive methods
        *****
        *****
        *****/

        public function manage_clinics()
        {
            if ( $this->session->userdata('isLoggedIn') == FALSE ||
            $_SESSION['user_role'] != 1) {
                redirect('/');
            }

            $data['clinicsList'] = $this->model_home->get_all_clinics();

            $this->load->view('panels/header');
            $this->load->view('panels/sidebar');
            $this->load->view('content/viewClinics', $data);
            $this->load->view('panels/footer');
        }

        public function modify_activation()
        {
            if ( $this->session->userdata('isLoggedIn') == FALSE ||
            $_SESSION['user_role'] != 1) {
                redirect('/');
            }

            $clinic_id =
            $_POST['clinic_id'];
            $this->model_home->update_activation($clinic_id);

            $this->manage_clinics();
        }

        public function clinic_requests()
        {
            if ( $this->session->userdata('isLoggedIn') == FALSE ||
            $_SESSION['user_role'] != 1) {
                redirect('/');
            }

            $data['clinicsList'] = $this->model_home->get_all_clinicrequests();

            $this->load->view('panels/header');
            $this->load->view('panels/sidebar');
            $this->load->view('content/viewClinicRequests', $data);
            $this->load->view('panels/footer');
        }
    }
}

```

```

        public function
approve_request($username)
    {
        if ($this->session-
>userdata('isLoggedIn') == FALSE ||
$_SESSION['user_role'] != 1) {
            redirect('/');
        }

        $this->model_home-
>approve_clinicrequest($username);
        $this->clinic_requests();
    }

    public function
delete_request($username)
    {
        if ($this->session-
>userdata('isLoggedIn') == FALSE ||
$_SESSION['user_role'] != 1) {
            redirect('/');
        }

        $this->model_home-
>delete_clinicrequest($username);
        $this->clinic_requests();
    }

    /*****
    *****
    *****
    *      Clinic Owner exclusive
    methods
    *****
    *****
    *****/

    public function
view_myclinic($clinic_owner_id)
    {
        if ( $this->session-
>userdata('isLoggedIn') == FALSE ||
$_SESSION['user_role'] != 2) {
            redirect('/');
        }

        $data['clinic'] = $this-
>model_home-
>get_myclinic_info($clinic_owner_id);

        $this->load-
>view('panels/header');
        $this->load-
>view('panels/sidebar-clinic', $data);
        $this->load-
>view('content/my-clinic', $data);
        $this->load-
>view('panels/footer');
    }

    public function edit_clinic()
    {
        if ( $this->session-
>userdata('isLoggedIn') == FALSE ||
$_SESSION['user_role'] != 2) {
            redirect('/');
        }

        $data['clinic'] = $this-
>model_home->get_clinic_info($this->session-
>userdata('clinic_owner_id'));

        $this->load-
>view('panels/header');
        $this->load-
>view('panels/sidebar');
        $this->load-
>view('content/edit-clinic', $data);
        $this->load-
>view('panels/footer');
    }

    public function editing_clinic()
    {
        if ( $this->session-
>userdata('isLoggedIn') == FALSE) {
            redirect('/');
        }

        $this->form_validation-
>set_rules('clinic_email', 'e-mail address',
'required|valid_email');
        $this->form_validation-
>set_rules('contact_no', 'contact number',
'required');
        $this->form_validation-
>set_rules('clinic_name', 'clinic name',
'required');
        $this->form_validation-
>set_rules('clinic_loc', 'clinic address',
'required');
        $this->form_validation-
>set_rules('clinic_hours', 'clinic
schedule',
'required|callback_check_schedule');

        $this->form_validation-
>set_rules('placepicker', 'clinic location',
'required');

        if ($this->form_validation-
>run() == FALSE)
        {
            $this->edit_clinic();
        }
        else
        {
            $clinic['clinic_name']
= $this->input->post('clinic_name');
            $clinic['address']
= $this->input->post('clinic_loc');
            $clinic['schedule']
= $this->input->post('clinic_hours');
            $clinic['contact_no']
= $this->input->post('contact_no');
            $clinic['email']
= $this->input->post('clinic_email');
            $clinic['lat']
= $this->input->post('lat');
            $clinic['lng']
= $this->input->post('lng');
            $this->model_home-
>update_clinicinfo($clinic, $this->session-
>userdata('clinic_owner_id'));
            $this-
>view_clinic($this->session-
>userdata('clinic_owner_id'));
        }
    }

    public function view_veterinarians()
    {

```

```

        if ( $this->session-
>userdata('isLoggedIn') == FALSE ||
$_SESSION['user_role'] != 2) {
            redirect('/');
        }

        $data['vetsList'] = $this-
>model_home-
>get_clinic_veterinarians($_SESSION['clinic_
id']);

        $this->load-
>view('panels/header');
        $this->load-
>view('panels/sidebar');
        $this->load-
>view('content/my-veterinarians', $data);
        $this->load-
>view('panels/footer');
    }

    public function view_vet($vet_id)
    {
        if ( $this->session-
>userdata('isLoggedIn') == FALSE) {
            redirect('/');
        }

        $data['vet_info'] = $this-
>model_home->get_veterinarian_info($vet_id);
        $data['profile'] = $this-
>model_home-
>get_profile($data['vet_info']['username']);

        $this->load-
>view('panels/header');
        $this->load-
>view('panels/sidebar-profile');
        $this->load-
>view('content/my-vet', $data);
        $this->load-
>view('panels/footer');
    }

    public function add_veterinarian()
    {
        if ( $this->session-
>userdata('isLoggedIn') == FALSE ||
$_SESSION['user_role'] != 2) {
            redirect('/');
        }

        $data['specializations'] =
$this->model_home->get_specializationlist();

        $this->load-
>view('panels/header');
        $this->load-
>view('panels/sidebar');
        $this->load-
>view('content/add-veterinarian', $data);
        $this->load-
>view('panels/footer');
    }

    public function search_vets()
    {
        if ( $this->session-
>userdata('isLoggedIn') == FALSE ||
$_SESSION['user_role'] != 2) {
            redirect('/');
        }

        $query = $_POST['searchvet'];
        $data['vetsResult'] = $this-
>model_home->search_vet($query);

        echo $data['vetsResult'];
    }

    public function addVet()
    {
        if ( $this->session-
>userdata('isLoggedIn') == FALSE ||
$_SESSION['user_role'] != 2) {
            redirect('/');
        }

        $username =
$_POST['hidden_username'];

        if($username == "") {
            // add new
            $this-
>form_validation->set_rules('specialize[]',
'specialization', 'required');
            $this-
>form_validation->set_rules('username',
'username',
'required|callback_check_username');
            $this-
>form_validation->set_rules('password',
'password',
'required|matches[confirm_password]');
            $this-
>form_validation-
>set_rules('confirm_password', 'confirm
password', 'required');
            $this-
>form_validation->set_rules('email', 'email
address',
'required|valid_email|callback_check_email');
            $this-
>form_validation->set_rules('last_name',
'last name', 'required');
            $this-
>form_validation->set_rules('first_name',
'first name', 'required');
            $this-
>form_validation->set_rules('middle_name',
'middle name', 'required');
            $this-
>form_validation->set_rules('license_no',
'license number', 'required');
            $this-
>form_validation->set_rules('contact_no',
'contact number', 'required');
            $this-
>form_validation->set_rules('schedule',
'schedule',
'required|callback_check_schedule');

            if ($this-
>form_validation->run() == FALSE) {
                $this-
>add_veterinarian();
            } else {
                // add to
                clinic_vet (clinic_id, vet_id, schedule)
            }
        }
    }
}

```

```

        $userRequest['username'] = $this-
>input->post('username');

        $userRequest['password'] = $this-
>input->post('password');

        $userRequest['email'] = $this-
>input->post('email');

        $userRequest['last_name'] = $this-
>input->post('last_name');

        $userRequest['first_name'] = $this-
>input->post('first_name');

        $userRequest['middle_name'] = $this-
>input->post('middle_name');

        $userRequest['contact_no'] = $this-
>input->post('contact_no');

        $userRequest['schedule'] = $this-
>input->post('schedule');

        $userRequest['role_id'] = 3;

        $userRequest['status'] = 1;

        $userRequest['license_no'] = $this-
>input->post('license_no');
        // add to user
        (username, password, role_id = 3, email,
        last_name, first_name, middle_name,
        contact_no, status = 1)
        $this-
>model_home->addUser($userRequest);

        // add to
        veterinarian (username, license_no) vet_id
        $this-
>model_home->addVetDB($userRequest);

        // add to
        clinic_vet (clinic_id, vet_id, schedule)
        $clinic_id =
        $this->session->userdata('clinic_owner_id');
        $vet_id =
        $this->model_home-
>get_vetid($userRequest['username']);
        $sched =
        $_POST['schedule'];
        $this-
>model_home->addClinicVet($clinic_id,
        $vet_id, $sched);

        $this-
>model_home->add_vetspecialization($vet_id,
        $_POST['specialize']);

        $this-
>view_veterinarians();
    }
} else {
    $this-
>form_validation-
>set_rules('hidden_username', '',
'callback_check_vetinclinic');
    $this-
>form_validation->set_rules('schedule',
'schedule',
'required|callback_check_schedule');

    $this-
>form_validation->set_rules('searchvet',
'search', 'required');

    if ($this-
>form_validation->run() == FALSE) {
        $this-
>add_veterinarian();
    }
    else
    {
        $clinic_id =
        $this->session->userdata('clinic_owner_id');
        $vet_id =
        $this->model_home->get_vetid($username);
        $sched =
        $_POST['schedule'];
        $this-
>model_home->addClinicVet($clinic_id,
        $vet_id, $sched);

        $this-
>view_veterinarians();
    }
}

    public function
    check_vetinclinic($username)
    {
        $vet_id = $this->model_home-
>get_vetid($username);
        if (array_filter($this-
>model_home-
>get_veterinarian_schedule($vet_id,
        $_SESSION['clinic_id'])) {
            $this-
>form_validation-
>set_message('check_vetinclinic', 'Vet is
        already in clinic\'s veterinarian list.');
```

```

        $this->load-
>view('panels/header');
        $this->load-
>view('panels/sidebar');
        $this->load-
>view('content/my-clinics', $data);
        $this->load-
>view('panels/footer');
    }

    public function
edit_clinic_sched($clinic_id)
    {
        if ( $this->session-
>userdata('isLoggedIn') == FALSE ||
$_SESSION['user_role'] != 3) {
            redirect('/');
        }

        $data['schedule'] = $this-
>model_home->get_schedule_str($clinic_id,
$_SESSION['vet_id'], 0);
        $data['clinic_id'] =
$clinic_id;

        $this->load-
>view('panels/header');
        $this->load-
>view('panels/sidebar');
        $this->load-
>view('content/edit-vet-sched', $data);
        $this->load-
>view('panels/footer');
    }

    public function update_vet_sched()
    {
        if ( $this->session-
>userdata('isLoggedIn') == FALSE ||
$_SESSION['user_role'] != 3) {
            redirect('/');
        }

        $this->form_validation-
>set_rules('new_sched', 'vet schedule',
'required|callback_check_schedule');

        if ($this->form_validation-
>run() == FALSE)
        {
            $this-
>edit_clinic_sched($_POST['clinic_id']);
        }
        else
        {
            $vet_sched['clinic_id'] =
$_POST['clinic_id'];
            $vet_sched['vet_id']
= $_SESSION['vet_id'];
            $vet_sched['schedule']
= $_POST['new_sched'];

            $this->model_home-
>update_vetclinicsched($vet_sched);
            $this->view_clinics();
        }
    }

    public function
view_patients($clinic_id = 0)
    {
        if ( $this->session-
>userdata('isLoggedIn') == FALSE ||
$_SESSION['user_role'] != 3) {
            redirect('/');
        }

        $data['petsList'] = $this-
>model_home->get_patients($this->session-
>userdata('vet_id'));

        $data['clinics'] = $this-
>model_home-
>get_veterinarian_schedule($this->session-
>userdata('vet_id'));

        $this->load-
>view('panels/header');
        $this->load-
>view('panels/sidebar');
        $this->load-
>view('content/my-patients', $data);
        $this->load-
>view('panels/footer');
    }

    public function add_patient()
    {
        if ( $this->session-
>userdata('isLoggedIn') == FALSE ||
$_SESSION['user_role'] != 3) {
            redirect('/');
        }

        $data['clinics'] = $this-
>model_home-
>get_veterinarian_schedule($_SESSION['vet_id
']);

        $this->load-
>view('panels/header');
        $this->load-
>view('panels/sidebar');
        $this->load-
>view('content/add-patient', $data);
        $this->load-
>view('panels/footer');
    }

    public function get_patient_list()
    {
        $clinic_id =
$_POST['clinic_id'];

        $pet_list = $this-
>model_home->get_patients($this->session-
>userdata('vet_id'), $clinic_id);

        $html_string = '<div>';
        $html_string .= '<table
class="patient" "width="700" border="0"
style="font-family: Arial, Helvetica, sans-
serif; font-size:14px;">
                <tr>
                    <th
width="189" style="border: #C0C0C0 1px
solid;"><strong>Pet Name</strong></th>
                    <th
width="189" style="border: #C0C0C0 1px
solid;"><strong>Specie</strong></th>

```



```

        <th
width="189" style="border: #C0C0C0 1px
solid;"><strong>Breed</strong></th>
        <th
width="392" style="border: #C0C0C0 1px
solid;"><strong>Owner</strong></th>
        <th
width="40" style="border: #C0C0C0 1px
solid;"><strong></strong></th>
    </tr>;
        foreach ($pet_list as $pet)
        {
            $pet_info = $this-
>model_home->get_pet_info($pet['pet_id']);
            $username = $this-
>model_home-
>get_owner_username($pet_info['owner_id']);
            $owner = $this-
>model_home->get_profile($username);
            $html_string .=
'<tr>;
                $html_string .=
'<td style="border: #C0C0C0 1px
solid;">' . $pet_info['pet_name'] . '</td>';
                $html_string .=
'<td style="border: #C0C0C0 1px
solid;">' . $pet_info['specie'] . '</td>';
                $html_string .=
'<td style="border: #C0C0C0 1px
solid;">' . $pet_info['breed'] . '</td>';
                $html_string .=
'<td style="border: #C0C0C0 1px
solid;">' . $owner['first_name'] . ' ' .
$owner['last_name'] . '</td>';
                $html_string .=
'<td style="border: #C0C0C0 1px
solid;">
                    <a href
= ' . "\"\" .
base_url("index.php/main/view_pet") . "/" .
$pet['pet_id'] . "\"\" . '>';
                    $html_string .=
'<img src="" .
base_url("resources/images/view.png") . "'
title="View Pet Info"width="20" height="20"
/></a>';
                    $html_string .=
'</td>';
                $html_string .=
'</tr>';
            }
            $html_string .=
'</table></div>';
            echo $html_string;
        }
        public function search_patients()
        {
            if ( $this->session-
>userdata('isLoggedIn') == FALSE ||
$_SESSION['user_role'] != 3) {
                redirect('/');
            }
            $query =
$_POST['searchpatient'];
            $data['patientsResult'] =
$this->model_home->search_patient($query);
            echo $data['patientsResult'];

```

```

        }
        public function add_owner()
        {
            if ( $this->session-
>userdata('isLoggedIn') == FALSE ||
$_SESSION['user_role'] != 3) {
                redirect('/');
            }
            $this->form_validation-
>set_rules('consultation_date',
'consultation date', 'required');
            $this->form_validation-
>set_rules('complaints', 'chief complaints',
'required');
            $this->form_validation-
>set_rules('diagnosis', 'diagnosis /
findings', 'required');
            $this->form_validation-
>set_rules('clinic_option', 'clinic',
'required');
            if
($ _POST['hidden_medication']) {
                $this-
>form_validation-
>set_rules('hidden_medication', '',
'required');
                $this-
>form_validation->set_rules('medicine',
'medicine', 'required');
                $this-
>form_validation->set_rules('dosage',
'dosage', 'required');
                $this-
>form_validation-
>set_rules('medication_date',
'medication_date', 'required');
            }
            if
($ _POST['hidden_vaccination']) {
                $this-
>form_validation-
>set_rules('hidden_vaccination', '',
'required');
                $this-
>form_validation->set_rules('vaccine_type',
'vaccine type', 'required');
                $this-
>form_validation->set_rules('vaccine_name',
'vaccine name', 'required');
                $this-
>form_validation-
>set_rules('expiration_date',
'expiration_date', 'required');
                $this-
>form_validation-
>set_rules('next_vaccine_date',
'next_vaccine_date', 'required');
            }
            if
($ _POST['hidden_laboratory']) {
                $this-
>form_validation-
>set_rules('hidden_laboratory', '',
'required');
                $this-
>form_validation-

```

```

>set_rules('laboratory_type', 'laboratory
type', 'required');
    $this-
>form_validation->set_rules('lab_findings',
'laboratory findings', 'required');
    $this-
>form_validation->set_rules('lab_date',
'laboratory date', 'required');
    }

    $pet_id =
$_POST['hidden_info'];
    if ($pet_id == 0) {
        // pet owner fields
        $this-
>form_validation->set_rules('username',
'username',
'required|callback_check_username');
        $this-
>form_validation->set_rules('password',
'password',
'required|matches[confirm_password]');
        $this-
>form_validation-
>set_rules('confirm_password', 'confirm
password', 'required');
        $this-
>form_validation->set_rules('email',
'email',
'required|valid_email|callback_check_email')
;
        $this-
>form_validation->set_rules('last_name',
'last name', 'required');
        $this-
>form_validation->set_rules('first_name',
'first name name', 'required');
        $this-
>form_validation->set_rules('middle_name',
'middle name', 'required');
        $this-
>form_validation->set_rules('contact_no',
'contact number', 'required');

        // pet info fields
        $this-
>form_validation->set_rules('pet_name', 'pet
name', 'required');
        $this-
>form_validation->set_rules('specie', 'pet
specie', 'required');
        $this-
>form_validation->set_rules('breed', 'pet
breed', 'required');
        $this-
>form_validation->set_rules('birthday', 'pet
birthday', 'required');
        $this-
>form_validation->set_rules('gender', 'pet
gender', 'required');
        //registration no. and
owner_id
        if ($this-
>form_validation->run() == FALSE) {
            $this-
>add_patient();
        } else {
            $userRequest['username'] =
$_POST['username'];

            $userRequest['password'] =
$_POST['password'];

            $userRequest['email'] =
$_POST['email'];

            $userRequest['last_name'] =
$_POST['last_name'];

            $userRequest['first_name'] =
$_POST['first_name'];

            $userRequest['middle_name'] =
$_POST['middle_name'];

            $userRequest['contact_no'] =
$_POST['contact_no'];

            $userRequest['role_id'] = 4;

            $userRequest['status'] = 1;
            // add to user
(username, password, role_id = 4, email,
last_name, first_name, middle_name,
contact_no, status = 1)
            $this-
>model_home->addUser($userRequest);
            $owner_id =
$this->model_home-
>addOwner($userRequest['username']);

            if ($owner_id)
            {
                $petRequest['owner_id'] =
$owner_id;

                $petRequest['pet_name'] =
$_POST['pet_name'];

                $petRequest['specie'] =
$_POST['specie'];

                $petRequest['breed'] =
$_POST['breed'];

                $petRequest['birthdate'] =
$_POST['birthday'];

                $petRequest['gender'] =
$_POST['gender'];

                $petRequest['registry'] =
$_POST['registry'];

                $pet_id
= $this->model_home->addPet($petRequest);
                if
($pet_id) {
                    $consultation['pet_id'] =
$pet_id;

                    $consultation['vet_id'] =
$_SESSION['vet_id'];

                    $consultation['clinic_id']
= $_POST['clinic_option'];

```

```

        $consultation['consultation_date'] =
$_POST['consultation_date'];
        $consultation['complaints'] =
$_POST['complaints'];
        $consultation['diagnosis'] =
$_POST['diagnosis'];
        $consultation['remarks'] =
$_POST['remarks'];
        $consultation['next_consult_date'] =
$_POST['next_consultation'];
        $consultation['schedule_appt_id'] =
'';

        $consultation_id = $this->model_home-
>addConsultation($consultation);

        if ($consultation_id) {
            if ($_POST['hidden_medication']) {
                $medication['pet_id']
= $pet_id;

                $medication['consultation_id'] =
$consultation_id;

                $medication['medicine']
= $_POST['medicine'];

                $medication['dosage']
= $_POST['dosage'];

                $medication['med_date']
= $_POST['medication_date'];

                $this->model_home-
>addMedication($medication);
            }

            if ($_POST['hidden_vaccination']) {
                $vaccination['pet_id']
= $pet_id;

                $vaccination['consultation_id'] =
$consultation_id;

                $vaccination['vaccine_type']
= $_POST['vaccine_type'];

                $vaccination['vaccine_name']
= $_POST['vaccine_name'];

                $vaccination['expiration']
= $_POST['expiration_date'];

                $vaccination['next_due']
= $_POST['next_vaccine_date'];

                $this->model_home-
>addVaccination($vaccination);
            }

            if ($_POST['hidden_laboratory']) {
                $laboratory['pet_id']
= $pet_id;

                $laboratory['consultation_id'] =
$consultation_id;

                $laboratory['lab_type'] =
$_POST['laboratory_type'];

                $laboratory['lab_findings']
= $_POST['lab_findings'];

                $laboratory['lab_date']
= $_POST['lab_date'];

                $this->model_home-
>addLaboratory($laboratory);
            }

            $this->view_patients();
        } else {
            $this-
>form_validation->set_rules('searchpatient',
'search', 'required');

            if ($this-
>form_validation->run() == FALSE)
            {
                $this-
>add_patient();
            } else {
                $consultation['pet_id'] =
$pet_id;

                $consultation['vet_id'] =
$_SESSION['vet_id'];

                $consultation['clinic_id'] =
$_POST['clinic_option'];

                $consultation['consultation_date'] =
$_POST['consultation_date'];

                $consultation['complaints'] =
$_POST['complaints'];
            }
        }
    }
}

```

```

        $consultation['diagnosis'] = $_POST['diagnosis'];
        $consultation['remarks'] = $_POST['remarks'];
        $consultation['next_consult_date'] = $_POST['next_consultation'];
        $consultation['schedule_appt_id'] = '';

        $consultation_id = $this->model_home->addConsultation($consultation);

        if ($consultation_id) {
            if ($POST['hidden_medication']) {
                $medication['pet_id'] = $pet_id;
                $medication['consultation_id'] = $consultation_id;
                $medication['medicine'] = $_POST['medicine'];
                $medication['dosage'] = $_POST['dosage'];
                $medication['med_date'] = $_POST['medication_date'];

                $this->model_home->addMedication($medication);
            }
            if ($POST['hidden_vaccination']) {
                $vaccination['pet_id'] = $pet_id;
                $vaccination['consultation_id'] = $consultation_id;
                $vaccination['vaccine_type'] = $_POST['vaccine_type'];
                $vaccination['vaccine_name'] = $_POST['vaccine_name'];
                $vaccination['expiration'] = $_POST['expiration_date'];
                $vaccination['next_due'] = $_POST['next_vaccine_date'];

                $this->model_home->addVaccination($vaccination);
            }
            if ($POST['hidden_laboratory']) {
                $laboratory['pet_id'] = $pet_id;
                $laboratory['consultation_id'] = $consultation_id;
                $laboratory['lab_type'] = $_POST['laboratory_type'];
                $laboratory['lab_findings'] = $_POST['lab_findings'];
                $laboratory['lab_date'] = $_POST['lab_date'];

                $this->model_home->addLaboratory($laboratory);
            }
        }
        $this->view_patients();
    }

    public function appointments()
    {
        if ($this->session->userdata('isLoggedIn') == FALSE) {
            redirect('/');
        }
        if ($SESSION['user_role'] == 4) {
            $data['history'] = 2;
            $data['appointments'] = $this->model_home->getAppointments($SESSION['owner_id'], 2);
        } else {
            $data['appointments'] = $this->model_home->getAppointments($SESSION['vet_id']);
            $data['history'] = 0;
        }
    }

    $this->load->view('panels/header');
    $this->load->view('panels/sidebar');
    $this->load->view('content/view-appointments', $data);
    $this->load->view('panels/footer');
}

/*****
*****
*****
***** Pet owner exclusive methods
*****
*****/

public function view_pets()
{

```

```

        if ( $this->session-
>userdata('isLoggedIn') == FALSE ||
$_SESSION['user_role'] != 4) {
            redirect('/');
        }

        $data['petsList'] = $this-
>model_home-
>get_mypets($_SESSION['owner_id']);

        $this->load-
>view('panels/header');
        $this->load-
>view('panels/sidebar');
        $this->load-
>view('content/my-pets', $data);
        $this->load-
>view('panels/footer');
    }

    public function add_pet()
    {
        if ( $this->session-
>userdata('isLoggedIn') == FALSE ||
$_SESSION['user_role'] != 4) {
            redirect('/');
        }

        $this->load-
>view('panels/header');
        $this->load-
>view('panels/sidebar');
        $this->load-
>view('content/add-pet');
        $this->load-
>view('panels/footer');
    }

    public function add_newpet()
    {
        if ( $this->session-
>userdata('isLoggedIn') == FALSE ||
$_SESSION['user_role'] != 4) {
            redirect('/');
        }

        $this->form_validation-
>set_rules('pet_name', 'pet name',
'required');
        $this->form_validation-
>set_rules('specie', 'specie', 'required');
        $this->form_validation-
>set_rules('breed', 'breed', 'required');
        $this->form_validation-
>set_rules('birthday', 'birthday',
'required');
        $this->form_validation-
>set_rules('gender', 'gender', 'required');

        if ($this->form_validation-
>run() == FALSE) {
            $this->add_pet();
        } else {
            $petRequest['owner_id'] =
$_SESSION['owner_id'];

            $petRequest['pet_name'] =
$_POST['pet_name'];
            $petRequest['specie']
= $_POST['specie'];

            $petRequest['breed']
= $_POST['breed'];
            $petRequest['birthdate'] =
$_POST['birthday'];
            $petRequest['gender']
= $_POST['gender'];
            $petRequest['registry'] =
$_POST['registry'];

            $this->model_home-
>addPet($petRequest);
            $this->view_pets();
        }
    }

    public function edit_pet($pet_id)
    {
        if ( $this->session-
>userdata('isLoggedIn') == FALSE ||
$_SESSION['user_role'] != 4) {
            redirect('/');
        }

        $data['pet'] = $this-
>model_home->get_pet_info($pet_id);
        $this->load-
>view('panels/header');
        $this->load-
>view('panels/sidebar');
        $this->load-
>view('content/edit-pet', $data);
        $this->load-
>view('panels/footer');
    }

    public function updatepet()
    {
        if ( $this->session-
>userdata('isLoggedIn') == FALSE ||
$_SESSION['user_role'] != 4) {
            redirect('/');
        }

        $this->form_validation-
>set_rules('pet_name', 'pet name',
'required');
        $this->form_validation-
>set_rules('specie', 'specie', 'required');
        $this->form_validation-
>set_rules('breed', 'breed', 'required');
        $this->form_validation-
>set_rules('birthday', 'birthday',
'required');
        $this->form_validation-
>set_rules('gender', 'gender', 'required');
        $pet_id = $_POST['pet_id'];

        if ($this->form_validation-
>run() == FALSE) {
            $this-
>edit_pet($pet_id);
        } else {
            $petRequest['pet_id']
= $pet_id;

            $petRequest['owner_id'] =
$_SESSION['owner_id'];

```

```

        $petRequest['pet_name'] =
$_POST['pet_name'];
        $petRequest['specie']
= $_POST['specie'];
        $petRequest['breed']
= $_POST['breed'];
        $petRequest['birthdate'] =
$_POST['birthday'];
        $petRequest['gender']
= $_POST['gender'];
        $petRequest['registry'] =
$_POST['registry'];
        $this->model_home-
>update_pet($petRequest);
        $this->view_pets();
    }

    public function view_pet($pet_id)
    {
        if ( $this->session-
>userdata('isLoggedIn') == FALSE) {
            redirect('/');
        }

        $data['pet'] = $this-
>model_home->get_pet_info($pet_id);
        if ($_SESSION['user_role'] ==
4)

            $data['consultationList'] = $this-
>model_home-
>get_consultation_history($pet_id);
            else

            $data['consultationList'] = $this-
>model_home-
>get_consultation_history($pet_id,
$_SESSION['vet_id']);

            $data['pet_id'] = $pet_id;

            $this->load-
>view('panels/header');
            $this->load-
>view('panels/sidebar-pet');
            $this->load-
>view('content/my-pet', $data);
            $this->load-
>view('panels/footer');
        }

        public function
view_consultation($consultation_id)
        {
            if ( $this->session-
>userdata('isLoggedIn') == FALSE) {
                redirect('/');
            }

            $data['consultation'] =
$this->model_home-
>get_consultation_info($consultation_id);
            $data['medicationList'] =
$this->model_home-
>get_consultation_medications($consultation_
id);

            $data['laboratoryList'] =
$this->model_home-
>get_consultation_laboratory($consultation_i
d);
            $data['vaccinationList'] =
$this->model_home-
>get_consultation_vaccinations($consultation
_id);

            $this->load-
>view('panels/header');
            $this->load-
>view('panels/sidebar-pet');
            $this->load-
>view('content/view-consultation', $data);
            $this->load-
>view('panels/footer');
        }

        public function appointment_request()
        {
            if ( $this->session-
>userdata('isLoggedIn') == FALSE ||
$_SESSION['user_role'] != 4) {
                redirect('/');
            }

            $data['pets'] = $this-
>model_home-
>get_mypets($_SESSION['owner_id']);

            $this->load-
>view('panels/header');
            $this->load-
>view('panels/sidebar');
            $this->load-
>view('content/add-appointment', $data);
            $this->load-
>view('panels/footer');
        }

        public function search()
        {
            if ( $this->session-
>userdata('isLoggedIn') == FALSE ||
$_SESSION['user_role'] != 4) {
                redirect('/');
            }

            $input = explode(':',
$_POST['search']);

            $query = $input[0];
            $option = $input[1];

            if ($option == "Clinic") {
                $data['result'] =
$this->model_home->search_clinic($query);
            } else {
                $data['result'] =
$this->model_home->search_vet($query);
            }

            echo $data['result'];
        }

        public function getVetList()
        {
            if ( $this->session-
>userdata('isLoggedIn') == FALSE ||
$_SESSION['user_role'] != 4) {

```

```

        redirect('/');
    }

    $clinic_id =
$_POST['clinic_id'];
    $vet_ids = $this-
>model_home-
>get_clinic_veterinarians($clinic_id);

    $result = "";
    foreach($vet_ids as $vet_id)
    {
        $vet_username = $this-
>model_home-
>get_veterinarian_info($vet_id['vet_id'])['u
sername'];
        $vet_profile = $this-
>model_home->get_profile($vet_username);
        $result .= '<option
value="" . $vet_username . "'>' .
$vet_profile['last_name'] . ', ' .
$vet_profile['first_name'] . ' ' .
$vet_profile['middle_name'] . '</option>';
    }

    $data['result'] = $result;
    echo $data['result'];
}

public function getClinicList()
{
    if ( $this->session-
>userdata('isLoggedIn') == FALSE ||
$_SESSION['user_role'] != 4) {
        redirect('/');
    }

    $vet_username =
$_POST['vet_username'];
    $vet_id = $this-
>model_home->get_vetid($vet_username);
    $clinic_ids = $this-
>model_home->get_vet_clinics($vet_id);

    $result = "";
    foreach($clinic_ids as
$clinic_id) {
        $clinic_info = $this-
>model_home-
>get_clinic_info($clinic_id['clinic_id']);
        $result .= '<option
value="" . $clinic_info['clinic_id'] . "'>'
. $clinic_info['clinic_name'] . '</option>';
    }

    $data['result'] = $result;
    echo $data['result'];
}

public function getVetDetails() {

    $vet_username =
$_POST['vet_username'];
    $clinic_id =
$_POST['clinic_id'];

    $vet_id = $this-
>model_home->get_vetid($vet_username);

    $result = "<div>";

```

```

        $result .= '<table><tr><td
style="text-align:right"><font
size="2"><strong>License No:
</strong></font></td><td><font size="2">' .
$this->model_home-
>get_vetlicense($vet_username) .
'</font></td></tr><tr><td style="text-
align:right"><font
size="2"><strong>Specialization:
</strong></font></td><td><font size="2">' .
$this->model_home-
>get_specializations_str($vet_id) .
'</font></td></tr><tr><td style="text-
align:right"><font
size="2"><strong>Schedule:
</strong></font></td><td><font size="2">' .
$this->model_home-
>get_schedule_str($clinic_id, $vet_id, 0);
        $result .=
"</td></tr></table></div>";

        $data['result'] = $result;
        echo $data['result'];
    }

    public function getClinicDetails() {

        $vet_username =
$_POST['vet_username'];
        $clinic_id =
$_POST['clinic_id'];

        $clinic_info = $this-
>model_home->get_clinic_info($clinic_id);

        $result = "<div>";
        $result .= '<table><tr><td
style="width:70px; text-align:right"><font
size="2"><strong>Address:
</strong></font></td><td><font size="2">' .
$clinic_info['address'] .
'</font></td></tr><tr><td style="text-
align:right"><font size="2"><strong>Contact
No: </strong></font></td><td><font
size="2">' . $clinic_info['contact_no'];
        $result .=
"</font></td></tr></table></div>";

        $data['result'] = $result;
        echo $data['result'];
    }

    public function add_appointment()
    {
        if ( $this->session-
>userdata('isLoggedIn') == FALSE) {
            redirect('/');
        }

        $this->form_validation-
>set_rules('search', 'search', 'required');
        $this->form_validation-
>set_rules('reason', 'reason', 'required');

        $this->form_validation-
>set_rules('pet_option', 'pet', 'required');
        $this->form_validation-
>set_rules('sched_date', 'date',
'required');
    }

```

```

        $this->form_validation-
>set_rules('sched_time', 'time',
'required');

        if ($_POST['search_option']
== "Clinic") {
            $this-
>form_validation->set_rules('vet_option',
'veterinarian', 'required');
        } else {
            $this-
>form_validation->set_rules('clinic_option',
'clinic', 'required');
        }

        if ($this->form_validation-
>run() == FALSE) {
            $this-
>appointment_request();
        } else {
            if
($ _POST['search_option'] == "Clinic") {
                $clinic_id =
$_POST['details'];
                $vet_username
= $_POST['vet_option'];
            } else {
                $vet_username
= $_POST['details'];
                $clinic_id =
$_POST['clinic_option'];
            }

            $pet_id =
$_POST['pet_option'];
            $vet_id = $this-
>model_home->get_vetid($vet_username);

            $appointment['clinic_id'] =
$clinic_id;
            $appointment['vet_id']
= $vet_id;

            $appointment['owner_id'] =
$_SESSION['owner_id'];
            $appointment['pet_id']
= $pet_id;
            $appointment['date']
= $_POST['sched_date'];
            $appointment['time']
= $_POST['sched_time'];

            $appointment['remarks'] =
$_POST['reason'];

            $appointment['confirmed'] = 0;

            $this->model_home-
>add_appointment($appointment);
            $this->home();
        }

        public function
approve_appointment($appointment_id)
        {
            if ($this->session-
>userdata('isLoggedIn') == FALSE) {
                redirect('/');
            }
        }
    }
}

```

```

        $this->model_home-
>confirm_appointment($appointment_id, 1);
        $this->appointments();
    }

    public function appointment_history()
    {
        if ($this->session-
>userdata('isLoggedIn') == FALSE) {
            redirect('/');
        }

        $data['history'] = 1;
        $data['appointments'] =
$this->model_home-
>getAppointments($_SESSION['vet_id'],
$data['history']);

        $this->load-
>view('panels/header');
        $this->load-
>view('panels/sidebar');
        $this->load-
>view('content/view-appointments', $data);
        $this->load-
>view('panels/footer');
    }

    public function
disapprove_appointment()
    {
        if ($this->session-
>userdata('isLoggedIn') == FALSE) {
            redirect('/');
        }

        $remarks =
$_POST['hidden_remarks'];
        $appointment_id =
$_POST['schedule_appt_id'];

        $this->model_home-
>confirm_appointment($appointment_id, 2,
$remarks);
        $this->appointments();
    }
}

```

Listing 2: models/model_home.php

```

<?php
class Model_home extends CI_Model {

    public function __construct()
    {
        $this->load->database();
    }

    /*****
    *****/
    * Unregistered Users
}

```



```

*****
*****
*****/

//for login
public function isUser($username,
$password) {
    //`password`=".md5($password).""
    $query = "SELECT * FROM user
WHERE username = :username";
    $query_params =
array(':username' => $username);

    $stmt = $this->db->conn_id-
>prepare($query);
    $result = $stmt-
>execute($query_params);

    //fetching all the rows from
the query
    $row = $stmt->fetch();

    if ($row) {
        //if we encrypted the
password, we would unencrypt it here, but in
our case we just
        //compare the two
passwords
        if (($password ===
$row['password']) && ($row['status'] == 1))
        {
            $newdata =
array(
                => $row['username'],
                'username'
                => $row['role_id'],
                'user_role'
                => $row['email'],
                'email'
                => $row['last_name'],
                'last_name'
                'first_name' => $row['first_name'],
                'middle_name' => $row['middle_name'],
                'contact_no' => $row['contact_no'],
                'isLoggedIn' => TRUE,
                'error_msg'
=> ""
            );
            if
($row['role_id'] == 2) {
                $clinic_owner_id = $this-
>get_clinicownerid($row['username']);
                $newdata['clinic_owner_id'] =
$clinic_owner_id;
                $clinic_id = $this-
>get_clinicid($clinic_owner_id);
                $newdata['clinic_id'] = $clinic_id;
            }

```

```

        else if
($row['role_id'] == 3) {
            $vet_id
= $this->get_vetid($row['username']);
            $newdata['vet_id'] = $vet_id;
        }
        else if
($row['role_id'] == 4) {
            $owner_id = $this-
>get_ownerid($row['username']);
            $newdata['owner_id'] = $owner_id;
        }
        $this-
>session->set_userdata($newdata);
        return true;
    } else {
        $newdata =
array();
        if
($row['status'] != 1) {
            $newdata['error_msg'] = "Clinic
account not yet approved or has been
suspended.";
        } else {
            $newdata['error_msg'] = "Username and
password mismatch. Please check username and
password used.";
        }
        $newdata['isLoggedIn'] = FALSE;
        $this-
>session->set_userdata($newdata);
        return false;
    }
    $this->session-
>set_userdata('error_msg', "Username does
not exist");
    return false;
}

public function
get_clinicownerid($username) {
    $query = "SELECT
clinic_owner.clinic_owner_id FROM
clinic_owner WHERE username = :username";
    $query_params =
array(':username' => $username);
    $stmt = $this->db->conn_id-
>prepare($query);
    $result = $stmt-
>execute($query_params);
    $row = $stmt->fetch();
    return
$row['clinic_owner_id'];
}

public function
get_clinicid($clinic_owner_id) {

```

```

        $query = "SELECT
clinic.clinic_id FROM clinic WHERE
clinic_owner_id = :clinic_owner_id";
        $query_params =
array(':clinic_owner_id' =>
$clinic_owner_id);

        $stmt = $this->db->conn_id-
>prepare($query);
        $result = $stmt-
>execute($query_params);
        $row = $stmt->fetch();

        return $row['clinic_id'];
    }

    public function get_vetid($username)
    {
        $query = "SELECT
veterinarian.vet_id FROM veterinarian WHERE
username = :username";
        $query_params =
array(':username' => $username);

        $stmt = $this->db->conn_id-
>prepare($query);
        $result = $stmt-
>execute($query_params);
        $row = $stmt->fetch();

        return $row['vet_id'];
    }

    public function
get_ownerid($username) {
        $query = "SELECT
pet_owner.owner_id FROM pet_owner WHERE
username = :username";
        $query_params =
array(':username' => $username);

        $stmt = $this->db->conn_id-
>prepare($query);
        $result = $stmt-
>execute($query_params);
        $row = $stmt->fetch();

        return $row['owner_id'];
    }

    public function
isUsernameUsed($userName) {
        $query = "SELECT * FROM user
WHERE username = :username";
        $query_params =
array(':username' => $userName);

        $stmt = $this->db->conn_id-
>prepare($query);
        $result = $stmt-
>execute($query_params);

        //fetching all the rows from
the query
        $row = $stmt->fetch();
        if ($row) {
            return true;
        }
        return false;
    }
}

```

```

    public function isEmailUsed($email) {
        $query = "SELECT * FROM user
WHERE email = :email";
        $query_params =
array(':email' => $email);

        $stmt = $this->db->conn_id-
>prepare($query);
        $result = $stmt-
>execute($query_params);

        //fetching all the rows from
the query
        $row = $stmt->fetch();
        if ($row) {
            return true;
        }
        return false;
    }

    public function addUser($userRequest)
    {
        $query = "INSERT INTO user(
        username,
        password,
        role_id,
        email,
        last_name,
        first_name,
        middle_name,
        contact_no,
        status
        )
VALUES(
        :username,
        :password,
        :role_id,
        :email,
        :last_name,
        :first_name,
        :middle_name,
        :contact_no,
        :status
        )";

        $query_params = array(
            ':username' =>
$userRequest['username'],
            ':password' =>
$userRequest['password'],
            ':role_id' =>
$userRequest['role_id'],
            ':email' =>
$userRequest['email'],
            ':last_name' =>
$userRequest['last_name'],
            ':first_name' =>
$userRequest['first_name'],
            ':middle_name' =>
$userRequest['middle_name'],
            ':contact_no' =>
$userRequest['contact_no'],
            ':status' =>
$userRequest['status']
        );

        $stmt = $this->db->conn_id-
>prepare($query);
        $result = $stmt-
>execute($query_params);
    }
}

```

```

    }
    public function
addClinicRequest($userRequest) {
    $query = "INSERT INTO
clinic_request(
    username,
    email,
    clinic_name,
    address,
    contact_no,
    schedule,
    x,
    y
    )
VALUES (
    :username,
    :email,
    :clinic_name,
    :address,
    :contact_no,
    :schedule,
    :x,
    :y
    )";
    $query_params = array(
    'username' =>
$userRequest['username'],
    'email' =>
$userRequest['email'],
    'clinic_name' =>
$userRequest['clinic_name'],
    'address' =>
$userRequest['clinic_addr'],
    'contact_no' =>
$userRequest['contact_no'],
    'schedule' =>
$userRequest['schedule'],
    ':x' =>
$userRequest['lat'],
    ':y' =>
$userRequest['lng']
    );
    $stmt = $this->db->conn_id-
>prepare($query);
    $result = $stmt-
>execute($query_params);
}

/*****
*****
*****
*
Registered Users
*****
*****
*****/

public function get_usertype() {
    $query = "SELECT roles.role
FROM roles WHERE role_id=:role_id";
    $query_params =
array(':role_id' => $this->session-
>userdata('user_role'));
    $stmt = $this->db->conn_id-
>prepare($query);
    $result = $stmt-
>execute($query_params);

```

```

    $usertype = $stmt->fetch();
return $usertype['role'];
}

public function
get_profile($username) {
    $query = "SELECT * FROM user
WHERE username=:username";
    $query_params =
array(':username' => $username);
    $stmt = $this->db->conn_id-
>prepare($query);
    $result = $stmt-
>execute($query_params);
return $stmt->fetch();
}

public function
get_vetlicense($username) {
    $query = "SELECT * FROM
veterinarian WHERE username=:username";
    $query_params =
array(':username' => $username);
    $stmt = $this->db->conn_id-
>prepare($query);
    $result = $stmt-
>execute($query_params);
    $row = $stmt->fetch();
return $row['license_no'];
}

public function
get_specializations_str($vet_id) {
    $specialization = "";
    $arr_spec = array();
    $count = 0;
    foreach ($this-
>get_specializations($vet_id) as $row) {
        $arr_spec [$count] =
$this-
>get_specialization($row['spec_id'])['specia-
lization'];
        $count++;
    }
    $specialization = implode(
",", $arr_spec);
return $specialization;
}

public function
get_specializations($vet_id) {
    $query = "SELECT * FROM
vet_specialization WHERE vet_id=:vet_id";
    $query_params =
array(':vet_id' => $vet_id);
    $stmt = $this->db->conn_id-
>prepare($query);
    $result = $stmt-
>execute($query_params);
return $stmt->fetchAll();
}

public function
get_specialization($spec_id) {

```

```

        $query = "SELECT * FROM
specialization WHERE spec_id=:spec_id";
        $query_params =
array(':spec_id' => $spec_id);

        $stmt = $this->db->conn_id-
>prepare($query);
        $result = $stmt-
>execute($query_params);

        return $stmt->fetch();
    }

    public function
get_specializationlist() {
        $query = "SELECT * FROM
specialization";

        $stmt = $this->db->conn_id-
>prepare($query);
        $result = $stmt->execute();

        return $stmt->fetchAll();
    }

    public function
update_userprofile($userRequest)
    {
        $query = "UPDATE user SET
password =
:password,
last_name =
:last_name,
first_name =
:first_name,
middle_name =
:middle_name,
contact_no =
:contact_no
WHERE
username =
:username";

        $query_params = array(
':password' =>
$userRequest['password'],
':last_name' =>
$userRequest['last_name'],
':first_name' =>
$userRequest['first_name'],
':middle_name' =>
$userRequest['middle_name'],
':contact_no' =>
$userRequest['contact_no'],
':username' =>
$userRequest['username']
);

        $stmt = $this->db->conn_id-
>prepare($query);
        $result = $stmt-
>execute($query_params);
    }

    public function
update_vetlicense($username, $license_no)
    {
        $query = "UPDATE veterinarian
SET license_no = :license_no WHERE username
= :username";

```

```

        $query_params = array(
':license_no' =>
$license_no,
':username' =>
$username
);

        $stmt = $this->db->conn_id-
>prepare($query);
        $result = $stmt-
>execute($query_params);
    }

    public function
add_vetspecialization($vet_id, $spec_list) {
        $this-
>delete_vetspecialization($vet_id);

        foreach($spec_list as
$spec_id) {
            $query = "INSERT INTO
vet_specialization(
                vet_id,
                spec_id
            )
VALUES(
                :vet_id,
                :spec_id
            )";

            $query_params = array(
':vet_id' =>
$vet_id,
':spec_id' =>
$spec_id
);

            $stmt = $this->db-
>conn_id->prepare($query);
            $result = $stmt-
>execute($query_params);
        }

        public function
delete_vetspecialization($vet_id)
        {
            $query = "DELETE FROM
vet_specialization WHERE vet_id = :vet_id";
            $query_params =
array(':vet_id' =>
$vet_id);

            $stmt = $this->db->conn_id-
>prepare($query);
            $result = $stmt-
>execute($query_params);
        }

        public function
get_clinic_info($clinic_id) {
            $query = "SELECT * FROM
clinic WHERE clinic_id=:clinic_id";
            $query_params =
array(':clinic_id' =>
$clinic_id);

            $stmt = $this->db->conn_id-
>prepare($query);
            $result = $stmt-
>execute($query_params);

            return $stmt->fetch();
        }

```

```

    }

    public function
    get_schedule_str($clinic_id, $vet_id, $br)
    {
        $schedule = "";
        foreach ($this->get_veterinarian_schedule($vet_id,
        $clinic_id) as $row) {
            $details =
            explode(":", $row['schedule']);
            if ($br)
                $schedule .=
                trim ($details[0]) . "<br /> (" . trim
                ($details[1]) . ")<br />";
            else
                $schedule .=
                trim ($details[0]) . " (" . trim
                ($details[1]) . ")";
        }
        return $schedule;
    }

    public function
    get_veterinarian_schedule($vet_id,
    $clinic_id = 0) {
        if ($clinic_id != 0)
        {
            $query = "SELECT *
            FROM clinic_vet WHERE clinic_id=:clinic_id
            AND vet_id=:vet_id";
            $query_params =
            array(':clinic_id' => $clinic_id, ':vet_id'
            => $vet_id);
        } else {
            $query = "SELECT *
            FROM clinic_vet WHERE vet_id=:vet_id";
            $query_params =
            array(':vet_id' => $vet_id);
        }

        $stmt = $this->db->conn_id->prepare($query);
        $result = $stmt->execute($query_params);

        $rows = $stmt->fetchAll();

        return $rows;
    }

    /*****
    *****
    *****
    *           Admin exclusive methods
    *****
    *****/

    public function get_all_clinics() {
        $query = "SELECT * FROM
        clinic";

        $stmt = $this->db->conn_id->prepare($query);
        $result = $stmt->execute();

        return $stmt->fetchAll();
    }

```

```

    public function
    update_activation($clinic_id) {
        $query = "SELECT
        clinic.status FROM clinic where clinic_id =
        :clinic_id";
        $query_params =
        array(':clinic_id' => $clinic_id);

        $stmt = $this->db->conn_id->prepare($query);
        $result = $stmt->execute($query_params);

        $row = $stmt->fetch();

        $query = "UPDATE clinic SET
        status = ". ($row['status'] == 1 ? 2 : 1) . "
        WHERE clinic.clinic_id = :clinic_id";
        $query_params =
        array(':clinic_id' => $clinic_id);

        $stmt = $this->db->conn_id->prepare($query);
        $result = $stmt->execute($query_params);

        $clinic_info = $this->get_clinic_info($clinic_id);
        $clinic_owner_username =
        $this->get_clinicowner_username($clinic_info['clin
        ic_owner_id']);

        $query = "UPDATE user SET
        status = ". ($row['status'] == 1 ? 2 : 1) . "
        WHERE username = :username";
        $query_params =
        array(':username' =>
        $clinic_owner_username);

        $stmt = $this->db->conn_id->prepare($query);
        $result = $stmt->execute($query_params);
    }

    public function
    get_clinicowner_username($clinic_owner_id) {
        $query = "SELECT * FROM
        clinic_owner WHERE
        clinic_owner_id=:clinic_owner_id";
        $query_params =
        array(':clinic_owner_id' =>
        $clinic_owner_id);

        $stmt = $this->db->conn_id->prepare($query);
        $result = $stmt->execute($query_params);

        $row = $stmt->fetch();
        return $row['username'];
    }

    public function
    get_all_clinicrequests() {
        $query = "SELECT * FROM
        clinic_request";

```

```

        $stmt = $this->db->conn_id-
>prepare($query);
        $result = $stmt->execute();
        return $stmt->fetchAll();
    }

    public function
approve_clinicrequest($username)
    {
        // update user.status
        $status = 1;
        $this-
>updateUserStatus($username, $status);
        // insert clinic owner then
get_clinic_owner_id
        $this-
>addClinicOwner($username);
        $clinic_owner_id = $this-
>get_clinicownerid($username);
        // get info from
clinic_request then insert to clinic
        $clinic = $this-
>getClinicInfo($username);
        $clinic['clinic_owner_id'] =
$clinic_owner_id;
        $this->addClinic($clinic);
        // delete info from
clinic_request
        $this-
>delete_clinicrequest($username);
    }

    public function
updateUserStatus($username, $status)
    {
        $query = "UPDATE user SET
status = :status WHERE username =
:username";

        $query_params = array(
            ':status' =>
$status,
            ':username' =>
$username
        );

        $stmt = $this->db->conn_id-
>prepare($query);
        $result = $stmt-
>execute($query_params);
    }

    public function
getClinicInfo($username)
    {
        $query = "SELECT * FROM
clinic_request WHERE username = :username";
        $query_params =
array(':username' => $username);

        $stmt = $this->db->conn_id-
>prepare($query);
        $result = $stmt-
>execute($query_params);
        $row = $stmt->fetch();

        return $row;
    }

    public function addClinic($clinic)
    {
        $query = "INSERT INTO clinic(
        clinic_name,
        clinic_owner_id,
        address,
        schedule,
        contact_no,
        email,
        x,
        y,
        status
        )
        VALUES(
            :clinic_name,
            :clinic_owner_id,
            :address,
            :schedule,
            :contact_no,
            :email,
            :x,
            :y,
            :status
        )";

        $query_params = array(
            ':clinic_name' =>
$clinic['clinic_name'],
            ':clinic_owner_id' =>
$clinic['clinic_owner_id'],
            ':address' =>
$clinic['address'],
            ':schedule' =>
$clinic['schedule'],
            ':contact_no' =>
$clinic['contact_no'],
            ':email' =>
$clinic['email'],
            ':x' =>
$clinic['x'],
            ':y' =>
$clinic['y'],
            ':status' =>
1
        );

        $stmt = $this->db->conn_id-
>prepare($query);
        $result = $stmt-
>execute($query_params);
    }

    public function
addClinicOwner($username)
    {
        $query = "INSERT INTO
clinic_owner (username) VALUES (:username)";

        $query_params =
array(':username' => $username);

        $stmt = $this->db->conn_id-
>prepare($query);
        $result = $stmt-
>execute($query_params);
    }

    public function
delete_clinicrequest($username)
    {
        $query = "DELETE FROM
clinic_request WHERE username = :username";
    }

```

```

        $query_params =
array(':username' => $username);

        $stmt = $this->db->conn_id-
>prepare($query);
        $result = $stmt-
>execute($query_params);
    }

    /*****
    *****/
    * Clinic Owner exclusive
    *****/
    methods
    *****/

        public function
get_myclinic_info($clinic_owner_id) {
    $query = "SELECT * FROM
clinic WHERE
clinic_owner_id=:clinic_owner_id";
    $query_params =
array(':clinic_owner_id' =>
$clinic_owner_id);

        $stmt = $this->db->conn_id-
>prepare($query);
        $result = $stmt-
>execute($query_params);

        return $stmt->fetch();
    }

        public function
update_clinicinfo($clinic, $clinic_owner_id)
    {
        $query = "UPDATE clinic SET
clinic_name =
:clinic_name,
address =
:address,
schedule =
:schedule,
contact_no =
:contact_no,
email =
:email,
x = :x,
y = :y
WHERE
clinic_owner_id =
:clinic_owner_id";

        $query_params = array(
':clinic_name' =>
$clinic['clinic_name'],
':address' =>
$clinic['address'],
':schedule' =>
$clinic['schedule'],
':contact_no' =>
$clinic['contact_no'],
':email' =>
$clinic['email'],
':x' =>
$clinic['lat'],
':y' =>
$clinic['lng'],
':clinic_owner_id' =>
$clinic_owner_id);

        $stmt = $this->db->conn_id-
>prepare($query);
        $result = $stmt-
>execute($query_params);

        return $stmt->fetchAll();
    }

        public function
get_clinic_veterinarians($clinic_id) {
    $query = "SELECT DISTINCT
vet_id FROM clinic_vet WHERE
clinic_id=:clinic_id";
    $query_params =
array(':clinic_id' => $clinic_id);

        $stmt = $this->db->conn_id-
>prepare($query);
        $result = $stmt-
>execute($query_params);

        return $stmt->fetchAll();
    }

        public function
get_veterinarian_info($vet_id) {
    $query = "SELECT * FROM
veterinarian WHERE vet_id=:vet_id";
    $query_params =
array(':vet_id' => $vet_id);

        $stmt = $this->db->conn_id-
>prepare($query);
        $result = $stmt-
>execute($query_params);

        return $stmt->fetch();
    }

        public function
search_vet($search_string) {
    $query = "SELECT contact_no,
username, last_name, first_name,
middle_name, email FROM user WHERE
(
username LIKE :condition OR
email LIKE :condition OR
last_name LIKE :condition OR
first_name LIKE :condition
) AND role_id = 3";
    $query_params =
array(':condition' => '%' . $search_string .
'%');

        $stmt = $this->db->conn_id-
>prepare($query);
        $result = $stmt-
>execute($query_params);
        $rows = $stmt->fetchAll();

        $vetresult = "";
        foreach ($rows as $row) {
            $name =
$row['last_name'] . ', ' .
$row['first_name'];
            $name .= ' ' .
$row['middle_name'];
            $email =
$row['email'];

```

```

        $b_username =
'<strong>'.$search_string.</strong>';
        $b_email =
'<strong>'.$search_string.</strong>';
        $final_username =
str_ireplace($search_string, $b_username,
$name);
        $final_email =
str_ireplace($search_string, $b_email,
$email);

        $vetresult .= '<div
class="show" id = "' . $row['username'] .'"
align="left">
                
                <span class="name"
id="test">' . $final_username .
'</span>&nbsp;<br/>'. $final_email . '
&nbsp;<br/>' .
$row['contact_no'] . '<br/>';
        </div>';
    }
    return $vetresult;
}

public function
addVetDB($userRequest) {
    $query = "INSERT INTO
veterinarian(
        username,
        license_no
    )
VALUES(
        :username,
        :license_no
    )";
    $query_params = array(
        ':username' =>
$userRequest['username'],
        ':license_no' =>
$userRequest['license_no']
    );
    $stmt = $this->db->conn_id-
>prepare($query);
    $result = $stmt-
>execute($query_params);
}

public function
addClinicVet($clinic_id, $vet_id, $sched) {
    $query = "INSERT INTO
clinic_vet(
        clinic_id,
        vet_id,
        schedule
    )
VALUES(
        :clinic_id,
        :vet_id,
        :schedule
    )";
    $query_params = array(
        ':clinic_id' =>
$clinic_id,

```



```

        $query_params =
array(':vet_id' => $vet_id);
    } else {
        $query = "SELECT
DISTINCT pet_id FROM consultation_history
WHERE vet_id=:vet_id AND clinic_id =
:clinic_id";
        $query_params =
array(':vet_id' => $vet_id, ':clinic_id' =>
$clinic_id);
    }

$stmt = $this->db->conn_id-
>prepare($query);
$result = $stmt-
>execute($query_params);

$rows = $stmt->fetchAll();

return $rows;
}

public function
search_patient($search_string) {
    $query = "SELECT pet_id,
pet_name, owner_id, specie, breed FROM pet
WHERE pet_name LIKE :pet_name";
    $query_params =
array(':pet_name' => '%' . $search_string .
'%');

    $stmt = $this->db->conn_id-
>prepare($query);
    $result = $stmt-
>execute($query_params);
    $rows = $stmt->fetchAll();

    $patientresult = "";
    foreach ($rows as $row) {
        $owner_username =
$this->get_owner_username($row['owner_id']);
        $owner_profile =
$this->get_profile($owner_username);
        $petname =
$row['pet_name'];
        $b_petname =
'<strong>'.$search_string.</strong>';
        $final_petname =
str_ireplace($search_string, $b_petname,
$petname);

        $patientresult.=
'<div class="show" id="' . $row['pet_id'] .
'" align="left">
        
        <span class="name">' .
$final_petname . '</span>&nbsp;<br/>' .
$row['specie'] . ': ' . $row['breed'] .
'<br/> Owner: ' .
$owner_profile['last_name'] . ', ' .
$owner_profile['first_name'] . '<br/>' .
'</div>';
    }

return $patientresult;
}
}

```

```

public function
get_owner_username($owner_id) {
    $query = "SELECT * FROM
pet_owner WHERE owner_id=:owner_id";
    $query_params =
array(':owner_id' => $owner_id);

    $stmt = $this->db->conn_id-
>prepare($query);
    $result = $stmt-
>execute($query_params);

    $row = $stmt->fetch();

return $row['username'];
}

public function addOwner($username)
{
    $query = "INSERT INTO
pet_owner (username) VALUES (:username)";

    $query_params =
array(':username' => $username);

    $stmt = $this->db->conn_id-
>prepare($query);
    $result = $stmt-
>execute($query_params);

    if ($result) {
        $query = "SELECT MAX(
owner_id ) AS max_id FROM pet_owner";
        $stmt = $this->db-
>conn_id->prepare($query);
        $result = $stmt-
>execute($query_params);
        $row = $stmt->fetch();

        if ($result) {
            return
$row['max_id'];
        }
    }
    return 0;
}

public function addPet($petRequest)
{
    $query = "INSERT INTO pet(
owner_id,
pet_name,
specie,
breed,
birthdate,
gender,
registration_no
)
VALUES(
:owner_id,
:pet_name,
:specie,
:breed,
:birthdate,
:gender,
:registration_no
)";

    $query_params = array(
':owner_id' =>
$petRequest['owner_id'],

```

```

        ':pet_name'           =>
$petRequest['pet_name'],
        ':specie'           =>
$petRequest['specie'],
        ':breed'           =>
$petRequest['breed'],
        ':birthdate'       =>
$petRequest['birthdate'],
        ':gender'          =>
$petRequest['gender'],
        ':registration_no' =>
$petRequest['registry']
    );

    $stmt = $this->db->conn_id-
>prepare($query);
    $result = $stmt-
>execute($query_params);

    if ($result) {
        $query = "SELECT MAX(
pet_id ) AS max_id FROM pet";
        $stmt = $this->db-
>conn_id->prepare($query);
        $result = $stmt-
>execute($query_params);
        $row = $stmt->fetch();

        if ($result) {
            return
$row['max_id'];
        }
    }
    return 0;
}

public function
addConsultation($consultation) {
    $query = "INSERT INTO
consultation_history(
        pet_id,
        vet_id,
        clinic_id,
        consultation_date,
        complaint,
        diagnosis,
        remarks,
        next_consult_date,
        schedule_appt_id
    )
VALUES(
        :pet_id,
        :vet_id,
        :clinic_id,
        :consultation_date,
        :complaint,
        :diagnosis,
        :remarks,
        :next_consult_date,
        :schedule_appt_id
    )";

    $query_params = array(
=> $consultation['pet_id'],
=> $consultation['vet_id'],
=> $consultation['clinic_id'],
=> $consultation['consultation_date'],
=> $consultation['complaint'],
=> $consultation['diagnosis'],
=> $consultation['remarks'],
=> $consultation['next_consult_date'],
=> $consultation['schedule_appt_id']
    );

    $stmt = $this->db->conn_id-
>prepare($query);
    $result = $stmt-
>execute($query_params);
}

        ':complaint'
=> $consultation['complaint'],
        ':diagnosis'
=> $consultation['diagnosis'],
        ':remarks'
=> $consultation['remarks'],
        ':next_consult_date'
=> $consultation['next_consult_date'],
        ':schedule_appt_id'
=> $consultation['schedule_appt_id']
    );

    $stmt = $this->db->conn_id-
>prepare($query);
    $result = $stmt-
>execute($query_params);

    if ($result) {
        $query = "SELECT MAX(
consultation_id ) AS max_cons_id FROM
consultation_history";
        $stmt = $this->db-
>conn_id->prepare($query);
        $result = $stmt-
>execute($query_params);
        $row = $stmt->fetch();

        if ($result) {
            return
$row['max_cons_id'];
        }
    }
    return 0;
}

public function
addMedication($medication) {
    $query = "INSERT INTO
medication(
        pet_id,
        consultation_id,
        medicine,
        dosage,
        med_date
    )
VALUES(
        :pet_id,
        :consultation_id,
        :medicine,
        :dosage,
        :med_date
    )";

    $query_params = array(
=> $medication['pet_id'],
=> $medication['consultation_id'],
=> $medication['medicine'],
=> $medication['dosage'],
=> $medication['med_date']
    );

    $stmt = $this->db->conn_id-
>prepare($query);
    $result = $stmt-
>execute($query_params);
}

```

```

        public function
addVaccination($vaccination) {
    $query = "INSERT INTO
vaccination(
        pet_id,
        consultation_id,
        vaccine_type,
        vaccine_name,
        expiration,
        next_due
    )
VALUES(
        :pet_id,
        :consultation_id,
        :vaccine_type,
        :vaccine_name,
        :expiration,
        :next_due
    )";

    $query_params = array(
=> $vaccination['pet_id'],
=> $vaccination['consultation_id'],
=> $vaccination['vaccine_type'],
=> $vaccination['vaccine_name'],
=> $vaccination['expiration'],
=> $vaccination['next_due']
    );

    $stmt = $this->db->conn_id-
>prepare($query);
    $result = $stmt-
>execute($query_params);
}

    public function
addLaboratory($laboratory) {
    $query = "INSERT INTO
laboratory(
        pet_id,
        consultation_id,
        lab_type,
        lab_findings,
        lab_date
    )
VALUES(
        :pet_id,
        :consultation_id,
        :lab_type,
        :lab_findings,
        :lab_date
    )";

    $query_params = array(
=> $laboratory['pet_id'],
=> $laboratory['consultation_id'],
=> $laboratory['lab_type'],
=> $laboratory['lab_findings'],
=> $laboratory['lab_date']
    );

    $stmt = $this->db->conn_id-
>prepare($query);
    $result = $stmt-
>execute($query_params);

    return $stmt->fetchAll();
}

    );

    $stmt = $this->db->conn_id-
>prepare($query);
    $result = $stmt-
>execute($query_params);
}

    // history = 0 - veterinarian for
approval of appointments
    // history = 1 - appointments history
vet view
    // history = 2 - appointment history
owner view
    public function getAppointments($id,
$history = 0) {
        if ($history == 0)
            $query = "SELECT *
FROM schedule_appt WHERE vet_id = :vet_id
AND date >= DATE(:date) AND confirmed = 0";
            else if ($history == 1)
                $query = "SELECT *
FROM schedule_appt WHERE vet_id = :vet_id
AND date <= DATE(:date) AND confirmed = 1";
            else
                $query = "SELECT *
FROM schedule_appt WHERE owner_id =
:owner_id";

        if ($history == 2) {
            $query_params = array(
                ':owner_id' => $id
            );
        } else {
            $date = date('Y-m-d');
            $query_params = array(
                ':vet_id' => $id,
                ':date' => $date
            );
        }

        $stmt = $this->db->conn_id-
>prepare($query);
        $result = $stmt-
>execute($query_params);

        return $stmt->fetchAll();
}

    /*****
    *****
    *****
    * Pet owner exclusive methods
    *****
    *****
    *****/

    public function get_mypets($owner_id)
{

```

```

        $query = "SELECT * FROM pet
WHERE owner_id=:owner_id";
        $query_params =
array(':owner_id' => $owner_id);

        $stmt = $this->db->conn_id-
>prepare($query);
        $result = $stmt-
>execute($query_params);

        return $stmt->fetchAll();
    }

    public function
update_pet($petRequest)
    {
        $query = "UPDATE pet SET
        pet_name =
:pet_name,
        specie =
:specie,
        breed =
:breed,
        birthdate =
:birthdate,
        gender =
:gender,
        registration_no =
:registration_no
WHERE
        pet_id = :pet_id";

        $query_params = array(
            ':pet_name' =>
$petRequest['pet_name'],
            ':specie' =>
$petRequest['specie'],
            ':breed' =>
$petRequest['breed'],
            ':birthdate' =>
$petRequest['birthdate'],
            ':gender' =>
$petRequest['gender'],
            ':registration_no' =>
$petRequest['registry'],
            ':pet_id' =>
$petRequest['pet_id']
        );

        $stmt = $this->db->conn_id-
>prepare($query);
        $result = $stmt-
>execute($query_params);
    }

    public function get_pet_info($pet_id)
    {
        $query = "SELECT * FROM pet
WHERE pet_id=:pet_id";
        $query_params =
array(':pet_id' => $pet_id);

        $stmt = $this->db->conn_id-
>prepare($query);
        $result = $stmt-
>execute($query_params);

        return $stmt->fetch();
    }

    public function findage($dob)
    {
        $today = getdate();
        $target = explode("-", $dob);

        $b_day = $target[2];
        $b_month = $target[1];
        $b_year = $target[0];

        $t_day = $today['mday'];
        $t_month = $today['mon'];
        $t_year = $today['year'];

        $years = $t_year - $b_year;
        $months = $t_month -
        $b_month;
        $days = $t_day - $b_day;

        if ($months < 0) {
            $years = ($years > 0)
? --$years : $years;
            $months += 12;
        }

        if ($days < 0) {
            if ($months > 0)
                $days += 30;
            else
                $days = 0;
            $months = ($months >
0) ? --$months : $months;
        }

        $age = "";
        $age .= $years;
        $age .= ($years > 1) ? '
years ' : ' year ';
        $age .= $months;
        $age .= ($months > 1) ? '
months ' : ' month ';
        $age .= $days;
        $age .= ($days > 1) ? '
' : ' day ';

        return $age;
    }

    public function
get_consultation_history($pet_id, $vet_id =
0) {
        if ($vet_id == 0) {
            $query = "SELECT *
FROM consultation_history WHERE
pet_id=:pet_id";
            $query_params =
array(':pet_id' => $pet_id);
        }
        else {
            $query = "SELECT *
FROM consultation_history WHERE
pet_id=:pet_id AND vet_id=:vet_id";
            $query_params =
array(':pet_id' => $pet_id, ':vet_id' =>
$vet_id);
        }

        $stmt = $this->db->conn_id-
>prepare($query);
        $result = $stmt-
>execute($query_params);
    }

```

```

        return $stmt->fetchAll();
    }

    public function
    get_consultation_info($consultation_id) {
        $query = "SELECT * FROM
    consultation_history WHERE
    consultation_id=:consultation_id";
        $query_params =
    array(':consultation_id' =>
    $consultation_id);

        $stmt = $this->db->conn_id-
    >prepare($query);
        $result = $stmt-
    >execute($query_params);

        return $stmt->fetch();
    }

    public function
    get_consultation_medications
    ($consultation_id) {
        $query = "SELECT * FROM
    medication WHERE
    consultation_id=:consultation_id";
        $query_params =
    array(':consultation_id' =>
    $consultation_id);

        $stmt = $this->db->conn_id-
    >prepare($query);
        $result = $stmt-
    >execute($query_params);

        return $stmt->fetchAll();
    }

    public function
    get_consultation_laboratory
    ($consultation_id) {
        $query = "SELECT * FROM
    laboratory WHERE
    consultation_id=:consultation_id";
        $query_params =
    array(':consultation_id' =>
    $consultation_id);

        $stmt = $this->db->conn_id-
    >prepare($query);
        $result = $stmt-
    >execute($query_params);

        return $stmt->fetchAll();
    }

    public function
    get_consultation_vaccinations
    ($consultation_id) {
        $query = "SELECT * FROM
    vaccination WHERE
    consultation_id=:consultation_id";
        $query_params =
    array(':consultation_id' =>
    $consultation_id);

        $stmt = $this->db->conn_id-
    >prepare($query);
        $result = $stmt-
    >execute($query_params);

```

```

        return $stmt->fetchAll();
    }

    public function
    search_clinic($search_string) {
        $query = "SELECT * FROM
    clinic WHERE clinic_name LIKE :clinic_name";
        $query_params =
    array(':clinic_name' => '%' . $search_string
    . '%');

        $stmt = $this->db->conn_id-
    >prepare($query);
        $result = $stmt-
    >execute($query_params);
        $rows = $stmt->fetchAll();

        $result = "";
        foreach ($rows as $row) {

            $clinic =
    $row['clinic_name'];
            $b_clinic =
    '<strong>'.$search_string.</strong>';
            $clinic_info =
    str_ireplace($search_string, $b_clinic,
    $clinic);

            $result .= '<div
    class="show" id="' . $row['clinic_id'] . '"
    align="left">

                <span class="name">' .
    $clinic_info . '</span>&nbsp;<br/>' .
    $row['schedule'] . '<br/>' .
    $row['contact_no'] . '<br/>' .
                '</div>';
        }

        return $result;
    }

    public function
    add_appointment($appointment) {
        $query = "INSERT INTO
    schedule_appt (

        clinic_id,
        vet_id,
        owner_id,
        pet_id,
        date,
        time,
        remarks,
        confirmed

    )
    VALUES(

        :clinic_id,
        :vet_id,
        :owner_id,
        :pet_id,
        :date,
        :time,
        :remarks,
        :confirmed

    )";
        $query_params = array(

```

```

        ':clinic_id' =>
$appointment['clinic_id'],
        ':vet_id' =>
$appointment['vet_id'],
        ':owner_id' =>
$appointment['owner_id'],
        ':pet_id' =>
$appointment['pet_id'],
        ':date' =>
$appointment['date'],
        ':time' =>
$appointment['time'],
        ':remarks' =>
$appointment['remarks'],
        ':confirmed' =>
$appointment['confirmed']
    );
    echo ($appointment['time']);
    $stmt = $this->db->conn_id-
>prepare($query);
    $result = $stmt-
>execute($query_params);
    }

    public function
    confirm_appointment($appointment_id,
$confirmed, $remarks = "") {
        if ($confirmed === 1) {
            $query = "UPDATE
schedule_appt SET
                confirmed = :confirmed
                WHERE
                    schedule_appt_id =
:schedule_appt_id";

            $query_params = array(
                ':confirmed'
=> $confirmed,
                ':schedule_appt_id' =>
$appointment_id
            );
        } else {
            $query = "UPDATE
schedule_appt SET
                remarks
= :remarks,
                confirmed = :confirmed
                WHERE
                    schedule_appt_id =
:schedule_appt_id";

            $query_params = array(
                ':remarks'
=> $remarks,
                ':confirmed'
=> $confirmed,
                ':schedule_appt_id' =>
$appointment_id
            );
        }
    }

```

```

        $stmt = $this->db->conn_id-
>prepare($query);
        $result = $stmt-
>execute($query_params);
    }
    ?>

```

Listing 3: views/home.php

```

<div class="homepage">
    <h2>Hello <?php echo
$this->session-
>userdata('username');?>!</h2>
    <h3>Welcome to the
SpotAVet Website.</h3>
    <p>
        SpotAVet is a
Web-based system application that provides
support for pet owners to get the service
they need for their pets. With an extension
to Android device that allows user to search
veterinary clinic base on location, the
system becomes more flexible.
    </p>
    <p>
        By using the
Android extension of this application, pet
owners, registered or not registered, could
already have the idea on where the clinic
is, as well as its schedule and its
veterinarians.
    </p>
    <p>
        Please
download the SpotAvet for Android to start
searching near clinics using your Android
device.
    </p>
</div>

```

Listing 4: views/login.php

```

<div id="login">
    <h2>Login</h2>
    <?php
        echo
form_open('main/verify_login');
        ?>
        <table width="403"
border="0">
            <tr>
                <td
width="73"
class="txt_fields"><strong>Username:</strong>
<</td>
                <td
width="320"><input type="text"
name="username" id="username"
placeholder="Username" class="text_box"
style="width:300px; height:25px;"/></td>
            </tr>
            <tr>
                <td
class="txt_fields"><strong>Password:</strong>
<</td>

```

```

        <td><input type="password"
name="password" id="password"
placeholder="Password" class="text_box"
style="width:300px; height:25px;"/></td>
        </tr>
        <tr>
            <td>
                <div class="error" style="width:
310px; display:<?php echo $status == "" ?
"none" : "block"; ?> >
                <font size=2><?php echo
$status;?></font>
            </div>
        </td>
    </tr>
    <tr>
        <td>
            <div class="txt_fields">&nbsp;</div>
        </td>
        <td>
            <div class="txt_fields">&nbsp;</div>
            <input type="submit"
name="button" id="button" value="Login"
class="search_button" style="height:40px;"/>&nbsp;</td>
        </tr>
    </table>
</form>
</div>

```

Listing 5: views/request_account.php

```

<script>
    $(document).ready(function() {
        // Basic usage
        $(".placepicker").each(function() {
            var placepicker =
            $(this).placepicker({
                placeChanged:
                function(place) {
                    var
                    location = this.getLocation();
                    document.getElementById('lat').value
                    = location['latitude'] ;
                    document.getElementById('lng').value
                    = location['longitude'] ;
                    console.log("place changed: ",
                    place.formatted_address,
                    this.getLocation());
                }
            }).data('placepicker');
        });
    }); // END document.ready
</script>
<style>

```

```

        .placepicker-map {
            width: 100%;
            height: 300px;
        }
        .field_label {
            width: 200px;
            height: 30px;
        }
        .field_text {
            width: 700px;
            height: 30px;
        }
    </style>
    <div id="registration" class="border"
style="width: 940px; ">
        <p><?php echo "<h1>Registration
Form</h1>" ?></p>
        <?php
            echo
            form_open('main/submit_request');
            ?>
            <fieldset>
                <legend>Clinic Owner
                Information</legend>
                <table width="800"
                border="0">
                    <tr>
                        <td
                        class="field_label">Register As:</td>
                        <td
                        class="field_text">
                            <select
                            name="role" class="txt_fields txt_width"
                            style="height:30px; width:400px;" id="role"
                            disabled>
                                <option value="clinic">Clinic
                                Owner</option>
                            </select>
                        </td>
                    </tr>
                    <tr>
                        <td
                        class="field_label">*Username:</td>
                        <td
                        class="field_text">
                            <input
                            name="id" type="hidden" class="txt_fields
                            txt_width" id="id" placeholder="Username"
                            />
                            <input
                            name="username" value="<?php echo
                            set_value('username'); ?>" type="text"
                            class="txt_fields txt_width" id="username"
                            placeholder="Username" />
                            <?php
                            echo form_error('username', '<div
                            class="error">', '</div>'); ?>
                        </td>
                    </tr>
                    <tr>
                        <td
                        class="field_label">*Password:</td>
                        <td
                        class="field_text">

```

```

        <input
name="password" value="<?php echo
set_value('password'); ?>" type="password"
class="txt_fields txt_width" id="password"
placeholder="Password"/>
        </tr>
        <tr>
        <td>
        class="field_label">*Middle Name:</td>
        <td>
        class="field_text">
        <input
name="middle_name" value="<?php echo
set_value('middle_name'); ?>" type="text"
class="txt_fields txt_width"
id="middle_name" placeholder="Middle Name"/>
        <?php echo
form_error('middle_name', '<div
class="error">', '</div>'); ?>
        </td>
        </tr>
        <tr>
        <td>
        class="field_label">*Confirm Password:</td>
        <td>
        class="field_text">
        <input
name="confirm_password" value="<?php echo
set_value('confirm_password'); ?>"
type="password" class="txt_fields txt_width"
id="confirm_password" placeholder="Confirm
Password"/>
        <?php echo
form_error('confirm_password', '<div
class="error">', '</div>'); ?>
        </td>
        </tr>
        <tr>
        <td>
        class="field_label">*Email:</td>
        <td>
        class="field_text">
        <input
name="email" value="<?php echo
set_value('email'); ?>" type="text"
class="txt_fields txt_width" id="email"
placeholder="Email"/>
        <?php echo
form_error('email', '<div class="error">',
'</div>'); ?>
        </td>
        </tr>
        <tr>
        <td>
        class="field_label">*Last Name:</td>
        <td>
        class="field_text">
        <input
name="last_name" value="<?php echo
set_value('last_name'); ?>" type="text"
class="txt_fields txt_width" id="last_name"
placeholder="Last Name"/>
        <?php echo
form_error('last_name', '<div
class="error">', '</div>'); ?>
        </td>
        </tr>
        <tr>
        <td>
        class="field_label">*First Name:</td>
        <td>
        class="field_text">
        <input
name="first_name" value="<?php echo
set_value('first_name'); ?>" type="text"
class="txt_fields txt_width" id="first_name"
placeholder="First Name"/>
        <?php echo
form_error('first_name', '<div
class="error">', '</div>'); ?>
        </td>
        </tr>
    </table>
</fieldset>
</legend>Clinic
Information</legend>
<table width="800"
border="0">
    <tr>
        <td>
        class="field_label">*Clinic Name:</td>
        <td>
        class="field_text">
        <input
name="clinic_name" value="<?php echo
set_value('clinic_name'); ?>" type="text"
class="txt_fields txt_width"
id="clinic_name" placeholder="Clinic Name"/>
        <?php echo
form_error('clinic_name', '<div
class="error">', '</div>'); ?>
        </td>
        </tr>
        <tr>
        <td>
        class="field_label">*Clinic Address:</td>
        <td>
        class="field_text">
        <input
name="clinic_addr" value="<?php echo
set_value('clinic_addr'); ?>" type="text"
class="txt_fields txt_width"
id="clinic_addr" placeholder="Clinic
Address"/>
        <?php echo
form_error('clinic_addr', '<div
class="error">', '</div>'); ?>
        </td>
        </tr>
    </table>

```



```

        <tr>
            <td
class="field_label">*Operating Hours:</td>
            <td
class="field_text">
                <input
name="clinic_hours" value="<?php echo
set_value('clinic_hours'); ?>" type="text"
class="txt_fields txt_width"
id="clinic_hours" placeholder="(Mon ~ Fri ::
8:00 AM ~ 5:00 PM)"/>
                <?php echo
form_error('clinic_hours', '<div
class="error">', '</div>'); ?>
            </td>
        </tr>
    </table>
</fieldset>

    <br /><br />
</fieldset>
<legend>Clinic Location
Details</legend>

class="col-md-6">
    <div
class="placepicker" id="placepicker"
class="placepicker" data-map-container-
id="collapseOne" value = "<?php echo
set_value('placepicker'); ?>" />
    <div
id="collapseOne" class="collapse">

        <div class="placepicker-map"></div>
    </div>

        Lat: <input name="lat"
type="text" id="lat" style="height:25px"
readonly />
        Lng: <input name="lng"
type="text" id="lng" style="height:25px"
readonly />
        <?php echo
form_error('placepicker', '<div
class="error">', '</div>'); ?>
    </fieldset>
    <br />
    <table width="920"
border="0">
        <tr>
            <td
class="field_label">&nbsp;</td>
            <td><input
name="register_button" type="submit"
class="search_button" id="register_button"
value="Register!" /></td>
        </tr>
    </table>
</form>
</div>

```

Listing 6: panels/header.php

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">

```

```

<?php
    $this->load->helper('url');
?>

    <head>
        <meta http-equiv="Content-
Type" content="text/html; charset=utf-8" />
        <script
type="text/javascript" src="<?php echo
base_url("resources/js/jquery-
1.11.3.min.js");?>"></script>
        <script
type="text/javascript"
src="https://maps.googleapis.com/maps/api/js
?sensor=true&libraries=places&region=PH&v=3"
></script>
        <script
type="text/javascript" src="<?php echo
base_url("resources/js/jquery.placepicker.js
");?>"></script>

        <link rel="stylesheet"
type="text/css" href="<?php echo
base_url("resources/css/style.css");?>"
media="screen"/>

        <title><?php
if(isset($title)) { echo $title; } else {
echo "SpotAVet"; } ?></title>
    </head>
    <body>

        <div id="header">
            <div class="section">
                <div class="logo">
                    <a href=<?php
echo "\"\" . base_url("index.php/main/home")
. "\""; ?>>SpotAVet</a>
                </div>
                <ul>
                    <?php if($this-
>session->userdata('isLoggedIn') == FALSE) {
?>
                        <li
class="selected"><a href=<?php echo "\"\" .
base_url("index.php/main/home") . "\"";
?>>Home</a></li>
                        <li><a
href=<?php echo "\"\" .
base_url("index.php/main/login") . "\"";
?>>Login</a></li>
                        <li><a
href=<?php echo "\"\" .
base_url("index.php/main/request_account") .
"\""; ?>>Register Clinic</a></li>
                            <?php } if ($this-
>session->userdata('user_role') == 1) { ?>
                                <li><a href=<?php echo "\"\"
. base_url("index.php/main/manage_clinics")
. "\""; ?>>View Clinics</a></li>
                                    <li
class="selected"><a href=<?php echo "\"\" .
base_url("index.php/main/clinic_requests") .
"\""; ?>>Clinic Requests</a></li>
                                        <?php } else if
($this->session->userdata('user_role') == 2)
{ ?>
                                            <li><a
href=<?php echo "\"\" .

```

```

base_url("index.php/main/view_profile") .
"\\""; ?>>My Profile</a></li>
    <li><a
href=<?php echo "\\" .
base_url("index.php/main/view_clinic") . "/"
. $this->session-
>userdata('clinic_owner_id') . "\\""; ?>>My
Clinic</a></li>
    <li><a href=<?php echo "\\"
.
base_url("index.php/main/view_veterinarians")
) . "\\""; ?>>My Veterinarians</a></li>

    <?php } else if
($this->session->userdata('user_role') == 3)
{ ?>
    <li><a
href=<?php echo "\\" .
base_url("index.php/main/view_profile") .
"\\""; ?>>My Profile</a></li>
    <li><a href=<?php echo "\\"
. base_url("index.php/main/view_clinics") .
"\\""; ?>>My Clinics</a></li>
    <li><a href=<?php echo "\\"
. base_url("index.php/main/view_patients") .
"\\""; ?>>My Patients</a></li>
    <li><a
href=<?php echo "\\" .
base_url("index.php/main/appointments") .
"\\""; ?>>My Appointments</a></li>

    <?php } else if
($this->session->userdata('user_role') == 4)
{ ?>
    <li><a href=<?php echo "\\"
. base_url("index.php/main/view_profile") .
"\\""; ?>>My Profile</a></li>
    <li><a href=<?php echo "\\"
. base_url("index.php/main/view_pets") .
"\\""; ?>>My Pets</a></li>
    <li><a
href=<?php echo "\\" .
base_url("index.php/main/appointments") .
"\\""; ?>>Appointments</a></li>
    <?php } ?>
    <?php if($this-
>session->userdata('isLoggedIn') == TRUE) {
?>
    <li><a href=<?php echo "\\" .
base_url("index.php/main/logout") . "\\"";
?>>Log out</a></li>
    <?php } ?>
</ul>
    <?php if($this-
>session->userdata('isLoggedIn') == TRUE) :
?>
    <div
style="position:absolute; top:0; right:0;">
        You are
        currently logged in as <strong><?php echo
$this->model_home-
>get_usertype();?></strong>|
    </div>
    <?php endif; ?>
</div>
</div>
<div class="content">

```

Listing 7: panels/sidebar-profile.php

```

    <div class="inner
user_design_1" id="left_content"
style="width:250px; background:#E5EEFE; ">
        <h3>PROFILE</h3>
        <p>
" width="200" height="250" /></p>
        <hr>
    </div>

```

Listing 8: panels/sidebar-clinic.php

```

    <div class="inner
user_design_1" id="left_content"
style="width:250px; background:#E5EEFE; ">
        <h3>LOCATION MAP</h3>
        <div id="map-canvas"
style="width:250px;height:200px;"></div>
        <hr>
    </div>
<script>
    var map;
    function initialize() {
        var myLatlng = new
google.maps.LatLng(<?= $clinic['x']?>, <?=
$clinic['y']?>);
        var mapOptions = {
            zoom: 14,
            center: myLatlng,
            disableDefaultUI: true
        };
        map = new
google.maps.Map(document.getElementById('map-
-canvas'),
            mapOptions);
        var marker = new
google.maps.Marker({
            position: myLatlng,
            map: map,
            title: 'Hello World!'
        });
        google.maps.event.addListener(wind
ow, 'load', initialize);
    }
</script>
<style>
    #map-canvas {
        height: 100%;
        margin: 0px;
        padding: 0px
    }
</style>

```

Listing 9: panels/sidebar-pet.php

```

    <div class="inner
user_design_1" id="left_content"
style="width:200px; background:#E5EEFE; ">

```

```

                <h3>PET</h3>
                <p> "
width="180" height="220" /></p>
                <hr>
            </div>

```

Listing 10: panels/footer.php

```

            </div>
            <div id="clear"></div>
            <div id="footer">
                <div>
                    <p>
                        cjgaleng &copy; 2015 | All rights
reserved.
                    </p>
                </div>
            </div>
        </body>
</html>

```

Listing 11: content/add-appointment.php

```

            <div id="right_content"
class="right_design border"
style="width:700px;">
                <?php
                    echo
form_open('main/add_appointment');
                    date_default_timezone_set("Asia/Kuala
_Lumpur");
                    ?>
                    <h3> APPOINTMENT
REQUEST </h3>
                    <div id="clear"></div>
                    <fieldset>
                        <input
type="hidden" name="details" id="details"
value="<?= set_value('details') ?>"/>
                        <div
id="search_field">
                            <div
class="content_search">
                                <input type="text" name="search"
class="search" id="search"
placeholder="Enter name to begin search."
value="<?= set_value('search') ?>"
autocomplete="off"/>
                                <select name="search_option"
id="search_option" class="search_option">
                                    <option value ="Clinic" <?php echo
set_select('search_option', 'Clinic', TRUE);
?>>Clinic</option>
                                    <option value ="Veterinarian" <?php
echo set_select('search_option',
'Veterinarian'); ?> >Veterinarian</option>
                                </select><em>*</em>

```

```

                <?php echo
form_error('search_option', '<div
class="error">', '</div>'); ?>
            </div id="search_result"></div>
            <?php echo form_error('search', '<div
class="error">', '</div>'); ?>
        </div>
            </div>
            </fieldset>
            <br> </br>
            <fieldset>
                <table>
                    <tr>
                        <td>&nbsp;   </td>
                    </tr>
                    <tr>
                        <td
class="vet" style="width:120px; text-
align:right">
                            <label><font size="2">VETERINARIAN:
</font></label>
                        </td>
                        <td
colspan="5" class="vet">
                            <select name="vet_option"
id="vet_option" class="vet_option"
style="height:30px; width:475px">
                                <!-- for loop -->
                            </select><em>*</em>
                        </td>
                    </tr>
                    <tr>
                        <td
class="clinic" style="display: none; text-
align:right">
                            <label><font size="2">CLINIC NAME:
</font></label>
                        </td>
                        <td
colspan="5" class="clinic" style="display:
none">
                            <select name="clinic_option"
id="clinic_option" class="clinic_option"
style="height:30px; width:475px">
                                <!--for loop clinic -->
                            </select><em>*</em>
                        </td>
                    </tr>
                    <tr>
                        <td
class="error">', '</div>'); ?>
                    </td>
                    <td
class="clinic" style="display: none; text-
align:right">
                        </td>
                    </tr>
                    <tr>
                        <td>&nbsp;   </td>

```

```

colspan="5">
    <div class="vet" style="height:80px;
width:470px">
    </div>
    <div class="clinic" style="display:
none; height:80px; width:470px">
    </div>
    </td>
    </tr>
    <tr>
    style="text-align:right">
    <td>
    <label><font size="2">PET NAME:
</font></label>
    </td>
    <td>
    <select name="pet_option"
id="pet_option" class="pet_option"
style="height:30px; width:100px">
    <?php foreach($pets as $pet) : ?>
    <option value ="<?php echo
$pet['pet_id'] ?>" <?php echo
set_select('pet_option', $pet['pet_name']);
?><?=$pet['pet_name'] ?></option>
    <?php endforeach; ?>
    </select><em>*</em>
    <?php echo form_error('pet_option',
'<div class="error">', '</div>'); ?>
    </td>
    <td>
    style="text-align:right">
    <label><font size="2">DATE:
</font></label>
    </td>
    <td>
    <input id="sched_date" type="date"
name="sched_date" class="input_field"
value="<?=$set_value('sched_date', date('Y-
m-d', strtotime("+ 1 day"))); ?>"
style="height:30px; width:125px"><em>*</em>
    <?php echo form_error('sched_date',
'<div class="error">', '</div>'); ?>
    </td>
    <td>
    style="text-align:right">
    <label><font size="2">TIME:
</font></label>
    </td>
    <td>
    <input id="sched_time" type="time"
name="sched_time" class="input_field"
value="<?=$set_value('sched_time',
date('H:i:s')); ?>" style="height:30px;
width:120px"><em>*</em>
    <?php echo form_error('sched_time',
'<div class="error">', '</div>'); ?>
    </td>
    </tr>
    <tr>
    <td>&nbsp;   </td>
    <td>
    style="text-align:right">
    <label><font size="2">REASON FOR
VISIT: </font></label>
    </td>
    <td>
    colspan="5">
    <input type="text" name="reason"
class="input_field" value="<?=$
set_value('reason'); ?>" style="height:30px;
width:475px" placeholder="State reason for
appointment.">
    <em>*</em>
    <?php echo form_error('reason', '<div
class="error">', '</div>'); ?>
    </td>
    </tr>
    </fieldset>
    <table>
    <tr>
    <td><input type="submit"
name="button" id="button" value="Submit
Request" class="search_button"
style="width:120px; height:40px;"
/>&nbsp;  </td>
    </tr>
    </table>
    </div>
    <script language="Javascript">
    $(function(){
        $(".search").keyup(function()
        {
            var searchid =
$(this).val();
            var dataString =
'search='+ searchid;
            dataString =
dataString + ":@" + $( "#search_option
option:selected" ).text();
            if(searchid != '')
            {
                $.ajax({
                    type:
"POST",
                    url:
<?php echo "\" .
base_url("index.php/main/search") . "\"";
?>,

```

```

dataString,
false,
    success: function(html)
    {
        $("#search_result").html(html).show();
    };
    } return false;
});
$(document).on("click",
"#search_result", function(e){
    var $clicked =
$(e.target);
        var $name =
$clicked.find('.name').html();
        var decoded =
$("<div/>").html($name).text();
        $('#search').val(decoded);
    });
    $(document).on("click",
".show", function(e) {
        setDetails(this.id);
    });
    $(document).on("click",
function(e) {
        var $clicked =
$(e.target);
        if (!$clicked.hasClass("search")){
            jQuery("#search_result").fadeOut();
        }
    });
    $('#search').click(function(){
        jQuery("#search_result").fadeIn();
    });
    $('#search_option').on('change',
function() {
        if (this.value ==
"Veterinarian") {
            $('.vet').hide();
            $('.clinic').show();
            $(".clinic_option")
            .find('option')
            .remove()
            .end()
            .append('<option value=""></option>');
        } else {
data:
cache:
$('.vet').show();
$('.clinic').hide();
$(".vet_option")
.find('option')
.remove()
.end()
.append('<option value=""></option>');
};
$("#search_result")
.find('div')
.remove()
.end()
;
$('#search').val('');
$("div.vet")
.find('div')
.remove()
.end()
;
$("div.clinic")
.find('div')
.remove()
.end()
;
$('#details').val("");
});
function setDetails(myValue)
{
    $('#details').val(myValue)
    .trigger('change');
}
$('#details').on('change',
function() {
        if ($("#search_option
option:selected").text() == "Clinic") {
            var clinic_id
= $('#details').val();
            $.ajax({
                type:
"POST",
                url:
<?php echo "" .
base_url("index.php/main/getVetList") .
""; ?>,
                data: {
                    clinic_id: clinic_id },
                cache:
false,
                success: function(html)
                {
                    $(".vet_option")
                    .find('option')
                    .remove()

```

```

        .end()
        .append(html)
        .trigger('change')
        ;
    });
    } else {
        var
vet_username = $('#details').val();
        $.ajax({
            type:
"POST",
            url:
<?php echo "\" .
base_url("index.php/main/getClinicList") .
"\""; ?>,
            data: {
vet_username: vet_username },
            cache:
false,
            success: function(html)
            {
                $(".clinic_option")
                .find('option')
                .remove()
                .end()
                .append(html)
                .trigger('change')
                ;
            }
        });
    }
}

$('.vet_option').on('change',
function() {
    var vet_username =
$('#vet_option').val();
    var clinic_id =
$('#details').val();
    $.ajax({
        type: "POST",
        url: <?php
echo "\" .
base_url("index.php/main/getVetDetails") .
"\""; ?>,
        data: {
vet_username: vet_username,
clinic_id: clinic_id
        },
        cache: false,
        success:
function(html)
        {
            $("div.vet")
            .find('div')
            .remove()
            .end()
            .append(html)
            ;
        }
    });
}

$('#sched_date').on('change',
function() {
    if (!check_date())
    {
        alert("Appointment date must be at
least a day from now");
        $('#sched_date').val("<?php echo
date('Y-m-d', strtotime("+ 1 day")) ?>");
        $('#sched_time').val("<?php echo
date('H:i:s') ?>");
    }
}

$('#sched_time').on('change',
function() {
}

```

```

function check_date() {
    var usrdate =
    $('#sched_date').val();
    var user_temp = new
    Array();

    user_temp =
    usrdate.split("-");
    var user_year =
    user_temp[0];
    var user_month =
    user_temp[1];
    var user_day =
    user_temp[2];

    var curdate = "<?php
    echo date('Y-m-d') ?>";
    var cur_temp = new
    Array();
    cur_temp =
    curdate.split("-");
    var cur_year =
    cur_temp[0];
    var cur_month =
    cur_temp[1];
    var cur_day =
    cur_temp[2];

    if (user_year >
    cur_year) {
        return true;
    } else {
        if (user_month
        > cur_month) {
            if
            (user_year == cur_year) {
                return true;
            } else
            {
                return false;
            }
        } else {
            if
            (user_day > cur_day) {
                if ((user_month >= cur_month) &&
                (user_year >= cur_year)){
                    return true;
                } else {
                    return false;
                }
            }
        }
    }
    return false;
}

function check_time() {
    var usertime =
    $('#sched_time').val();
    var user_temp = new
    Array();

    usertime.split(":");
    user_temp =
    user_temp[0];
    user_temp[1];
    user_temp[2];

    var curtime = "<?php
    echo date('H:i:s') ?>";
    var cur_temp = new
    Array();
    cur_temp =
    curtime.split(":");
    var cur_hr =
    cur_temp[0];
    var cur_min =
    cur_temp[1];
    var cur_ss =
    cur_temp[2];

    if (cur_hr + 1 <=
    user_hr) {
        return true;
    } else if (cur_hr + 1 ==
    user_hr) {
        if (cur_min +
        1 <= user_min) {
            return true;
        }
        return false;
    }
}

</script>

<style type="text/css">
    .content_search{
        width: 600px;
        margin: 0 auto;
    }
    #search
    {
        width:420px;
        border:solid 1px #000;
        padding:10px;
        font-size:14px;
    }
    #search_option
    {
        width:130px;
        border:solid 1px #000;
        padding:10px;
        font-size:14px;
    }
    #search_result
    {
        position:absolute;
        width:420px;
        max-height:300px;
        padding:10px;
        display:none;
        margin-top:-1px;
        border-top:0px;
        overflow-y:auto;
        border:1px #CCC solid;
        background-color: white;
    }
}

```

```

}
.show
{
padding:10px;
border-bottom:1px #999 dashed;
font-size:15px;
height:50px;
}
.show:hover
{
background:#4c66a4;
color:#FFF;
cursor:pointer;
}
em {
color: rgb(255,0,0);
}
</style>

```

Listing 12: content/add-patient.php

```

<div id="right_content"
class="right_design border"
style="width:700px;"
    <?php
        echo
form_open('main/add_owner');
    >
    <div id="clear"></div>
    <fieldset>
    <legend>New Patient
Record</legend>
        <input id="r1"
type="radio" onchange="hideB(this)"
name="aorb" value="1" <?php echo
set_radio('aorb', '1', TRUE); ?> > Select
from registered pet list |
        <input id="r2"
type="radio" onchange="hideA(this)"
name="aorb" value="2" <?php echo
set_radio('aorb', '2'); ?>> Add new
patient's owner account
        <input
type="hidden" name="hidden_info"
id="hidden_info" value="<?=?
set_value('hidden_info', 0) ?>"/>
        <br> </br>
        <div id="A">
        <div
class="content_searchpatient">
            <input type="text"
name="searchpatient" class="searchpatient"
id="searchpatient" placeholder="Enter (pet
name) to search." value="<?=?
set_value('searchpatient') ?>"
autocomplete="off" />
            <div id="result_searchpatient"></div>
            </div>
            <div id="B"
style="display: none">
                <h3>Pet
Owner Information</h3>
                <table
width="611" border="0">
                    <tr>
                        <td>Register As:</td>
                        <td>
                            <select name="role"
class="txt_fields txt_width"
style="height:30px; width:408px;" id="role"
disabled>
                                <option
value="clinic">Pet Owner</option>
                            </select>
                        </td>
                    </tr>
                    <tr>
                        <td width="129">*Username:</td>
                        <td>
                            <input type="text"
name="username" id="username" value="<?=?
set_value('username') ?>" class="txt_fields"
/>
                            <?php echo
form_error('username', '<div
class="error">', '</div>'); ?>
                        </td>
                    </tr>
                    <tr>
                        <td>*Password:</td>
                        <td><input type="password"
name="password" id="password" value="<?=?
set_value('password') ?>" class="txt_fields"
/>
                            <?php echo form_error('password',
'<div class="error">', '</div>'); ?>
                        </td>
                    </tr>
                    <tr>
                        <td>*Confirm Password:</td>
                        <td><input type="password"
name="confirm_password"
id="confirm_password" value="<?=?
set_value('confirm_password') ?>"
class="txt_fields" />
                            <?php echo
form_error('confirm_password', '<div
class="error">', '</div>'); ?>
                        </td>
                    </tr>
                </table>
            </div>
        </tr>
    </div>

```



```

<tr>

<td>*Email Address:</td>

<td><input type="text" name="email"
id="email" value="<?= set_value('email') ?>"
class="txt_fields" />

<?php echo form_error('email', '<div
class="error">', '</div>'); ?>

</td>

</tr>

<tr>

<td>*Last Name:</td>

<td><input type="text"
name="last_name" id="last_name" value="<?=
set_value('last_name') ?>"
class="txt_fields" />

<?php echo form_error('last_name',
'<div class="error">', '</div>'); ?>

</td>

</tr>

<tr>

<td>*First Name:</td>

<td><input type="text"
name="first_name" id="first_name" value="<?=
set_value('first_name') ?>"
class="txt_fields" />

<?php echo form_error('first_name',
'<div class="error">', '</div>'); ?>

</td>

</tr>

<tr>

<td>*Middle Name:</td>

<td><input type="text"
name="middle_name" id="middle_name"
value="<?= set_value('middle_name') ?>"
class="txt_fields" />

<?php echo form_error('middle_name',
'<div class="error">', '</div>'); ?>

</td>

</tr>

<tr>

<td>*Contact No:</td>

<td><input type="text"
name="contact_no" id="contact_no" value="<?=

```

```

set_value('contact_no') ?>"
class="txt_fields" />

<?php echo form_error('contact_no',
'<div class="error">', '</div>'); ?>

</td>

</tr>

</table>

<h3>Pet
Information</h3>

<table
width="611" border="0">

<tr>

<td width="129">*Pet Name: </td>

<td>

<input type="text"
name="pet_name" id="pet_name" value="<?=
set_value('pet_name') ?>" class="txt_fields"
/>

<?php echo
form_error('pet_name', '<div
class="error">', '</div>'); ?>

</td>

</tr>

<tr>

<td>*Specie: </td>

<td><input type="text"
name="specie" id="specie" value="<?=
set_value('specie') ?>" class="txt_fields"
/>

<?php echo
form_error('specie', '<div class="error">',
'</div>'); ?>

</td>

</tr>

<tr>

<td>*Breed: </td>

<td>

<input type="text"
name="breed" id="breed" value="<?=
set_value('breed') ?>" class="txt_fields" />

<?php echo
form_error('breed', '<div class="error">',
'</div>'); ?>

</td>

</tr>

```

```

<tr>
    <td>*Birth Date: </td>
    <td><input type="date"
name="birthday" id="birthday" value="<?=
set_value('birthday') ?>" class="txt_fields"
/>
    </td>
</tr>
<tr>
    <td>*Gender: </td>
    <td>
        <select name="gender"
id="gender" class="txt_fields"
style="height:30px; width:408px;" >
            <option value =""
<?php echo set_select('gender', '', TRUE);
?></option>
            <option value ="Male"
<?php echo set_select('gender', 'Male');
?>>Male</option>
            <option value
="Female" <?php echo set_select('gender',
'Female'); ?> >Female</option>
        </select>
        <?php echo
form_error('gender', '<div class="error">',
'</div>'); ?>
    </td>
</tr>
<tr>
    <td>Registration No.: </td>
    <td><input type="text"
name="registry" id="registry" value="<?=
set_value('registry') ?>" class="txt_fields"
placeholder='Registration number (if
applicable)' />
        <?php echo form_error('registry',
'<div class="error">', '</div>'); ?>
    </td>
</tr>
</table>
</div>
</fieldset>
<br> </br>
<fieldset>
<table>
    <tr>
        <td style="text-align:right"
class="clinic" style="text-align:right">
            <label><em>*</em>Clinic: </label>
        </td>
        <td
class="clinic">
            <select name="clinic_option"
id="clinic_option" class="clinic_option"
style="height:30px; width:495px">
                <?php foreach ($clinics as $clinic) :
                ?>
                    <?php $clinic_info = $this-
>model_home-
>get_clinic_info($clinic['clinic_id']); ?>
                    <option value="<?php echo
$clinic_info['clinic_id']?>'><?php echo
$clinic_info['clinic_name'] ?></option>
                    <!--for loop clinic -->
                    <?php endforeach; ?>
                </select>
                <?php echo
form_error('clinic_option', '<div
class="error">', '</div>'); ?>
            </td>
        </tr>
</table>
</fieldset>
<br> </br>
<div
id="consultation">
    <fieldset>
        <legend>Consultation Details</legend>
        <table
width="611" border="0">
            <tr>
                <td width="129">*Consultation Date:
            </td>
            <td>
                <input type="date"
name="consultation_date"
id="consultation_date" value="<?=
set_value('consultation_date') ?>"
class="txt_fields" />
                <?php echo
form_error('consultation_date', '<div
class="error">', '</div>'); ?>
            </td>
        </tr>
    </table>
</div>

```

```

        <td>*Chief Complaints: </td>
        <td>
            <input type="text"
name="complaints" id="complaints" value="<?=
set_value('complaints') ?>"
class="txt_fields" />
            <?php echo
form_error('complaints', '<div
class="error">', '</div>'); ?>
        </td>
    </tr>
    <tr>
        <td>*Diagnosis/Findings: </td>
        <td>
            <input type="text"
name="diagnosis" id="diagnosis" value="<?=
set_value('diagnosis') ?>"
class="txt_fields" />
            <?php echo
form_error('diagnosis', '<div
class="error">', '</div>'); ?>
        </td>
    </tr>
    <tr>
        <td>Other Remarks: </td>
        <td>
            <input type="text"
name="remarks" id="remarks" value="<?=
set_value('remarks') ?>" class="txt_fields"
/>
            <?php echo
form_error('remarks', '<div class="error">',
'</div>'); ?>
        </td>
    </tr>
    <tr>
        <td>Next Consultation Date: </td>
        <td>
            <input type="date"
name="next_consultation"
id="next_consultation" value="<?=
set_value('next_consultation') ?>"
class="txt_fields" />
            <?php echo
form_error('next_consultation', '<div
class="error">', '</div>'); ?>
        </td>
    </tr>
    <tr>
        <td>*Date Given: </td>
        <td>
            <input type="date"
name="medication_date" id="medication_date"
value="<?= set_value('medication_date') ?>"
class="txt_fields" />
        </td>
    </tr>
    <tr>
        <td>*Medicine: </td>
        <td>
            <input type="text"
name="medicine" id="medicine" value="<?=
set_value('medicine') ?>" class="txt_fields"
/>
            <?php echo
form_error('medicine', '<div
class="error">', '</div>'); ?>
        </td>
    </tr>
    <tr>
        <td>*Dosage: </td>
        <td>
            <input type="text"
name="dosage" id="dosage" value="<?=
set_value('dosage') ?>" class="txt_fields"
/>
            <?php echo
form_error('dosage', '<div class="error">',
'</div>'); ?>
        </td>
    </tr>
    <tr>
        <td>Medication Details</td>
        <table
width="611" border="0">
            <tr>
                <td style="display: none">
                    <br> </br>
                    <input
type="hidden" name="hidden_medication"
id="hidden_medication" value="<?=
set_value('hidden_medication', 0) ?>" />
                </td>
            </tr>
        </table>
    </tr>
    </table>
</div>
</fieldset>

```

```

        <?php echo
form_error('medication_date', '<div
class="error">', '</div>'); ?>

    </td>

</tr>

</table>

        </fieldset>
</div>

    <div id="vaccination"
style="display: none">
        <br> </br>
        <input
type="hidden" name="hidden_vaccination"
id="hidden_vaccination" value="<?=
set_value('hidden_vaccination', 0) ?>"/>
        <fieldset>

            <legend>Vaccination Details</legend>
            <table
width="611" border="0">

                <tr>

                    <td width="129">*Vaccine Type: </td>

                    <td>

                        <input type="text"
name="vaccine_type" id="vaccine_type"
value="<?= set_value('vaccine_type') ?>"
class="txt_fields" />

                        <?php echo
form_error('vaccine_type', '<div
class="error">', '</div>'); ?>

                    </td>

                </tr>

                <tr>

                    <td>*Vaccine Name: </td>

                    <td>

                        <input type="text"
name="vaccine_name" id="vaccine_name"
value="<?= set_value('vaccine_name') ?>"
class="txt_fields" />

                        <?php echo
form_error('vaccine_name', '<div
class="error">', '</div>'); ?>

                    </td>

                </tr>

                <tr>

                    <td>*Expiration Date: </td>

                    <td>

                        <input type="date"

```

```

name="expiration_date" id="expiration_date"
value="<?= set_value('expiration_date') ?>"
class="txt_fields" />

        <?php echo
form_error('expiration_date', '<div
class="error">', '</div>'); ?>

    </td>

</tr>

        <tr>

            <td>*Next Vaccination Date: </td>

            <td>

                <input type="date"
name="next_vaccine_date"
id="next_vaccine_date" value="<?=
set_value('next_vaccine_date') ?>"
class="txt_fields" />

                <?php echo
form_error('next_vaccine_date', '<div
class="error">', '</div>'); ?>

            </td>

        </tr>

    </table>

        </fieldset>
</div>

    <div id="laboratory"
style="display: none">
        <br> </br>
        <input
type="hidden" name="hidden_laboratory"
id="hidden_laboratory" value="<?=
set_value('hidden_laboratory', 0) ?>"/>
        <fieldset>

            <legend>Laboratory Details</legend>
            <table
width="611" border="0">

                <tr>

                    <td width="129">*Laboratory Type:

                    <td>

                        <input type="text"
name="laboratory_type" id="laboratory_type"
value="<?= set_value('laboratory_type') ?>"
class="txt_fields" />

                        <?php echo
form_error('laboratory_type', '<div
class="error">', '</div>'); ?>

                    </td>

                </tr>

                <tr>

                    <td>

                    <td>

                        <input type="date"

```

```

<td>*Findings: </td>

<td>
    <input type="text"
name="lab_findings" id="lab_findings"
value="<?= set_value('lab_findings') ?>"
class="txt_fields" />

    <?php echo
form_error('lab_findings', '<div
class="error">', '</div>'); ?>

</td>
</tr>

<tr>
<td>*Laboratory Date: </td>

<td>
    <input type="date"
name="lab_date" id="lab_date" value="<?=
set_value('lab_date') ?>" class="txt_fields"
/>

    <?php echo
form_error('lab_date', '<div
class="error">', '</div>'); ?>

</td>
</tr>
</table>
</fieldset>
</div>
<table>
<tr>
<td
width="120px">&nbsp;&nbsp;&nbsp;</td>
<td><input type="button"
name="button" id="med"
onclick="toggle_visibility('medication');"
value="Medication" class="search_button"
style="height:40px;" />&nbsp;&nbsp;&nbsp;</td>

<td><input type="button"
name="button" id="med"
onclick="toggle_visibility('vaccination');"
value="Vaccination" class="search_button"
style="height:40px;" />&nbsp;&nbsp;&nbsp;</td>

<td><input type="button"
name="button" id="button"
onclick="toggle_visibility('laboratory');"
value="Laboratory" class="search_button"
style="height:40px;" />&nbsp;&nbsp;&nbsp;</td>

<td><input type="submit"
name="button" id="button" value="Submit"
class="search_button" style="height:40px;"
/>&nbsp;&nbsp;&nbsp;</td>
</tr>
</table>
</div>

```

```

<script language="Javascript">
$(document).ready(function() {
    if($('#r1').is(':checked')) {
        document.getElementById("B").style.display="none";

        document.getElementById("A").style.display="block";
    } else {
        document.getElementById("A").style.display="none";

        document.getElementById("B").style.display="block";
    }
    if
    ($('#hidden_medication').val() == 1) {
        document.getElementById("medication")
        .style.display="block";
    } else {
        document.getElementById("medication")
        .style.display="none";
    }
    if
    ($('#hidden_laboratory').val() == 1) {
        document.getElementById("laboratory")
        .style.display="block";
    } else {
        document.getElementById("laboratory")
        .style.display="none";
    }
    if
    ($('#hidden_vaccination').val() == 1) {
        document.getElementById("vaccination")
        .style.display="block";
    } else {
        document.getElementById("vaccination")
        .style.display="none";
    }
});

function toggle_visibility(id) {
    var e = document.getElementById(id);
    if(e.style.display == 'block') {
        e.style.display = 'none';
        $("#hidden_" + id).val(0);
    } else {
        e.style.display = 'block';
        $("#hidden_" + id).val(1);
    }
}

function hideA(x)
{
    if (x.checked)
    {
        document.getElementById("A").style.display="none";
    }
}

```

```

        document.getElementById("B").style.display="block";

        $("#hidden_info").val('');
    }

    function hideB(x)
    {
        if (x.checked)
        {
            document.getElementById("B").style.display="none";

            document.getElementById("A").style.display="block";

            $("#hidden_info").val('');
        }
    }

    $(function(){

        $(".searchpatient").keyup(function()
        {
            var searchid =
$(this).val();
            var dataString =
'searchpatient='+ searchid;
            if(searchid!='')
            {
                $.ajax({
                    type:
"POST",
                    url:
<?php echo "\"" .
base_url("index.php/main/search_patients") .
"\\"; ?>,
                    data:
dataString,
                    cache:
false,
                    success: function(html)
                    {
                        $("#result_searchpatient").html(html)
                        .show();
                    }
                });
            } return false;
        });

        $(document).on("click",
"#result_searchpatient", function(e){
            var $clicked =
$(e.target);

            var $name =
$clicked.find('.name').html();
            var decoded =
$("<div/>").html($name).text();

            $("#searchpatient").val(decoded);
        });

        $(document).on("click",
".show", function(e) {
            $("#hidden_info").val(this.id);
        });

        $(document).on("click",
function(e) {
            var $clicked =
$(e.target);
            if (!
$clicked.hasClass("searchpatient")){
                jQuery("#result_searchpatient").fadeOut();
            }

            $("#searchpatient").click(function(){
                jQuery("#result_searchpatient").fadeIn();
            });
        });
    });
</script>

<style type="text/css">
    .content_searchpatient{
        width:611px;
        margin:0 auto;
    }
    #searchpatient
    {
        width:500px;
        border:solid 1px #000;
        padding:10px;
        font-size:14px;
    }
    #result_searchpatient
    {
        position:absolute;
        width:500px;
        max-height:300px;
        padding:10px;
        display:none;
        margin-top:-1px;
        border-top:0px;
        overflow-y:auto;
        border:1px #CCC solid;
        background-color: white;
    }
    .show
    {
        padding:10px;
        border-bottom:1px #999 dashed;
        font-size:15px;
        height:50px;
    }
    .show: hover
    {
        background:#4c66a4;
        color:#FFF;
        cursor:pointer;
    }
</style>

```

Listing 13: content/add-pet.php

```

        <div id="right_content"
class="right_design border"
style="width:700px;">
        <h3>Pet
Information</h3>
        <?php
            echo
form_open('main/add_newpet');
        >
        <table width="701"
border="0" style="font-family:Arial,
Helvetica, sans-serif; font-size:16px;">
        <tr>
            <td>*Name:</td>
            <td>
                <input type="text" name="pet_name"
id="pet_name" value="<?=  

set_value('pet_name') ?>" class="txt_fields"
/>
                <?php echo form_error('pet_name',
'<div class="error">', '</div>'); ?>
            </td>
        </tr>
        <tr>
            <td>*Specie:</td>
            <td>
                <input type="text" name="specie"
id="specie" value="<?=  

set_value('specie') ?>" class="txt_fields" />
                <?php echo form_error('specie', '<div
class="error">', '</div>'); ?>
            </td>
        </tr>
        <tr>
            <td>*Breed:</td>
            <td>
                <input type="text" name="breed"
id="breed" value="<?=  

set_value('breed') ?>"
class="txt_fields" />
                <?php echo form_error('breed', '<div
class="error">', '</div>'); ?>
            </td>
        </tr>
        <tr>
            <td>*Birth date:</td>
            <td>
                <input type="date" name="birthday"
id="birthday" value="<?=  

set_value('birthday') ?>" class="txt_fields"
/>
                <?php echo form_error('birthday',
'<div class="error">', '</div>'); ?>
            </td>
        </tr>
        <tr>
            <td>*Gender:</td>
            <td>

```

```

        <select name="gender" id="gender"
class="txt_fields" style="height:30px;
width:408px;" >
            <option value ="" <?php echo
set_select('gender', '', TRUE); ?></option>
            <option value ="Male" <?php echo
set_select('gender', 'Male');
?>>Male</option>
            <option value ="Female" <?php echo
set_select('gender', 'Female'); ?>
?>Female</option>
        </select>
        <?php echo form_error('gender', '<div
class="error">', '</div>'); ?>
        </tr>
        <tr>
            <td
width="128">Registration No.:</td>
            <td
width="563">
                <input type="text" name="registry"
id="registry" value="<?=  

set_value('registry', "") ?>"
class="txt_fields" placeholder='Registration
number (if applicable)' />
                <?php echo form_error('registry',
'<div class="error">', '</div>'); ?>
            </td>
        </tr>
        <tr>
            <td>&nbsp;</td>
            <td><input type="submit"
name="save_button" id="save_button"
class="search_button" value="Save" /></td>
        </tr>
    </table>
</form>
</div>

```

Listing 14: content/add-veterinarian.php

```

        <div id="right_content"
class="right_design border"
style="width:700px;">
        <?php
            echo
form_open('main/addVet');
        >
        <div id="clear"><br
/></div>
        <fieldset>
        <legend>Veterinarian
Info</legend>
            <input id="r1"
type="radio" onchange="hideB(this)"
name="aorb" value="1" <?php echo
set_radio('aorb', '1', TRUE); ?> > Search
from existing veterinarian |

```

```

                                <input id="r2"
type="radio" onchange="hideA(this)"
name="aorb" value="2" <?php echo
set_radio('aorb', '2'); ?> > Add new
veterinarian account
                                <input
type="hidden" name="hidden_username"
id="hidden_username" value="<?=  

set_value('hidden_username') ?>"/>
                                <div id="A">
                                <div
class="content_searchvet">
                                <input type="text" name="searchvet"
class="searchvet" id="searchvet"
placeholder="Enter (name or email) to search
account." value="<?=  

set_value('searchvet')
?>" autocomplete="off" />
                                <?php echo
form_error('hidden_username', '<div
class="error">', '</div>'); ?>
                                <div id="result_searchvet"></div>
                                </div>
                                <div id="B"
style="display: none">
                                <table
width="611" border="0">
                                <tr>
                                <td>Register As:</td>
                                <td>
                                <select name="role"
class="txt_fields txt_width"
style="height:30px; width:408px;" id="role"
disabled>
                                <option
value="clinic">Veterinarian</option>
                                </select>
                                </td>
                                </tr>
                                <tr>
                                <td width="129">*Username:</td>
                                <td>
                                <input type="text"
name="username" id="username" value="<?=  

set_value('username') ?>" class="txt_fields"
/>
                                <?php echo
form_error('username', '<div
class="error">', '</div>'); ?>
                                </td>
                                </tr>
                                <tr>

```

```

                                <td>*Password:</td>
                                <td><input type="password"
name="password" id="password" value="<?=  

set_value('password') ?>" class="txt_fields"
/>
                                <?php echo form_error('password',
'<div class="error">', '</div>'); ?>
                                </td>
                                </tr>
                                <tr>
                                <td>*Confirm Password:</td>
                                <td><input type="password"
name="confirm_password"
id="confirm_password" value="<?=  

set_value('confirm_password') ?>"
class="txt_fields" />
                                <?php echo
form_error('confirm_password', '<div
class="error">', '</div>'); ?>
                                </td>
                                </tr>
                                <tr>
                                <td>*Email Address:</td>
                                <td><input type="text" name="email"
id="email" value="<?=  

set_value('email') ?>"
class="txt_fields" />
                                <?php echo form_error('email', '<div
class="error">', '</div>'); ?>
                                </td>
                                </tr>
                                <tr>
                                <td>*Last Name:</td>
                                <td><input type="text"
name="last_name" id="last_name" value="<?=  

set_value('last_name') ?>"
class="txt_fields" />
                                <?php echo form_error('last_name',
'<div class="error">', '</div>'); ?>
                                </td>
                                </tr>
                                <tr>
                                <td>*First Name:</td>
                                <td><input type="text"
name="first_name" id="first_name" value="<?=  


```



```

set_value('first_name') ?>"
class="txt_fields" />

    <?php echo form_error('first_name',
'<div class="error">', '</div>'); ?>

    </td>
    </tr>
    <tr>
        <td>*Middle Name:</td>
        <td>
            <input type="checkbox"
name="specialize[]" value="<?=$specialize['spec_id'] ?>" <?php echo
set_checkbox('specialize[]',
$specialize['spec_id']); ?> /> <?=$specialize['specialization'] ?>
            </td>
        </tr>
        <td><input type="text"
name="middle_name" id="middle_name"
value="<?=$set_value('middle_name') ?>"
class="txt_fields" />
        <td>
            <?php echo form_error('middle_name',
'<div class="error">', '</div>'); ?>
        </td>
    </tr>
    <tr>
        <td>*Contact No:</td>
        <td><input type="text"
name="contact_no" id="contact_no" value="<?=$set_value('contact_no') ?>"
class="txt_fields" />
        <td>
            <?php echo form_error('contact_no',
'<div class="error">', '</div>'); ?>
        </td>
    </tr>
    <tr>
        <td>*License No:</td>
        <td><input type="text"
name="license_no" id="license_no" value="<?=$set_value('license_no') ?>"
class="txt_fields" />
        <td>
            <?php echo form_error('license_no',
'<div class="error">', '</div>'); ?>
        </td>
    </tr>
</table>
</div>
</fieldset>
</fieldset>
<legend></legend>
<div id="C"
style="display: none">
<table
width="611" border="0">
    <tr>
        <td>*Schedule:</td>
        <td><input type="text"
name="schedule" id="schedule" value="<?=$set_value('schedule') ?>" class="txt_fields"
/>
        <td>
            <?php echo form_error('schedule',
'<div class="error">', '</div>'); ?>
        </td>
    </tr>
    <tr>
        <td class="txt_fields">&nbsp;</td>
        <td><input type="submit"
name="button" id="button" value="Add Vet"
class="search_button" style="height:40px;"
/>&nbsp;</td>
    </tr>
</table>
</div>
</fieldset>
</form>
</div>
<script language="Javascript">
$(document).ready(function() {
    if($('#r1').is(':checked')) {
        document.getElementById("C").style.display="none";
    }
});

```

```

        document.getElementById("B").style.display="none";

        document.getElementById("A").style.display="block";
    } else {

        document.getElementById("A").style.display="none";

        document.getElementById("B").style.display="block";

        document.getElementById("C").style.display="block";
    }
});

function hideA(x)
{
    if (x.checked)
    {
        document.getElementById("A").style.display="none";

        document.getElementById("B").style.display="block";

        document.getElementById("C").style.display="block";

        $("#hidden_username").val("");
    }
}

function hideB(x)
{
    if (x.checked)
    {
        document.getElementById("C").style.display="none";

        document.getElementById("B").style.display="none";

        document.getElementById("A").style.display="block";

        $("#hidden_username").val("");
    }
}

$(function(){

    $(".searchvet").keyup(function()

    {
        var searchid =
$(this).val();
        var dataString =
'searchvet='+ searchid;
        if(searchid!='')
        {
            $.ajax({
                type:
"POST",
                url:
<?php echo "\".

```

```

base_url("index.php/main/search_vets") .
"\\"; ?>,
                                data:
dataString,
                                cache:
false,
                                success: function(html)
                                {
                                    $("#result_searchvet").html(html).show();
                                }
                                });
                                } return false;
                            });
                        });
                    $(document).on("click",
"#result_searchvet", function(e) {
                        var $clicked =
$(e.target);
                        var $name =
$clicked.find('.name').html();
                        var decoded =
$("<div/>").html($name).text();

                        $('#searchvet').val(decoded);
                    });

                    $(document).on("click",
".show", function(e) {

                        $('#hidden_username').val(this.id);
                    });

                    $(document).on("click",
function(e) {
                        var $clicked =
$(e.target);
                        if (!
$clicked.hasClass("searchvet")){

                            jQuery("#result_searchvet").fadeOut();
                        }
                    });

                    $('#searchvet').click(function() {

                        jQuery("#result_searchvet").fadeIn();
                    });
                });
            </script>

            <style type="text/css">
                .content_searchvet{
                    width:611px;
                    margin:0 auto;
                }
                #searchvet
                {
                    width:500px;
                    border:solid 1px #000;
                    padding:10px;
                    font-size:14px;
                }
                #result_searchvet
                {
                    position:absolute;
                    width:500px;

```

```

        max-height:300px;
padding:10px;
display:none;
margin-top:-1px;
border-top:0px;
overflow-y:auto;
border:1px #CCC solid;
background-color: white;
}
.show
{
padding:10px;
border-bottom:1px #999 dashed;
font-size:15px;
height:50px;
}
.show:hover
{
background:#4c66a4;
color:#FFF;
cursor:pointer;
}
}
</style>

```

Listing 15: content/edit-clinic.php

```

<script>
    $(document).ready(function() {
        // Basic usage
        $(".placepicker").each(function() {
            var placepicker =
$(this).placepicker({
                latitude:
document.getElementById('lat').value,
                longitude :
document.getElementById('lng').value,
                placeChanged:
function(place) {
                    var
location = this.getLocation();
                    document.getElementById('lat').value
= location['latitude'] ;
                    document.getElementById('lng').value
= location['longitude'] ;
                    console.log("place changed: ",
place.formatted_address,
this.getLocation());
                }
            }).data('placepicker');
        }); // END document.ready
</script>
<style>
    .placepicker-map {
width: 100%;
height: 300px;

```

```

    }
</style>
        <div id="right_content"
class="border" style="width:700px;">
            <h3>Clinic
Information</h3>
            <div class="pet_info">
                <?php
echo
form_open('main/editting_clinic'); ?>
                <table
width="611" border="0">
                    <tr>
                        <td width="129">Clinic Name:</td>
                        <td>
                            <input type="text" name="clinic_name"
id="clinic_name" value="<?=
set_value('clinic_name',
$clinic['clinic_name']) ?>"
class="txt_fields" />
                            <?php echo form_error('clinic_name',
'<div class="error">', '</div>'); ?>
                        </td>
                    </tr>
                    <tr>
                        <td>Location:</td>
                        <td><input type="text"
name="clinic_loc" id="clinic_loc" value="<?=
set_value('clinic_loc', $clinic['address'])
?>" class="txt_fields" />
                            <?php echo form_error('clinic_loc',
'<div class="error">', '</div>'); ?>
                        </td>
                    </tr>
                    <tr>
                        <td>Contact No:</td>
                        <td><input type="text"
name="contact_no" id="contact_no" value="<?=
set_value('contact_no',
$clinic['contact_no']) ?>"
class="txt_fields" />
                            <?php echo form_error('contact_no',
'<div class="error">', '</div>'); ?>
                        </td>
                    </tr>
                    <tr>
                        <td>Email Address:</td>
                        <td><input type="text"
name="clinic_email" id="clinic_email"
value="<?= set_value('clinic_email',
$clinic['email']) ?>" class="txt_fields" />
                            <?php echo form_error('clinic_email',
'<div class="error">', '</div>'); ?>
                        </td>
                    </tr>
                </table>
            </div>

```

```

</td>
</tr>
<tr>
<td>Operating Hours:</td>
<td><input type="text"
name="clinic_hours" id="clinic_hours"
value="<?= set_value('clinic_hours',
$clinic['schedule']) ?>" class="txt_fields"
/>
<?php echo form_error('clinic_hours',
'<div class="error">', '</div>'); ?>
</td>
</tr>
</table>
<br /><br />
<fieldset>
<legend>Location details</legend>
<div
class="col-md-6">
* <input name="placepicker"
id="placepicker" class="placepicker" data-
map-container-id="collapseOne" value =
"<?php echo set_value('placepicker'); ?>" />
<div id="collapseOne"
class="collapse">
<div class="placepicker-map"></div>
</div>
</div>
Lat:
<input name="lat" type="text" id="lat"
value="<?= set_value('lat', $clinic['x'])
?>" style="height:25px" readonly />
Lng:
<input name="lng" type="text" id="lng"
value="<?= set_value('lng', $clinic['y'])
?>" style="height:25px" readonly />
<?php
echo form_error('placepicker', '<div
class="error">', '</div>'); ?>
</fieldset>
<br />
<input
type="submit" name="edit" id="edit"
value="Edit Profile" class="search_button"
/>
</form>
</div>
</div>

```

Listing 16: content/edit-pet.php

```

<div id="right_content"
class="right_design border"
style="width:700px;">
<h3>Pet
Information</h3>
<?php

```

```

echo
form_open('main/updatepet/');
?>
<input type="hidden"
name="pet_id" id="pet_id" value="<?=
set_value('pet_id', $pet['pet_id']) ?>" />
<table width="701"
border="0" style="font-family:Arial,
Helvetica, sans-serif; font-size:16px;">
<tr>
<td>*Name:</td>
<td>
<input type="text" name="pet_name"
id="pet_name" value="<?=
set_value('pet_name', $pet['pet_name']) ?>"
class="txt_fields" />
<?php echo form_error('pet_name',
'<div class="error">', '</div>'); ?>
</td>
</tr>
<tr>
<td>*Specie:</td>
<td>
<input type="text" name="specie"
id="specie" value="<?= set_value('specie',
$pet['specie']) ?>" class="txt_fields" />
<?php echo form_error('specie', '<div
class="error">', '</div>'); ?>
</td>
</tr>
<tr>
<td>*Breed:</td>
<td>
<input type="text" name="breed"
id="breed" value="<?= set_value('breed',
$pet['breed']) ?>" class="txt_fields" />
<?php echo form_error('breed', '<div
class="error">', '</div>'); ?>
</td>
</tr>
<tr>
<td>*Birth date:</td>
<td>
<input type="date" name="birthday"
id="birthday" value="<?=
set_value('birthday', $pet['birthdate']) ?>"
class="txt_fields" />
<?php echo form_error('birthday',
'<div class="error">', '</div>'); ?>
</td>
</tr>
<tr>
<td>*Gender:</td>
<td>
<select name="gender" id="gender"
class="txt_fields" style="height:30px;
width:408px;">

```

```

        <option value ="" <?php echo
set_select('gender', ''); ?></option>

        <option value ="Male" <?php
if($pet['gender'] == "Male") echo
"selected"; ?> <?php echo
set_select('gender', 'Male');
?>>Male</option>

        <option value ="Female" <?php
if($pet['gender'] == "Female") echo
"selected"; ?> <?php echo
set_select('gender', 'Female'); ?>
>Female</option>

    </select>

    <?php echo form_error('gender', '<div
class="error">', '</div>'); ?>
    </td>
    </tr>
    <tr>
        <td
width="128">Registration No.:</td>
        <td
width="563">
            <input type="text" name="registry"
id="registry" value="<?=
set_value('registry',
$pet['registration no']) ?>"
class="txt_fields" placeholder='Registration
number (if applicable)' />
            </td>
    </tr>
    <tr>
        <td>&nbsp;</td>
        <td><input type="submit"
name="save_button" id="save_button"
class="search_button" value="Save" /></td>
    </tr>
</table>
</form>
</div>

```

Listing 17: content/edit-profile.php

```

        <div id="right_content"
class="right_design border"
style="width:700px;">
            <h3>User
Information</h3>
            <div
class="error_div">
                <?php
                    echo
form_open('main/editting_profile'); ?>
                </div>
                <table width="701"
border="0" style="font-family:Arial,
Helvetica, sans-serif; font-size:16px;">
                    <tr>
                        <td
width="128">Username:</td>
                        <td
width="563">
                            <input
type="text" name="username" id="username"
value="<?= $profile['username'] ?>"
class="txt_fields" readonly />
                            <?php
                                echo form_error('username', '<div
class="error">', '</div>'); ?>
                        </td>
                    </tr>
                    <tr>
                        <td>*Password:</td>
                        <td><input type="password"
name="password" id="password" value="<?=
set_value('password', $profile['password'])
?>" class="txt_fields"/>
                            <?php
                                echo form_error('password', '<div
class="error">', '</div>'); ?>
                        </td>
                    </tr>
                    <tr>
                        <td>*Confirm Password:</td>
                        <td><input type="password"
name="confirm_password"
id="confirm_password" value="<?=
set_value('confirm_password',
$profile['password']) ?>"
class="txt_fields"/>
                            <?php
                                echo form_error('confirm_password', '<div
class="error">', '</div>'); ?>
                        </td>
                    </tr>
                    <tr>
                        <td>Email:</td>
                        <td><input type="text" name="email"
id="email" value="<?= $profile['email'] ?>"
class="txt_fields" readonly />
                            <?php
                                echo form_error('email', '<div
class="error">', '</div>'); ?>
                        </td>
                    </tr>
                    <tr>
                        <td>*Last Name:</td>
                        <td><input type="text"
name="last_name" id="last_name" value="<?=
set_value('last_name',
$profile['last_name']) ?>"
class="txt_fields" />
                            <?php
                                echo form_error('last_name', '<div
class="error">', '</div>'); ?>
                        </td>
                    </tr>
                    <tr>
                        <td>*First Name:</td>
                        <td><input type="text"
name="first_name" id="first_name" value="<?=

```



```

>get_clinic_info($clinic_id)['clinic_name']
?>" :</h3>
        <table width="701"
border="0" style="font-family:Arial,
Helvetica, sans-serif; font-size:16px;">
        <tr>
                <td>Current Schedule:</td>
                <td>
                        <label class="sched">
                                <?php echo $schedule ?>
                        </label>
                </td>
        </tr>
        <tr>
                <td>New
                <td>
                        <input type="text" name="new_sched"
id="new_sched" value="<?=
set_value('new_sched') ?>"
class="txt_fields" />
                        <?php echo form_error('new_sched',
'<div class="error">', '</div>'); ?>
                </td>
        </tr>
        <tr>
                <td>&nbsp;</td>
                <td><input type="submit"
name="save_button" id="save_button"
class="search_button" value="Save" /></td>
        </tr>
</table>
</div>
<div>
<div id="right_content"
class="inner user_design_r"
style="width:650px;" >
        <h3
style="background:#6CF; font-
size:14px;color:#333; border:#69C 1px solid;
padding:8px;">Clinic Information and
Location</h3>
        <div class="pet_info">
                <li><strong>Clinic Name:

```

Listing 19: content/my-clinic.php

```

</strong><?php echo $clinic['clinic_name'];
?></li>
        <li><strong>Location: </strong> <?php
echo $clinic['address']; ?></li>
        <li><strong>Telephone: </strong><?php
echo $clinic['contact_no']; ?></li>
        <li><strong>Email: </strong><?php
echo $clinic['email']; ?></li>
        <li><strong>Operating Hours:
</strong><?php echo $clinic['schedule'];
?></li>
</div>
<hr>
<div
id="clear" style="padding-
bottom:10px;"></div>
        <a href="<?php
if ($_SESSION['user_role'] == 2) {
                echo
base_url("index.php/main");
        }
        else {
                echo "javascript:window.history.go(-1);";
        } ?>"
class="search_button" style="padding:10px;
text-decoration:none;
background:#F93;">Back</a>
        <?php if
($this->session->userdata('user_role') == 2)
: ?>
        <a
href="<?php echo
base_url("index.php/main/edit_clinic"); ?>"
class="search_button" style="padding:10px;
text-decoration:none; background:#F93;">Edit
Clinic Information</a>
        <?php endif;
?>
</div>
</div>

```

Listing 20: content/my-clinics.php

```

        <div id="right_content"
class="right_design border"
style="width:700px;">
                <div id="clear"><br
/></div>
                <table width="710"
border="0" style="font-family: Arial,
Helvetica, sans-serif; font-size:14px;">
                        <tr >
                                <th
width="200" style="border: #C0C0C0 1px
solid;"><strong>Clinic Name</strong></th>
                                <th
width="400" style="border: #C0C0C0 1px
solid;"><strong>Address</strong></th>
                                <th
width="200" style="border: #C0C0C0 1px
solid;"> <strong>My Schedule</strong></th>

```

```

        <th
width="60" style="border: #C0C0C0 1px solid;
text-align: center;"><strong></strong></th>
        </tr>
        <?php foreach
($clinicsList as $clinic) :
        <tr>
                <td
style="border: #C0C0C0 1px solid;"> <?php
                $clinic = $this->model_home-
>get_clinic_info($clinic['clinic_id']);
                echo $clinic['clinic_name']; ?> </td>
                <td
style="border: #C0C0C0 1px solid;"> <?php
                $clinic['address'] ?> </td>
                <td
style="border: #C0C0C0 1px solid; text-
align: center;"> <?php $this->model_home-
>get_schedule_str($clinic['clinic_id'],
$_SESSION['vet_id'], 1) ?> </td>
                <td
style="border: #C0C0C0 1px solid;">
                <a href = <?php echo "\"" .
base_url("index.php/main/view_clinic") . "/"
. $clinic['clinic_id'] . "\""; ?> >
                 "
title="View Clinic"width="20" height="20" />
                </a>
                <a href = <?php echo "\"" .
base_url("index.php/main/edit_clinic_sched")
. "/" . $clinic['clinic_id'] . "\""; ?> >
                
" title="Edit Schedule" width="20"
height="20" />
                </a>
        </td>
        </tr>
        <?php endforeach; ?>
</table>
</div>

```

Listing 21: content/my-patients.php

```

        <div id="right_content"
class="right_design border"
style="width:700px;">
                <a href=<?php echo
        "\"" .
base_url("index.php/main/add_patient") .
        "\"" ; ?> class="search_button"
                style="font-
family:Arial, Helvetica, sans-
serif;padding:5px; text-
decoration:none;margin-bottom:10px;">Add
Patient Record</a>

```

```

/></div>
        <div id="clear"><br
        <fieldset>
        <table>
                <tr>
                        <td
class="clinic" style="text-align:right">
                                <label><em>*</em>Clinic: </label>
                                </td>
                                <td
class="clinic">
                                        <select name="clinic_option"
id="clinic_option" class="clinic_option"
style="height:30px; width:495px">
                                                <!--for loop clinic -->
                                                <?php foreach ($clinics as $clinic) :
                                                        ?>
                                                                <?php $clinic_info = $this-
                                                                >model_home-
                                                                >get_clinic_info($clinic['clinic_id']); ?>
                                                                <option value="<?php echo
                                                                $clinic_info['clinic_id']?>"><?php echo
                                                                $clinic_info['clinic_name'] ?></option>
                                                                <?php endforeach; ?>
                                                                </select>
                                                                <?php echo
                                                                form_error('clinic_option', '<div
                                                                class="error">', '</div>'); ?>
                                                                </td>
                                                                </tr>
                                                                </table>
                                                                </fieldset>
                                                                <div class="patient">
                                                                </div>
                                                                </div>
<script>
        $( document ).ready(function() {
                $(''.clinic_option').trigger('change')
;
                });
                $(''.clinic_option').on('change',
function() {
                var clinic_id =
                $(''.clinic_option').val();
                $.ajax({
                        type: "POST",
                        url: <?php echo "\"" .
base_url("index.php/main/get_patient_list")
. "\""; ?>,
                        data: { clinic_id:
clinic_id },
                        cache: false,
                        success:
function(html)
                        {
                                $("div.patient")
                                .find('div')

```



```

.remove()
.end()

.append(html)
    }
    });
</script>

```

Listing 22: content/my-pet.php

```

<div id="right_content"
class="inner user_design_r"
style="width:700px;" >
    <h3
style="background:#6CF; font-
size:14px;color:#333; border:#69C 1px solid;
padding:8px;">Pet Information</h3>
        <div class="pet_info">

            <li><strong>Pet Name:</strong><?php
echo $pet['pet_name']; ?></li>

            <li><strong>Registration
No.:</strong> <?php echo
$pet['registration_no'] != '' ?
$pet['registration_no'] : '< Not registered
>'; ?></li>

            <li><strong>Specie: </strong><?php
echo $pet['specie']; ?></li>

            <li><strong>Breed: </strong><?php
echo $pet['breed']; ?></li>

            <li><strong>Birth date:
</strong><?php echo $pet['birthdate'];
?></li>

            <li><strong>Age : </strong><?php
echo $this->model_home-
>findage($pet['birthdate']); ?></li>

            <li><strong>Gender: </strong><?php
echo $pet['gender']; ?></li>

            <hr>
            <?php
if($_SESSION['user_role'] == 4) : ?>
                <br> </br>
                <a href=<?php
echo "\"" .
base_url("index.php/main/edit_pet") . "/" .
$pet_id . "\""; ?> " class="search_button"
style="padding:10px; text-decoration:none;
height:40px;">Edit Pet Info</a>
                <?php endif;
?>
                <br> </br>
            </div>
            <?php if
(array_filter($consultationList)) : ?>
                <div id="clear"><br>
                <h3
style="background:#6CF; font-

```

```

size:14px;color:#333; border:#69C 1px solid;
padding:8px;">Consultation History</h3>

```

```

        <table width="710"
border="0" style="font-family: Arial,
Helvetica, sans-serif; font-size:14px;">
            <tr>
                <th
width="189" style="border: #C0C0C0 1px
solid;"><strong>Date</strong></th>
                <th
width="189" style="border: #C0C0C0 1px
solid;"><strong>Diagnosis</strong></th>
                <th
width="189" style="border: #C0C0C0 1px
solid;"><strong>Remarks</strong></th>
                <th
width="80" style="border: #C0C0C0 1px
solid;"><strong></strong></th>
            </tr>
            <?php foreach
($consultationList as $consultation) :
                ?>
                    <tr>
                        <td
style="border: #C0C0C0 1px solid;"> <?=
$consultation['consultation_date'] ?> </td>
                        <td
style="border: #C0C0C0 1px solid;"> <?=
$consultation['diagnosis'] ?> </td>
                        <td
style="border: #C0C0C0 1px solid;"> <?=
$consultation['remarks'] == "" ? "< No
remarks >" : $consultation['remarks'] ?>
</td>
                        <td
style="border: #C0C0C0 1px solid;">
                            <a href = <?php echo "\"" .
base_url("index.php/main/view_consultation")
. "/" . $consultation['consultation_id'] .
"\\""; ?> >
                             "
title="View Consultation Info"width="20"
height="20" />
                        </a>
                    </td>
                </tr>
            <?php endforeach; ?>
        </table>
    <?php endif; ?>
    <div id="clear"
style="padding-bottom:10px;"></div>
    <a href="<?php
echo base_url("index.php/main/"); ?>"
class="search_button" style="padding:10px;
text-decoration:none;
background:#F93;">Back</a>
</div>

```

Listing 23: content/my-pets.php

```

        <div id="right_content"
class="right_design border"
style="width:700px;">
    <?php foreach ($petsList as
$pet) : ?>

        <div id="admin_box"
style = "display:inline-block; width:180px;
height:100px">
            <table>
                <tr>

                    <td width = "40"><img src=<?php echo
"\ " . base_url("resources/images/paw.png")
. "\ " ; ?> width="35" height="40" /></td>

                    <td style="text-align: center;
vertical-align: middle;"> <div
style="display: table-cell; vertical-align:
middle;">

                        <h3><a href=<?php echo "\ " .
base_url("index.php/main/view_pet") . "/" .
$pet['pet_id'] . "\ " ; ?>><? =
$pet['pet_name'] ?></a></h3>

                        <font size=2><? = $pet['specie'] . ' :
' . $pet['breed'] ?></font>

                    </div>

                </td>

            </tr>
        </table>
        </div>
    <?php endforeach; ?>

    <div id="admin_box"
style="background-color : transparent;
border:0;"><h3><a href=<?php echo "\ " .
base_url("index.php/main/add_pet") . "\ " ;
?>>

        <img src=<?php
echo "\ " .
base_url("resources/images/create.png") .
"\ " ; ?> width="35" height="40" title="Add
pet" />

        </a></h3></div>

</div>

```

Listing 24: content/my-profile.php

```

    <div id="right_content"
class="inner user_design_r"
style="width:650px;" >
        <h3
style="background:#6CF; font-
size:14px;color:#333; border:#69C 1px solid;
padding:8px;">User Information</h3>
        <div class="pet_info">

            <li><strong>Username:</strong><?php
echo $profile['username']; ?></li>

```

```

            <li><strong>Email Address:</strong>
<?php echo $profile['email']; ?></li>

            <li><strong>Last Name: </strong><?php
echo $profile['last_name']; ?></li>

            <li><strong>First Name:
</strong><?php echo $profile['first_name'];
?></li>

            <li><strong>Middle Name:
</strong><?php echo $profile['middle_name'];
?></li>

            <li><strong>Contact No.:
</strong><?php echo $profile['contact_no'];
?></li>

            <?php
if($_SESSION['user_role'] == 3) : ?>

                <li><strong>License No:
</strong><?php echo $this->model_home-
>get_vetlicense($profile['username']);
?></li>

                <li><strong>Specialization:
</strong><?php echo $this->model_home-
>get_specializations_str($_SESSION['vet_id']
); ?></li>

            <?php endif; ?>
            <hr>
            <div
id="clear" style="padding-
bottom:10px;"></div>
            <a href="<?php
echo base_url("index.php/main"); ?>"
class="search_button" style="padding:10px;
text-decoration:none;
background:#F93;">Back</a>

            <a href="<?php
echo
base_url("index.php/main/edit_profile"); ?>"
class="search_button" style="padding:10px;
text-decoration:none; background:#F93;">Edit
Profile</a>

        </div>
    </div>

```

Listing 25: content/my-vet.php

```

    <div id="right_content"
class="inner user_design_r"
style="width:650px;" >
        <h3
style="background:#6CF; font-
size:14px;color:#333; border:#69C 1px solid;
padding:8px;">My Veterinarian
Information</h3>
        <div class="pet_info">

            <li><strong>Username: </strong><?php
echo $profile['username']; ?></li>

            <li><strong>Email Address: </strong>
<?php echo $profile['email']; ?></li>

```

```

        <li><strong>Last Name: </strong><?php
echo $profile['last_name']; ?></li>

        <li><strong>First Name:
</strong><?php echo $profile['first_name'];
?></li>

        <li><strong>Middle Name:
</strong><?php echo $profile['middle_name'];
?></li>

        <li><strong>Contact No.:
</strong><?php echo $profile['contact_no'];
?></li>

        <li><strong>License No:
</strong><?php echo $vet_info['license_no'];
?></li>

        <li><strong>Specialization:
</strong><?php echo $this->model_home-
>get_specializations_str($vet_info['vet_id']
); ?></li>
        <hr>
        <div
id="clear" style="padding-
bottom:10px;"></div>
        <a href="<?php
echo
base_url("index.php/main/view_veterinarians"
); ?>" class="search_button"
style="padding:10px; text-decoration:none;
background:#F93;">Back</a>
        </div>
</div>

```

Listing 26: content/my-veterinarians.php

```

        <div id="right_content"
class="right_design border"
style="width:700px;">
        <a href=<?php echo
"\ " .
base_url("index.php/main/add_veterinarian")
. "\ " ; ?> class="search_button"
style="font-
family:Arial, Helvetica, sans-
serif;padding:5px; text-
decoration:none;margin-bottom:10px;">Add
Veterinarian</a>
        <div id="clear"><br
/></div>
        <table width="710"
border="0" style="font-family: Arial,
Helvetica, sans-serif; font-size:14px;">
        <tr >
        <th
width="200" style="border: #C0C0C0 1px
solid;"><strong>Name</strong></th>
        <th
width="200" style="border: #C0C0C0 1px
solid;"><strong>Email</strong></th>
        <th
width="200" style="border: #C0C0C0 1px
solid;"><strong>Specialization</strong></th>

```

```

        <th
width="200" style="border: #C0C0C0 1px
solid;"> <strong>Schedule</strong></th>
        <th
width="20" style="border: #C0C0C0 1px solid;
text-align: center;"><strong></strong></th>
        </tr>
        <?php foreach
($vetsList as $vet) : ?>
        <tr>
        <td
style="border: #C0C0C0 1px solid;"> <?php
        $username = $this->model_home-
>get_veterinarian_info($vet['vet_id'])['user
name'];
        $profile = $this->model_home-
>get_profile($username);
        echo $profile['first_name'] . " " .
$profile['last_name']; ?> </td>
        <td
style="border: #C0C0C0 1px solid;"> <?=
$profile['email'] ?> </td>
        <td
style="border: #C0C0C0 1px solid;"> <?=
$this->model_home-
>get_specializations_str($vet['vet_id']) ?>
</td>
        <td
style="border: #C0C0C0 1px solid; text-
align: center;"> <?= $this->model_home-
>get_schedule_str($_SESSION['clinic_id'],
$vet['vet_id'], 1) ?> </td>
        <td
style="border: #C0C0C0 1px solid;">
        <a href = <?php echo "\ " .
base_url("index.php/main/view_vet") . "/" .
$vet['vet_id'] . "\ " ; ?> >
         "
title="View Profile"width="20" height="20"
/>
        </a>
        </td>
        </tr>
        <?php endforeach; ?>
        </table>
        </div>

```

Listing 27: content/view-appointments.php

```

        <div id="right_content"
class="right_design border"
style="width:700px;">
        <form
name="account_form" action="<?=
base_url("index.php/main/disapprove_appointm
ent") ?>" onsubmit="return InputReason()"
method="post">

```

```

                                <?php if ($history ==
0) : ?>
                                <a href=<?php echo
"\ " .
base_url("index.php/main/appointment_history
") . "\ " ; ?> class="search_button"
                                style="font-
family:Arial, Helvetica, sans-
serif;padding:5px; text-
decoration:none;margin-bottom:10px;">View
Appointment History</a>
                                <?php elseif($history
== 2) : ?>
                                <a href=<?php echo
"\ " .
base_url("index.php/main/appointment_request
") . "\ " ; ?> class="search_button"
                                style="font-
family:Arial, Helvetica, sans-
serif;padding:5px; text-
decoration:none;margin-bottom:10px;">Request
Appointment</a>
                                <?php else : ?>
                                    <h3
style="background:#6CF; font-
size:14px;color:#333; border:#69C 1px solid;
padding:8px;">Appointment History</h3>
                                <?php endif; ?>
                                <div id="clear"><br
/></div>
                                <table width="710"
border="0" style="font-family: Arial,
Helvetica, sans-serif; font-size:14px;">
                                    <tr >
                                        <th
width="90" style="border: #C0C0C0 1px
solid;"><strong>Date</strong></th>
                                        <th
width="70" style="border: #C0C0C0 1px
solid;"><strong>Time</strong></th>
                                        <?php if
($SESSION['user_role'] == 3) : ?>
                                            <th
width="150" style="border: #C0C0C0 1px
solid;"><strong>Vet Name</strong></th>
                                            <?php endif;
if ($SESSION['user_role'] == 4) : ?>
                                                <th
width="150" style="border: #C0C0C0 1px
solid;"><strong>Clinic Name</strong></th>
                                                <th
width="150" style="border: #C0C0C0 1px
solid;"><strong>Vet Name</strong></th>
                                                <?php endif;
?>
                                                    <th
width="100" style="border: #C0C0C0 1px
solid;"> <strong>Pet Name</strong></th>
                                                    <th
width="200" style="border: #C0C0C0 1px
solid;"> <strong>Reason for
Visit</strong></th>
                                        <?php
if ($history == 0) : ?>
                                            <th
width="20" style="border: #C0C0C0 1px solid;
text-align: center;" colspan =
"2"><strong></strong></th>

```

```

                                <?php
elseif ($history == 2) : ?>
                                    <th
width="20" style="border: #C0C0C0 1px solid;
text-align:
center;"><strong>Status</strong></th>
                                <?php
endif; ?>
                                    </tr>
                                <?php foreach
($appointments as $appointment) : ?>
                                    <tr>
                                        <td
style="border: #C0C0C0 1px solid; text-
align: center;"> <?= $appointment['date'] ?>
                                        <td
style="border: #C0C0C0 1px solid; text-
align: center;"> <?=
substr($appointment['time'], 0, 8); ?> </td>
                                        <?php if
($SESSION['user_role'] == 3) : ?>
                                            <td
style="border: #C0C0C0 1px solid; text-
align: center;"> <?php
                                                $username = $this->model_home-
>get_owner_username($appointment['owner_id']
);
                                                $profile = $this->model_home-
>get_profile($username);
                                                echo $profile['first_name'] . " " .
$profile['last_name']; ?> </td>
                                            <?php endif;
if ($SESSION['user_role'] == 4) : ?>
                                                <!--
Clinic name -->
                                                <td
style="border: #C0C0C0 1px solid; text-
align: center;"> <?php
                                                    $clinic_info = $this->model_home-
>get_clinic_info($appointment['clinic_id']);
                                                    echo $clinic_info['clinic_name']; ?>
                                                </td>
                                                <!-- Vet name -->
                                                <td
style="border: #C0C0C0 1px solid; text-
align: center;"> <?php
                                                    $username = $this->model_home-
>get_veterinarian_info($appointment['vet_id'
])['username'];
                                                    $vet_profile = $this->model_home-
>get_profile($username);
                                                    echo $vet_profile['first_name'] . " "
. $vet_profile['last_name']; ?>
                                                </td>
                                            <?php endif;
?>
                                        <td
style="border: #C0C0C0 1px solid; text-
align: center;"> <?php echo $this-
>model_home-

```

```

>get_pet_info($appointment['pet_id'])['pet_name']; ?> </td>

                                <td
style="border: #C0C0C0 1px solid; text-align: center;"> <?=$appointment['remarks'] ?> </td>

                                <?php
if ($history == 0) : ?>
                                <td
style="border: #C0C0C0 1px solid;" id =
"test_approve">

                                <a id="approve" class="approve"
href=<?php echo "\" .
base_url("index.php/main/approve_appointment
") . "/" . $appointment['schedule_appt_id']
. "\""; ?> >

                                <img src=<?php echo "\" .
base_url("resources/images/check.png") .
\""; ?> width="20" height="20" />

                                </a>

                                </td>
                                <td
style="border: #C0C0C0 1px solid;">

                                <input type="image" alt="submit"
name="schedule_appt_id" value="<?=$appointment['schedule_appt_id'] ?>"
width="20" height="20"

                                src=" <?=$
base_url("resources/images/delete.png") ?>"
onClick="">

                                </td>
                                <?php
elseif ($history == 2) : ?>
                                <td
style="border: #C0C0C0 1px solid; text-align: center;"> <?=$appointment['confirmed'] == 0 ? 'For
Approval' : ($appointment['confirmed'] == 1
? 'Confirmed' : 'Declined') ?> </td>

                                <?php
endif; ?>

                                </tr>
                                <?php endforeach; ?>
                                <input
type="hidden" name="hidden_remarks"
id="hidden_remarks" value="" />
                                </table>
                                </form>

                                </div>

<script>
function InputReason() {
    var text = prompt("Reason for not
    approving request:", "");
    if (text === "") {
        return false;
    } else if (text) {
        $('#hidden_remarks').val(text);
        return true;
    } else {
        return false;
    }
}
}

```

```

$('#approve').click(function(){
    var check = confirm("Approve
    appointment request?");
    if (check == false) {
        return false;
    } else {
        return true;
    }
});
</script>

```

Listing 28: content/viewClinicRequests.php

```

                                <div id="right_content"
class="right_design border"
style="width:700px;">
                                <table width="710"
border="0" style="font-family: Arial,
Helvetica, sans-serif; font-size:14px;">
                                <div id="clear"></div>
                                <tr >
                                    <th
width="189" style="border: #C0C0C0 1px
solid;"><strong>Clinic Name</strong></th>
                                    <th
width="392" style="border: #C0C0C0 1px
solid;"> <strong>Clinic
Address</strong></th>
                                    <th
width="101" style="border: #C0C0C0 1px
solid;"><strong>Contact No.</strong></th>
                                    <th
width="60" style="border: #C0C0C0 1px
solid;" colspan = "2"><strong></strong></th>
                                </tr>
                                <?php foreach
($clinicsList as $clinic) : ?>
                                    <tr>
                                        <td
style="border: #C0C0C0 1px solid;"> <?=$clinic['clinic_name'] ?> </td>
                                        <td
style="border: #C0C0C0 1px solid;"> <?=$clinic['address'] ?> </td>
                                        <td
style="border: #C0C0C0 1px solid;"> <?=$clinic['contact_no'] ?> </td>
                                        <td
style="border: #C0C0C0 1px solid;" id =
"test_approve">
                                            <a href=<?php echo "\" .
base_url("index.php/main/approve_request") .
"/" . $clinic['username'] . "\""; ?> >
                                                <img src=<?php echo "\" .
base_url("resources/images/check.png") .
\""; ?> width="20" height="20" />
                                                </a>
                                            </td>
                                        </tr>
                                    <td
style="border: #C0C0C0 1px solid;">

```

```

        <a href=<?php echo "\"" .
base_url("index.php/main/delete_request") .
"/" . $clinic['username'] . "\""; ?> >

        <img src=<?php echo "\"" .
base_url("resources/images/delete.png") .
"/" ?> width="20" height="20" /> </a>
                                </td>
                                </tr>
                                </table>
                                <?php endforeach; ?>
                                </div>
</script>

function validateForm() {
    var r=confirm("Are you sure you want
to change activation of user?");
    if (r==true) {
        return true;
    }
    else {
        return false;
    }
}
</script>

```

Listing 29: content/viewClinics.php

```

        <div id="right_content"
class="right_design border"
style="width:700px;">
            <table width="710"
border="0" style="font-family: Arial,
Helvetica, sans-serif; font-size:14px;">
                <div id="clear"></div>
                <tr >
                    <th
width="189" style="border: #C0C0C0 1px
solid;"><strong>Clinic Name</strong></th>
                    <th
width="392" style="border: #C0C0C0 1px
solid;"> <strong>Location</strong></th>
                    <th
width="101" style="border: #C0C0C0 1px
solid;"><strong>Tel. Number</strong></th>
                    <th
width="60" style="border: #C0C0C0 1px
solid;"><strong>Status</strong></th>
                    <th
width="80" style="border: #C0C0C0 1px
solid;"><strong></strong></th>
                </tr>
                <form
name="account_form" action="<?=?
base_url("index.php/main/modify_activation")
?>" onsubmit="return validateForm()"
method="post">
                    <?php foreach
($clinicsList as $clinic) : ?>
                        <tr>

```

```

                                <td
style="border: #C0C0C0 1px solid;"> <?=?
$clinic['clinic_name'] ?> </td>
                                <td
style="border: #C0C0C0 1px solid;"> <?=?
$clinic['address'] ?> </td>
                                <td
style="border: #C0C0C0 1px solid;"> <?=?
$clinic['contact_no'] ?> </td>
                                <td
style="border: #C0C0C0 1px solid;"> <?=?
($clinic['status'] == 1 ? "Active" :
"Deactivated") ?> </td>
                                <td
style="border: #C0C0C0 1px solid;">
                                    <a href = <?php echo "\"" .
base_url("index.php/main/view_clinic") . "/"
. $clinic['clinic_id'] . "\""; ?> >
                                         "
title="View Clinic" width="20" height="20"
/>
                                            </a>
                                        <input type="image" alt="submit"
name="clinic_id" value="<?=?
$clinic['clinic_id'] ?>" width="20"
height="20"
                                            src=" <?=? ($clinic['status'] == 1 ?
base_url("resources/images/delete.png") .
"\ " title='Deactivate'"
                                            :
base_url("resources/images/check.png") . "\ "
title='Activate'" ) ?> onClick="">
                                                </td>
                                </tr>
                                <?php endforeach; ?>
                                </form>
                                </table>
                                </div>
</script>
function validateForm() {
    var r=confirm("Are you sure you want
to change activation of user?");
    if (r==true) {
        return true;
    }
    else {
        return false;
    }
}
</script>

```

Listing 30: content/view-consultation.php

```

        <div id="right_content"
class="inner user_design_r"
style="width:700px;" >

```

```

                <h2
style="background:#6CF; font-
size:14px;color:#333; border:#69C 1px solid;
padding:8px;">Consultation Information</h2>
                <div class="pet_info">

                    <li><strong>Consultation
Date:</strong><?php echo
$consultation['consultation_date']; ?></li>

                    <li><strong>Chief Complaint:</strong>
<?php echo $consultation['complaint'];
?></li>

                    <li><strong>Diagnosis: </strong><?php
echo $consultation['diagnosis']; ?></li>

                    <li><strong>Remarks: </strong><?php
echo $consultation['remarks'] == "" ? "< No
remarks >" : $consultation['remarks'];
?></li>

                    <li><strong>Next Consultation Date:
</strong><?php echo
$consultation['next_consult_date'] == "0000-
00-00" ? "Not required by vet." :
$consultation['next_consult_date']; ?></li>

                    <li><strong>Veterinarian:
</strong><?php

                        $username = $this->model_home-
>get_veterinarian_info($consultation['vet_id
'])['username'];

                        $profile = $this->model_home-
>get_profile($username);
echo
$profile['first_name'] . " " .
$profile['last_name']; ?></li>
                    <hr>

                </div>
                <div id="clear"><br
/><br /></div>

                <?php if
(array_filter($medicationList)) : ?>
                <div class="pet_info
border">

                    <h2
style="background:#6CF; font-
size:14px;color:#333; border:#69C 1px solid;
padding:8px;">Medications</h2><br />
                    <table
width="710" border="0" style="font-family:
Arial, Helvetica, sans-serif; font-
size:14px;">
                        <tr>

                            <th width="189" style="border:
#C0C0C0 1px solid;"><strong>Date
Prescribed</strong></th>

                            <th width="189" style="border:
#C0C0C0 1px
solid;"><strong>Medicine</strong></th>

                            <th width="289" style="border:
#C0C0C0 1px
solid;"><strong>Dosage</strong></th>

```

```

                </tr>
                <?php foreach
($medicationList as $medication) : ?>
                <tr>

                    <td style="border: #C0C0C0 1px
solid;"> <?=$medication['med_date'] ?>
</td>

                    <td style="border: #C0C0C0 1px
solid;"> <?=$medication['medicine'] ?>
</td>

                    <td style="border: #C0C0C0 1px
solid;"> <?=$medication['dosage'] ?> </td>
                </tr>
                <?php
endforeach; ?>
                </table>
                <hr>
            </div>
            <?php endif; ?>
            <?php if
(array_filter($laboratoryList)) : ?>
            <div id="clear"><br
/><br /></div>

            <div class="pet_info
border">
                <h2
style="background:#6CF; font-
size:14px;color:#333; border:#69C 1px solid;
padding:8px;">Laboratory tests</h2><br />
                <table
width="710" border="0" style="font-family:
Arial, Helvetica, sans-serif; font-
size:14px;">
                    <tr>

                        <th width="189" style="border:
#C0C0C0 1px solid;"><strong>Date
Performed</strong></th>

                        <th width="189" style="border:
#C0C0C0 1px solid;"><strong>Laboratory
Type</strong></th>

                        <th width="289" style="border:
#C0C0C0 1px solid;"><strong>Laboratory
Findings</strong></th>
                    </tr>
                    <?php foreach
($laboratoryList as $laboratory) : ?>
                    <tr>

                        <td style="border: #C0C0C0 1px
solid;"> <?=$laboratory['lab_date'] ?>
</td>

                        <td style="border: #C0C0C0 1px
solid;"> <?=$laboratory['lab_type'] ?>
</td>

                        <td style="border: #C0C0C0 1px
solid;"> <?=$laboratory['lab_findings'] ?>
</td>
                    </tr>
                <?php
endforeach; ?>

```

```

                </table>
                <hr>
            </div>
            <?php endif; ?>
            <?php if
(array_filter($vaccinationList)) : ?>
                <div id="clear"><br
/><br /></div>

                <div class="pet_info
border">
                    <h2
style="background:#6CF; font-
size:14px;color:#333; border:#69C 1px solid;
padding:8px;">Vaccinations</h2><br />
                    <table
width="710" border="0" style="font-family:
Arial, Helvetica, sans-serif; font-
size:14px;">
                        <tr>
                            <th width="189" style="border:
#C0C0C0 1px solid;"><strong>Date
Administered</strong></th>
                            <th width="189" style="border:
#C0C0C0 1px solid;"><strong>Vaccination
Type</strong></th>
                            <th width="289" style="border:
#C0C0C0 1px solid;"><strong>Vaccination
Name</strong></th>
                            <th width="289" style="border:
#C0C0C0 1px
solid;"><strong>Expiration</strong></th>
                            <th width="289" style="border:
#C0C0C0 1px solid;"><strong>Next
Due</strong></th>
                        </tr>
                        <?php foreach
($vaccinationList as $vaccination) : ?>
                            <tr>
                                <td style="border: #C0C0C0 1px
solid;"> <?=<br>
                                $consultation['consultation_date'] ?> </td>
                                <td style="border: #C0C0C0 1px
solid;"> <?=<br>
                                $vaccination['vaccine_type'] ?>
                                </td>
                                <td style="border: #C0C0C0 1px
solid;"> <?=<br>
                                $vaccination['vaccine_name'] ?>
                                </td>
                                <td style="border: #C0C0C0 1px
solid;"> <?=<br>
                                $vaccination['expiration'] ?>
                                </td>
                                <td style="border: #C0C0C0 1px
solid;"> <?=<br>
                                $vaccination['next_due'] ?>
                                </td>
                            </tr>
                        <?php
endforeach; ?>
                    </table>
                <hr>
            </div>
            <?php endif; ?>

```

```

                <div id="clear"
style="padding-bottom:10px;"><br /><br
/></div>
                <a href="<?php echo
base_url("index.php/main/view_pet") . "/" .
$consultation['pet_id']; ?>"
class="search_button" style="padding:10px;
text-decoration:none;
background:#F93;">Back</a>
            </div>

```

Listing 31: resources/css/style.css

```

@charset "utf-8";

/* CSS Document */

body{
    background: #EDEDDE url(..images/bg-
header.gif) repeat-x top center;
    line-height: 1;
    font-size: 14px;
    font-family: Arial, Helvetica, sans-
serif;
    margin: 0;
    padding: 0;
}
/*****HEADER (top part)*****/
#header {
    margin: 0 auto;
    padding: 0;
    width: 1175px;
}
#header .section {
    margin: 0;
    padding: 0;
    overflow: hidden;
}
#header .section .logo {
    float: left;
    margin: 0;
    padding: 60px 0 0 10px;
    width: 324px;
}
#header .section .logo a {
    background: url(..images/bg-
logo.png) repeat-x top left;
    color: #fff;
    display: block;
    font-family: 'Arimo-Bold';
    font-size: 60px;
    font-weight: normal;
    height: 130px;
    line-height: 130px;
    margin: 0;
    padding: 0;
    text-align:center;
    text-decoration: none;
    text-transform: uppercase;
    width: 350px;
}
#header .section ul {
    display: inline-block;
    float: right;
    margin: 0;
    overflow: hidden;
    padding: 114px 0 0;
}

```



```

}

#header .section ul li {
    background: url(../images/bg-nav-
right.gif) no-repeat top right;
    display: inline-block;
    float: left;
    list-style: none;
    margin: 0 10px;
    padding: 0;
}

#header .section ul li a {
    background: url(../images/bg-nav-
left.gif) top left; /*no-repeat */
    color: #fff;
    display: inline-block;
    font-size: 16px;
    height: 30px;
    letter-spacing: 0.05em;
    line-height: 30px;
    margin: 0;
    padding: 0 15px;
    text-align: center;
    text-decoration: none;
    text-transform: uppercase;
}

#header .section ul li a:hover {
    color: #ccc;
}

#header .section ul li.selected {
    background: url(../images/bg-nav-
right-selected.gif) top right; /*no-repeat
*/
}

#header .section ul li.selected a {
    background: url(../images/bg-nav-
left-selected.gif) top left; /*no-repeat */
}

/*****END*****/

.head_design{
    font-family: "Trebuchet MS", Arial,
Helvetica, sans-serif;
    font-size:40px;
    color:#F60;
    margin-top:20px;
}

.list{
    background: #FCFCFC;
    border
}

.search_button{
    background:#36C;
    height:50px;
    color:#FFF;
    border:#36C;
    border-radius:3px;
    width:100px;
    margin-top:15px;
}

.content{
    width: 1000px;
    margin: 0 auto;
    margin-top: 60px;
    padding: 5px;
}

.border{
    border:#CCC 1px solid;
}

#right_content{

width:600px;
float: left;
padding:10px;
min-height:320px;
margin-bottom:50px;
}

.left_design{
    background:#F9F9F9;
    border-top: #09C 40px solid;
    border-radius: 5px 5px 0px 0px;
}

.left_design ul{
    padding-top:7px;
    padding-left:5px;
}

.left_design ul li{
    display: block;
    color: white;
    margin: 3px;
    background: #FC6;
    padding:7px;
    border-radius:3px;
    font-family:Arial, Helvetica, sans-
serif;
    font-size:16px;
}

.left_design ul li a{
    color:#666;
    text-decoration:none;
}

.table_name{
    font-family:Arial, Helvetica, sans-
serif;
    font-size:17px;
    padding:5px;
    color: #6C6C6C;
}

.txt_width{
    width:400px;
    height:20px;
    padding:5px;
}

/*****LEFT SIDEBAR*****/

.content #content_left {
    width: 70px;
    height: 178px;
    background: #519194;
    border: 5px solid #5fb3bd;
    border-radius: 4px;
    float: none;
    margin: 0;
    padding: 30px 15px 19px 210px;
    position: relative;
    /*margin-left: -55px;*/
}

.content #content_left img {
    display: block;
    left: 0;
    margin: 0;
    padding: 0;
    position: absolute;
    top: -13px;
    z-index: 5;
}

/*****END*****/

```

```

/*****RIGHT SIDEBAR*****/
.content #content_right {
    background: #fff;
    border: 3px solid #5fb3bd;
    border-radius: 4px;
    min-height: 350px;
    padding: 45px 15px 15px;
    position: relative;
    width: 650px;
}
.content #content_right h2 {
    text-align: center;
}
.content #content_right h3 {
    color: #354057;
    font-size: 20px;
    font-weight: normal;
    margin: 0;
    padding: 10px 0 0;
    text-transform: none;
}
#body div .featured img {
    display: block;
    float: left;
    margin: 10px 30px 25px 0;
    padding: 0;
}
#body div .featured p {
    clear: both;
    color: #464646;
    font-size: 12px;
    line-height: 24px;
    margin: 0 0 24px;
    padding: 0;
}
#body div .featured p a {
    color: #464646;
}
#body div .featured p a:hover {
    color: #252525;
}
/*****END*****/
.user_design_l{
    width: 250px;border: #CCC 1px solid;
}
.user_design_r{
    width:650px;border:#CCC 1px solid;
}
.inner{
    background: #F7F7F7;
    font-size:14px;
    font-family:Arial, Helvetica, sans-
    serif;
    color:#333;
    margin-right:5px;
}
.text_box{
    height:40px;
    border: #A9A9A9 1px solid;
    border-radius:5px;
    width:580px;
    font-family:Arial, Helvetica, sans-
    serif;
    font-size:16px;
    padding:5px;
}
.se_lect{
    height:50px;
    border: #A9A9A9 1px solid;
    border-radius:5px;
    width:590px;
    font-family:Arial, Helvetica, sans-
    serif;
    font-size:16px;
    padding:5px;
    color: #ACACAC;
}
#left_content{
    width:300px;
    float: left;
    padding:10px;
}
.right_design{
    background:#F9F9F9;
    border-top: #09C 40px solid;
    margin-left:10px;
    border-radius: 5px 5px 0px 0px;
}
#clear{
    clear:both;
}
#footer {
    background: #4c8f96;
    border-top: 1px solid #9bd9e0;
    margin: 0;
    padding: 0;
}
#footer > div {
    margin: 0 auto;
    padding: 30px 0 28px;
    overflow: hidden;
    text-align: center;
    width: 960px;
}
#footer div p {
    color: #ededed;
    display: inline-block;
    float: right;
    font-size: 12px;
    letter-spacing: 0.05em;
    margin: 0;
    padding: 27px 0 0;
    text-align: center;
}
#footer div .connect {
    background: #387a82;
    float: right;
    height: 18px;
    margin: 0 20px 0 150px;
    padding: 22px 20px;
    overflow: hidden;
    width: 144px;
}
#footer div .connect span {
    color: #ededed;
    display: inline-block;
    float: left;
    font-size: 12px;
    height: 19px;
    line-height: 21px;
    margin: 0;
    padding: 0 0 0 39px;
}
#footer div .connect a {
    background: url(../images/icons.png)
    no-repeat;
    display: block;
    float: left;
}

```

```

        height: 20px;
        margin: 0 8px;
        padding: 0;
        text-indent: -99999px;
        width: 20px;
    }
    #footer div .connect #twitter {
        background-position: 0 -32px;
    }
    #footer div .connect #twitter:hover {
        background-position: -25px -32px;
    }
    #footer div .connect #facebook {
        background-position: 0 -57px;
    }
    #footer div .connect #facebook:hover {
        background-position: -25px -57px;
    }
    #footer div .connect #googleplus {
        background-position: 0 -82px;
    }
    #footer div .connect #googleplus:hover {
        background-position: -25px -82px;
    }
    #footer div .connect #pinterest {
        background-position: 0 -107px;
    }
    #footer div .connect #pinterest:hover {
        background-position: -25px -107px;
    }
    /*****LOGIN FOR USER*****/
    .content h2 {
        background: url(..images/bg-
        featured.png) repeat-x;
        color: #fff;
        font-family: 'Comfortaa-Regular';
        font-size: 20px;
        font-weight: normal;
        height: 47px;
        letter-spacing: 0.1em;
        left: 15px;
        line-height: 47px;
        margin: 0;
        padding: 0 25px;
        position: absolute;
        text-transform: capitalize;
        top: -14px;
    }
    .content #login {
        width: 460px;
        background: #519194;
        border: 5px solid #5fb3bd;
        border-radius: 4px;
        float: none;
        margin: 0;
        position: relative;
    }
    .content #login {
        color: #f5f5f5;
        height: 175px;
        margin-bottom: 35px;
        padding: 55px 15px 20px;
        text-align: right;
    }
    .content #login h2 {
        text-align: left;
    }
    .txt_fields{

```

```

        font-family:Arial, Helvetica, sans-
        serif;
        font-size:16px;
        /*color:#6C6C6C;*/
        height:25px;
        padding:2px;
        width:400px;
    }
    /*****END*****/
    /*****REGISTRATION*****/
    .content #registration {
        background: #fff;
        border: 3px solid #5fb3bd;
        border-radius: 4px;
        margin: 25px auto 28px;
        overflow: hidden;
        padding: 0;
        width: 936px;
    }
    .content #registration img {
        display: block;
        float: left;
        margin: 0 50px 0 0;
        padding: 0;
    }
    .content #registration h1 {
        color: #1b2130;
        font-family: 'ChangaOne';
        font-size: 40px;
        font-weight: normal;
        margin: 0;
        line-height: 50px;
        padding: 20px 20px 20px;
    }
    /*****END*****/
    /*****HOMEPAGE*****/
    .content .homepage {
        background: #fff;
        border: 5px solid #5fb3bd;
        border-radius: 4px;
        min-height: 250px;
        padding: 45px 15px 15px;
        position: relative;
        width: 620px;
        margin-bottom: 30px;
    }
    .content .homepage h2 {
        text-align: center;
    }
    .content .homepage h3 {
        color: #354057;
        font-size: 20px;
        font-weight: normal;
        margin: 0;
        padding: 10px 0 0;
        text-transform: none;
    }
    #body div .featured img {
        display: block;
        float: left;
        margin: 10px 30px 25px 0;
        padding: 0;
    }
    #body div .featured p {
        clear: both;

```

```

        color: #464646;
        font-size: 12px;
        line-height: 24px;
        margin: 0 0 24px;
        padding: 0;
    }
    #body div .featured p a {
        color: #464646;
    }
    #body div .featured p a:hover {
        color: #252525;
    }
}

/*****END*****/

/*****Warning and success
message*****/
.warning{
    background:#FFCC99;
    border:#FF6600 1px solid;
    font-family:Arial, Helvetica, sans-
serif;
    color: #333333;
    padding:5px;
    width:381px;
}
.success{
    background: #FFC;
    border: #0C0 1px solid;
    font-family:Arial, Helvetica, sans-
serif;
    color: #333333;
    padding:5px;
    width:381px;
}
.error {
    color: #D8000C;
    background-color: #FFBABA;
    width: 400px;
    padding: 2px;
}
.error_div {
    color: #ff0000;
    size: 2;
}
/*****user settings*****/
#admin_box{
    width:180px;
    border:#CCC 1px solid;
    background:#FFC;
    margin: 10px;
    padding:10px;
    float:left;
    font-family:Arial, Helvetica, sans-
serif;
    font-size:16px;
    border-radius:3px;
}
.pet_info{
    padding:5px;
    color:#333;
    line-height:20px;
    font-size:12px;
    color:#369;
    overflow-x:auto;
}
.pet_info li{
    display: list-item;

```

```

}
#pet_list{
    width:625px;
    border: #EBEBEB 1px solid;
    margin-top:20px;
    background: #FBFBFB;
    padding:10px;
}
}

```

Listing 32: resources/js/ jquery.placepicker.js

```

// obtained originally from
http://www.jqueryscript.net/form/jquery-
Location-Autocomplete-with-Google-Maps-
Places-Library-Placepicker.html
+(function ( $, window) {
    var pluginName = "placepicker";

    var defaults = {
        map: "",
        mapOptions: {
            zoom: 15
        },
        places: {
            icons: false
        },
        autoCompleteOptions: {
        },
        // callbacks
        placeChanged: null,
        location: null,
        preventSubmit: true
    };

    function PlacePicker(element,
options) {

        var instance = this;
        var geocoder = null;

        var mapElement, map, marker;

        var service = null;

        var autocomplete;

        // stores the current place
        var _place = null;
        var _latLng = null;

        /**
         * Generates a piece of HTML
         that is used to replace the input element
         * with one that also contains
         a globe button.
         */
        function template() {
            var templateString =
                "<div class=\"input-
group\">" +
                "<span class=\"input-
group-btn\">" +
                "<button data-
toggle=\"collapse\" href=\"#\" +

```

```

options.mapContainerId + "\" class=\"btn
btn-default\">" +
    "<span
class=\"glyphicon glyphicon-globe\"></span>"
+
    "</button>" +
    "</span>" +
    "</div>";

    return templateString;
}

/**
 * Modifies the DOM to add a
globe button that hides and shows a map
 * element.
 */
function initDomElements() {
    if
(!options.mapContainerId) {
        return;
    }

    // Find the index of
our element under its parent
    var $element =
$(element);
    var $parent =
$element.parent();
    var index =
$parent.children().index(element);

    // Replace the element
with our template code
    //
$element.replaceWith(template());

    // Then inject back in
the existing element. This retains all
// attributes on the
element.

    $parent.children().eq(index).append(e
lement);
}

function codePlace(query) {

    if (!query) {
        return;
    }

    var request = {
        query: query
    };

    if (service) {

        service.textSearch(request, function
(results, status) {

            if
(status ===
google.maps.places.PlacesServiceStatus.OK) {

                for (var i = 0; i < results.length;
i++) {

                    setPlace(results[i]);

                }

            }

            return;
        }
    }
}

```

```

    }
}

});

}

function codeLatLng(latLng) {

    geocoder.geocode({"latLng": latLng},
function(results, status) {

        if (status ===
google.maps.GeocoderStatus.OK) {

            if
(results[0]) {

                var place = results[0];

                setPlace(place, false);

            } else
{

                //
alert("No results found");

            }
        } else {
            //
            alert("Geocoder failed due to: " + status);
        }
    });
}

/**
 * Find and store the DOM
element that holds the Google Map.
 * @return a boolean
indicating whether an element was
successfully
 * set.
 */
function setMapElement() {

    mapElement =
$(options.map).get(0);

    if (!mapElement) {
        if
(options.mapContainerId) {

            mapElement = $("#" +
options.mapContainerId + ".placepicker-
map").get(0);

        }

        return mapElement ?
true : false;
    }

    function initMap() {

        if (!setMapElement())
{

            return;

        }

        map = new
google.maps.Map(mapElement,
options.mapOptions);

        autocomplete.bindTo("bounds", map);
    }
}

```

```

        google.maps.event.addListener(map,
"click", function(e) {
            var pos =
e.latLng;

            marker.setPosition(pos);

            map.panTo(pos);

            element.blur();

            codeLatLng(pos);
        });

        marker = new
google.maps.Marker({
            map: map,
            draggable:
true
        });

        if (options.latitude
&& options.longitude) {
            pos = new
google.maps.LatLng(options.latitude,
options.longitude);

            marker.setPosition(pos);
        }

        google.maps.event.addListener(marker,
'dragend', function(e) {
            var pos =
e.latLng;

            marker.setPosition(pos);

            map.panTo(pos);

            element.blur();

            codeLatLng(pos);
        });

        service = new
google.maps.places.PlacesService(map);

        // When the map is
made visible, if we have no location set
then
        // attempt
geolocation. The css() calls ensure that the
map is
        // refreshed.

        $(mapElement).parent().on("show.bs.co
llapse", function(e) {
            $(e.target).css("display", "block");

            if
(!element.value) {

                instance.geolocation();
            } else {

                instance.resize();
            }
        }

```

```

        $(e.target).css("display", "");
    });
}

function initAutoComplete() {
    autocomplete = new
google.maps.places.Autocomplete(element,
options.autoCompleteOptions);

    google.maps.event.addListener(autocom
plete, "place_changed", function() {
        var place =
autocomplete.getPlace();
        if
(place.geometry) {
            setPlace(place);
        }
    });

    function resizeHandler() {
        instance.resize.call(instance);
    }

    function init() {
        geocoder = new
google.maps.Geocoder();

        if (options.latitude
&& options.longitude) {
            $(options.mapOptions).prop("center",
new google.maps.LatLng(options.latitude,
options.longitude));
        }

        initDomElements();
        initAutoComplete();
        initMap();

        if (!element.value) {
            var lat =
options.latitude ||
$(options.latitudeInput).prop("value");
            var lng =
options.longitude ||
$(options.longitudeInput).prop("value");
            if (lat &&
lng) {
                instance.setLocation(lat, lng);
            }
            else {
                instance.geolocation();
            }
        } else {
            codePlace(element.value);
        }

        $(window).on("resize",
resizeHandler);

        $(element).on("keypress", function(e)
{

```

```

        if
(options.preventSubmit && e.keyCode === 13)
{
    e.preventDefault();
    e.stopImmediatePropagation();
    });
}
function setPlace(place,
updateMap) {
    updateMap = typeof
updateMap === "undefined" ? true : false;
    _place = place;
    instance.resize();
    var pos =
place.geometry.location;
    if (updateMap) {
        updatePosition(pos);
    }
    $(options.latitudeInput).prop("value"
, pos.lat());
    $(options.longitudeInput).prop("value"
, pos.lng());
    // update inputs
    if (!updateMap) {
        element.value
= place.formatted_address;
    }
    if (typeof
options.placeChanged === "function") {
        options.placeChanged.call(instance,
place);
    }
}
function updatePosition(pos)
{
    if (!map) {
        return;
    }
    map.setCenter(pos);
    var icon =
options.icon || options.placesIcon &&
place.icon ? place.icon : null;
    if (icon) {
        var
iconOptions = {
            url:
            icon,
            size:
            new google.maps.Size(71, 71),
            origin:
            new google.maps.Point(0, 0),

```

```

            anchor:
            new google.maps.Point(17, 34),
            scaledSize: new google.maps.Size(35,
            35)
        });
        marker.setIcon(iconOptions);
    }
    marker.setPosition(pos);
    marker.setVisible(true);
}
this.setValue =
function(value) {
    element.value = value;
    codePlace(value);
};
this.getValue = function() {
    return element.value;
};
this.setLocation =
function(latitude, longitude) {
    var latLng = new
google.maps.LatLng(latitude, longitude);
    this.setLatLng(latLng);
};
this.getLocation = function()
{
    var latLng =
this.getLatLng();
    if (latLng) {
        return {
            latitude: latLng.lat(),
            longitude: latLng.lng()
        };
    }
};
this.setLatLng =
function(latLng) {
    _latLng = latLng;
    codeLatLng(_latLng);
};
this.getLatLng = function() {
    if (_place &&
_place.geometry) {
        return
_place.geometry.location;
    }
    return _latLng;
};
this.getMap = function() {
    return map;
};
this.reload = function() {
    if (map) {

```

```

        codePlace(element.value);
    }
};

this.resize = function() {
    if (map) {
        var center =
map.getCenter();

        google.maps.event.trigger(map,
"resize");

        map.setCenter(center);
    }

    this.geoLocation =
function(callback) {
    // Try HTML5
geolocation

    if(navigator.geolocation) {

        navigator.geolocation.getCurrentPosit
ion(function(position) {

            var pos
= new
google.maps.LatLng(position.coords.latitude,
position.coords.longitude);

            updatePosition(pos);

            codeLatLng(pos);

            if
(callback) {

                callback(pos);

            }

        }, function()
{

            //
error

            if
(callback) {

                callback(null);

            }

        });
    } else {
        // Browser
doesn't support Geolocation

        if (callback)
{

            callback(null);

        }

    }

};

    init.call(this);
}

var pluginClass = PlacePicker;

// register plugin
$.fn[pluginName] = function(options)
{

```

```

        return this.each(function() {
            if
(!$(this).data(pluginName)) {

                $(this).data(pluginName, new
pluginClass(this, $.extend({}, defaults,
options, $(this).data())));

            }

            return $(this);

        });

    });

})( jQuery, window );

```

B. Source Code - Android

Listing 33: app/build.gradle

```

apply plugin: 'com.android.application'

android {
    compileSdkVersion 22
    buildToolsVersion "22.0.1"

    defaultConfig {
        applicationId
"ph.edu.upm.cjgaleng.spotavet"
        minSdkVersion 14
        targetSdkVersion 22
        versionCode 1
        versionName "1.0"
    }
    buildTypes {
        release {
            minifyEnabled false
            proguardFiles
getDefaultProguardFile('proguard-
android.txt'), 'proguard-rules.pro'
        }
    }
}

dependencies {
    compile fileTree(dir: 'libs', include:
['*.jar'])
    compile 'com.android.support:appcompat-
v7:22.2.0'
    compile 'com.google.android.gms:play-
services:6.5.87'
    compile 'com.android.support:cardview-
v7:21.0.+'
}

<?xml version="1.0" encoding="utf-8"?>
<manifest
xmlns:android="http://schemas.android.com/ap
k/res/android"

```

Listing 34: src/main/AndroidManifest.xml


```

package="ph.edu.upm.cjgaleng.spotavet" >

<uses-feature
    android:glEsVersion="0x00020000"
    android:required="true" />

    <!-- Used by the API to download map
tiles from Google Maps servers. -->
    <uses-permission
android:name="android.permission.INTERNET"
/>
    <!-- Allows the API to check the
connection status in order to determine
whether data can be downloaded. -->
    <uses-permission
android:name="android.permission.ACCESS_NETW
ORK_STATE" />
    <!-- Allows the API to cache map tile
data in the device's external storage area.
-->
    <uses-permission
android:name="android.permission.WRITE_EXTER
NAL_STORAGE" />
    <uses-permission
android:name="com.google.android.providers.g
sf.permission.READ_GSERVICES" />
    <!--
        The
ACCESS_COARSE/FINE_LOCATION permissions are
not required to use
        Google Maps Android API v2, but are
recommended.
    -->
    <!-- Allows the API to use WiFi or
mobile cell data (or both) to determine the
device's location. -->
    <uses-permission
android:name="android.permission.ACCESS_COAR
SE_LOCATION" />
    <!-- Allows the API to use the Global
Positioning System (GPS) to determine the
device's location to within a very small
area. -->
    <uses-permission
android:name="android.permission.ACCESS_FINE
_LOCATION" />

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >
        <meta-data

android:name="com.google.android.gms.version
"

android:value="@integer/google_play_services
_version" />
        <meta-data

android:name="com.google.android.geo.API_KEY
"

android:value="AIzaSyDIK_s6IEhjLf9sBf0j3fbM3
mGC08BFW40" />

        <activity
            android:name=".MainActivity"
            android:label="@string/app_name"
            android:screenOrientation="portrait" >
                <intent-filter>
                    <action
android:name="android.intent.action.MAIN" />

                    <category
android:name="android.intent.category.LAUNCH
ER" />
                </intent-filter>
            </activity>
            <activity
android:name=".SearchMapActivity"

android:label="@string/title_activity_search
_map"

android:screenOrientation="portrait" >
            </activity>
            <activity
android:name=".ClinicInfoActivity"

android:label="@string/title_activity_clinic
_info"

android:screenOrientation="portrait" >
            </activity>
            <activity
                android:name=".VetInfoActivity"

android:label="@string/title_activity_vet_in
fo"

android:screenOrientation="portrait" >
            </activity>
            <activity
                android:name=".ListNearestClinicsActivity"

android:label="@string/title_activity_list_n
earest_clinics" >
            </activity>
        </application>
    </manifest>

```

Listing 35: MainActivity.java

```

package ph.edu.upm.cjgaleng.spotavet;

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

import android.app.ProgressDialog;
import android.content.Intent;
import android.location.Address;
import android.location.Geocoder;
import android.location.Location;
import android.os.AsyncTask;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;

```

```

import android.util.Log;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.ImageButton;
import android.widget.TextView;
import android.widget.Toast;

import
com.google.android.gms.common.ConnectionResu
lt;
import
com.google.android.gms.common.GooglePlayServ
icesUtil;
import
com.google.android.gms.common.api.GoogleApiC
lient;
import
com.google.android.gms.common.api.GoogleApiC
lient.ConnectionCallbacks;
import
com.google.android.gms.common.api.GoogleApiC
lient.OnConnectionFailedListener;
import
com.google.android.gms.location.LocationRequ
est;
import
com.google.android.gms.location.LocationServ
ices;
import
com.google.android.gms.location.LocationList
ener;
import
com.google.android.gms.maps.CameraUpdateFact
ory;
import
com.google.android.gms.maps.GoogleMap;
import
com.google.android.gms.maps.GoogleMap.OnMapC
lickListener;
import
com.google.android.gms.maps.GoogleMap.OnCame
raChangeListener;
import
com.google.android.gms.maps.MapFragment;
import
com.google.android.gms.maps.SupportMapFragme
nt;
import
com.google.android.gms.maps.model.BitmapDesc
riptorFactory;
import
com.google.android.gms.maps.model.CameraPosi
tion;
import
com.google.android.gms.maps.model.LatLng;
import
com.google.android.gms.maps.model.Marker;
import
com.google.android.gms.maps.model.MarkerOpti
ons;
import
com.google.android.gms.maps.model.VisibleReg
ion;

import java.io.IOException;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;

public class MainActivity extends
AppCompatActivity implements
ConnectionCallbacks,
OnConnectionFailedListener,
LocationListener, OnClickListener,
OnMapClickListener, OnCameraChangeListener {

    private GoogleMap gMap;
    MarkerOptions marker;
    private Button btnChangeAddress,
btnMapList, btnFilter;
    private ImageButton btnSearchClinic,
btnShowLocation;
    private TextView txtLatlng, txtAddress;
    private ProgressDialog pDialog;

    // server IP address
    private static final String CLINICS_URL
=
"http://192.168.43.149:80/SpotAVet/resources
/json/nearest-clinics.php";

    // JSON parser class
    JSONParser jsonParser = new
JSONParser();

    private static final String TAG =
MainActivity.class.getSimpleName();
    private final static int
PLAY_SERVICES_RESOLUTION_REQUEST = 1000;

    private Location mLastLocation;
    private GoogleApiClient
mGoogleApiClient;
    double myLat, myLng;

    static ArrayList<HashMap<String,
String>> clinicsList;

    LatLng upperLeft, lowerRight;

    private LocationRequest
mLocationRequest;

    // Location updates intervals in sec
    private static int UPDATE_INTERVAL =
10000; // 10 sec
    private static int FASTEST_INTERVAL =
5000; // 5 sec
    private static int DISPLACEMENT = 10; //
10 meters

    ConnectionDetector connectionDetector;
    // flag for Internet connection status
    Boolean isInternetPresent = false;
    // Alert Dialog Manager
    AlertDialogManager alert = new
AlertDialogManager();

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

        setContentView(R.layout.activity_main);

        btnChangeAddress = (Button)
findViewById(R.id.btnChangeAddress);

```

```

        btnSearchClinic = (ImageButton)
findViewById(R.id.btnSearchClinic);
        btnShowLocation = (ImageButton)
findViewById(R.id.btnShowLocation);
        btnMapList = (Button)
findViewById(R.id.btnMapList);
        btnFilter = (Button)
findViewById(R.id.btnFilter);

        txtLatlng =
(TextView) findViewById(R.id.textView_latlng)
;
        txtAddress =
(TextView) findViewById(R.id.textView_address
);
        txtLatlng.bringToFront();
        txtAddress.bringToFront();

btnChangeAddress.setOnClickListener(this);
btnSearchClinic.setOnClickListener(this);

btnShowLocation.setOnClickListener(this);
        btnMapList.setOnClickListener(this);
        btnFilter.setOnClickListener(this);

        connectionDetector = new
ConnectionDetector(getApplicationContext());

        // Check if Internet is present
        isInternetPresent =
connectionDetector.isConnectingToInternet();
        if (!isInternetPresent) {
            // Internet Connection is not
present

alert.showAlertDialog(MainActivity.this,
"Internet Connection Error",
                "Please connect to
working Internet connection", false);
            // stop executing code by return
return;
        }

        if (checkPlayServices()) {
            buildGoogleApiClient();
            createLocationRequest();
        }

        setUpMapIfNeeded();
    }

    @Override
    public void onClick(View v) {
        Intent i;
        switch (v.getId()) {
            case R.id.btnChangeAddress :
                i = new Intent(this,
SearchMapActivity.class);
                startActivity(i);
            //
startActivityForResult(nextScreen, 0);
            break;
            case R.id.btnSearchClinic :
                VisibleRegion visibleRegion
= gMap.getProjection().getVisibleRegion();
                upperLeft =
visibleRegion.farLeft;
                lowerRight =
visibleRegion.nearRight;

                new
NearestClinics().execute();
                break;
            case R.id.btnShowLocation:
                displayLocation();
                createLocationRequest();
                break;
            case R.id.btnMapList:
                break;
            case R.id.btnFilter:
                i = new Intent(this,
ListNearestClinicsActivity.class);
                startActivity(i);
                break;
            default:
                }
        }

        @Override
        public void onMapClick(LatLng latLng) {
            btnFilter.setEnabled(false);

gMap.setOnCameraChangeListener(this);

            myLat = latLng.latitude;
            myLng = latLng.longitude;
            gMap.clear();

gMap.animateCamera(CameraUpdateFactory.newLa
tLng(new LatLng(myLat, myLng)));

            marker = new
MarkerOptions().position(new LatLng(myLat,
myLng));

            marker.setIcon(BitmapDescriptorFactory.fromReso
urce(R.drawable.userlocation));
            gMap.addMarker(marker);

            new
GetAddressFromLatLng().execute();
        }

        @Override
        public void
onCameraChange(CameraPosition
cameraPosition) {
            VisibleRegion vr =
gMap.getProjection().getVisibleRegion();

            LatLng topLeft = vr.farLeft;
            LatLng bottomRight = vr.nearRight;

            if ((myLat > topLeft.latitude) ||
(myLat < bottomRight.latitude) ||
                ((myLng < topLeft.longitude)
|| (myLng > bottomRight.longitude))) {

                btnSearchClinic.setEnabled(false);
            }
            else {

                btnSearchClinic.setEnabled(true);
            }
        }
    }
    /**

```

```

    * Sets up the map if it is possible to
do so (i.e., the Google Play services APK is
correctly
    * installed) and the map has not
already been instantiated.. This will ensure
that we only ever
    * call {@link #setUpMap()} once when
{@link #gMaps} is not null.
    * <p/>
    * If it isn't installed {@link
SupportMapFragment} (and
    * {@link
com.google.android.gms.maps.MapView
MapView}) will show a prompt for the user to
    * install/update the Google Play
services APK on their device.
    * <p/>
    * A user can return to this
FragmentActivity after following the prompt
and correctly
    * installing/updating/enabling the
Google Play services. Since the
FragmentActivity may not
    * have been completely destroyed during
this process (it is likely that it would
only be
    * stopped or paused), {@link
#onCreate(Bundle)} may not be called again
so we should call this
    * method in {@link #onResume()} to
guarantee that it will be called.
    */
    private void setUpMapIfNeeded() {
        // Do a null check to confirm that
we have not already instantiated the map.
        if (gMaps == null) {
            // Try to obtain the map from
the SupportMapFragment.
            gMap = ((MapFragment)
getFragmentManager().findFragmentById(R.id.ma
p))
                .getMap();
            // Check if we were successful
in obtaining the map.
            if (gMaps != null) {
                setUpMap();
            }
        }
    }

    /**
    * This is where we can add markers or
lines, add listeners or move the camera. In
this case, we
    * just add a marker near Africa.
    * <p/>
    * This should only be called once and
when we are sure that {@link #gMaps} is not
null.
    */
    private void setUpMap() {

gMaps.setMapType(GoogleMap.MAP_TYPE_NORMAL);

        marker = new
MarkerOptions().position(new
LatLng(14.5567427, 121.024322));

marker.setIcon(BitmapDescriptorFactory.defaultM
arker(BitmapDescriptorFactory.HUE_ROSE));
        //gMaps.addMarker(marker);

        gMap.setMyLocationEnabled(true);

gMaps.getUiSettings().setZoomControlsEnabled(
true);

gMaps.getUiSettings().setCompassEnabled(false
);

gMaps.getUiSettings().setRotateGesturesEnable
d(false);

gMaps.getUiSettings().setMyLocationButtonEnab
led(false);

        gMap.setOnMapClickListener(this);

gMaps.setOnCameraChangeListener(this);
    }

    /**
    * Method to verify google play services
on the device
    * @return
    */
    private boolean checkPlayServices() {
        int resultCode =
GooglePlayServicesUtil.isGooglePlayServicesA
vailable(this);

        if (resultCode !=
ConnectionResult.SUCCESS) {
            if
(GooglePlayServicesUtil.isUserRecoverableErr
or(resultCode)) {

GooglePlayServicesUtil.getErrorDialog(result
Code, this,
PLAY_SERVICES_RESOLUTION_REQUEST).show();
            }
            else {

Toast.makeText(getApplicationContext(),
"This device is not supported.",
Toast.LENGTH_LONG).show();
                finish();
            }
            return false;
        }
        return true;
    }

    /**
    * Creating google api client object
    */
    protected synchronized void
buildGoogleApiClient() {
        mGoogleApiClient = new
GoogleApiClient.Builder(this)

.addConnectionCallbacks(this)

.addOnConnectionFailedListener(this)

.addApi(LocationServices.API).build();
    }

    private void displayLocation() {

```

```

        mLastLocation =
LocationServices.FusedLocationApi.getLastLoc
ation(mGoogleApiClient);
        btnFilter.setEnabled(false);
        //btnSearchClinic.setEnabled(false);

gMap.setOnCameraChangeListener(this);

        if (mLastLocation != null) {
            myLat=
mLastLocation.getLatitude();
            myLng=
mLastLocation.getLongitude();

            gMap.clear();

gMap.moveCamera(CameraUpdateFactory.newLatLn
gZoom(new LatLng(myLat, myLng), 15));
            marker = new
MarkerOptions().position(new LatLng(myLat,
myLng));

marker.setIcon(BitmapDescriptorFactory.fromReso
urce(R.drawable.userlocation));
            gMap.addMarker(marker);

            new
GetAddressFromLatLng().execute();
        }

        private class GetAddressFromLatLng
extends AsyncTask<String, String, String> {

            String mainAddress = "";

            @Override
            protected void onPreExecute() {
                super.onPreExecute();
            }

            @Override
            protected String
doInBackground(String... params) {
                Geocoder geocoder = new
Geocoder(getApplicationContext());
                String result = null;

                try {
                    List<Address> addressList =
geocoder.getFromLocation(
                        myLat, myLng, 1);
                    if (addressList != null &&
addressList.size() > 0) {
                        Address address =
addressList.get(0);
                        StringBuilder sb = new
StringBuilder();
                        for (int i = 0; i <
address.getMaxAddressLineIndex(); i++) {

                            sb.append(address.getAddressLine(i)).append(
", ");
                        }
                        mainAddress =
sb.toString();
                        sb = new
StringBuilder();
                        sb.append("Locality :
").append(address.getLocality()).append("\n"
);
                    }
                }
            }
        }

```

```

        sb.append("Postal Code :
").append(address.getPostalCode()).append("\n
");
        sb.append("Country :
").append(address.getCountryName());
        result = sb.toString();
    }
    return result;
} catch (IOException e) {
    e.printStackTrace();
}
return null;
}

@Override
protected void onPostExecute(String
result) {
    btnSearchClinic.setEnabled(true);
    txtLatLng.setText("Lat: " +
(float) myLat + " Lng: " + (float) myLng);
    if (result != null) {
        result = "\nAddress:\n" +
result;
    }
    txtAddress.setText(mainAddress);
    } else {
        result = "\n Unable to get
address for this lat-long.";
    }
}

class NearestClinics extends
AsyncTask<String, String, String> {

    /**
     * Before starting background thread
     Show Progress Dialog
     */
    JSONArray clinicArray = null;
    HashMap<LatLng, String>
clinicMarkers;

    @Override
    protected void onPreExecute() {
        super.onPreExecute();
        btnFilter.setEnabled(false);
    }

    btnSearchClinic.setEnabled(false);

    pDialog = new
ProgressDialog(MainActivity.this);
    pDialog.setMessage("Getting
clinics...");
    pDialog.setIndeterminate(false);
    pDialog.setCancelable(true);
    pDialog.show();
}

@Override
protected String
doInBackground(String... args) {
    // Check for success tag
    int success;
    try {
        // Building Parameters
        List<NameValuePair> params =
new ArrayList<NameValuePair>();
    }
}

```

```

        params.add(new
BasicNameValuePair("north",
Double.toString(upperLeft.latitude));
        params.add(new
BasicNameValuePair("south",
Double.toString(lowerRight.latitude));
        params.add(new
BasicNameValuePair("west",
Double.toString(upperLeft.longitude));
        params.add(new
BasicNameValuePair("east",
Double.toString(lowerRight.longitude));
        params.add(new
BasicNameValuePair("mylat",
Double.toString(myLat));
        params.add(new
BasicNameValuePair("mylng",
Double.toString(myLng));

        clinicsList = new
ArrayList<HashMap<String, String>>();
        clinicMarkers = new
HashMap<>();

        Log.d("request!",
"starting");

        // getting product details
by making HTTP request
        JSONObject json =
jsonParser.makeHttpRequest(
                CLINICS_URL ,
"POST", params);

        // check your log for json
response
        Log.d("Search clinics
attempt", json.toString());

        // json success tag
success =
json.getInt("success");
        if (success == 1) {
            clinicArray =
json.getJSONArray("clinics");

            for (int i = 0; i <
clinicArray.length(); i++) {
                JSONObject c =
clinicArray.getJSONObject(i);

                // Storing each json
item in variable
                String clinic_id =
c.getString("clinic_id");
                String clinic_name =
c.getString("clinic_name");
                String address =
c.getString("address");
                String lat =
c.getString("x");
                String lng =
c.getString("y");

                // creating new
HashMap
                HashMap<String,
String> map = new HashMap<String, String>();

                // adding each child
node to HashMap key => value

                map.put("clinic_id",
clinic_id);
                map.put("clinic_name", clinic_name);
                map.put("address",
address);
                map.put("x", lat);
                map.put("y", lng);

                // adding HashList
to ArrayList
                clinicsList.add(map);

                Double latitude =
Double.parseDouble(lat);
                Double longitude =
Double.parseDouble(lng);

                clinicMarkers.put(new LatLng(latitude,
longitude), clinic_id);
            }

            return
json.getString("message");
        } else {
            Log.d("No clinics in
area", json.getString("message"));
            return
json.getString("message");
        }
    } catch (JSONException e) {
        e.printStackTrace();
    }

    return null;
}

/**
 * After completing background task,
dismiss the progress dialog
 * @param file_url
 */
protected void onPostExecute(String
file_url) {
    // dismiss the dialog once
product deleted
    pDialog.dismiss();

    Toast.makeText(getApplicationContext(),
file_url, Toast.LENGTH_SHORT).show();
    btnFilter.setEnabled(true);

    gMap.setOnCameraChangeListener(null);
    gMap.clear();

    marker = new
MarkerOptions().position(new LatLng(myLat,
myLng));

    marker.setIcon(BitmapDescriptorFactory.fromReso
urce(R.drawable.userlocationgray));
    gMap.addMarker(marker);

    for (int i = 0; i <
clinicsList.size(); i++) {

```

```

        final HashMap<String,
String> clinic_info = clinicsList.get(i);
        double lat =
Double.parseDouble(clinic_info.get("x"));
        double lng =
Double.parseDouble(clinic_info.get("y"));

        marker = new
MarkerOptions().position(new LatLng(lat,
lng));

marker.icon(BitmapDescriptorFactory.defaultM
arker(BitmapDescriptorFactory.HUE_AZURE));
        if (i == 0)

marker.icon(BitmapDescriptorFactory.defaultM
arker(BitmapDescriptorFactory.HUE_ROSE));

marker.title(clinic_info.get("clinic_name"))
;

marker.snippet(clinic_info.get("address"));
        gMap.addMarker(marker);
    }

gMap.setOnInfoWindowClickListener(new
GoogleMap.OnInfoWindowClickListener() {
    @Override
    public void
onInfoWindowClick(Marker marker) {
        Intent nextScreen = new
Intent(MainActivity.this,
ClinicInfoActivity.class);

nextScreen.putExtra("clinic_id",
clinicMarkers.get(marker.getPosition()));

startActivity(nextScreen);
    }
});

/*gMap.setInfoWindowAdapter(new
GoogleMap.InfoWindowAdapter() {
    public View
getInfoWindow(Marker arg0) {
        View v =
getLayoutInflater().inflate(R.layout.custom_
infowindow, null);
        return v;
    }

    @Override
    public View
getInfoContents(Marker marker) {
        return null;
    }
});*/
}

@Override
public boolean onCreateOptionsMenu(Menu
menu) {
    // Inflate the menu; this adds items
to the action bar if it is present.

//getMenuInflater().inflate(R.menu.menu_main
, menu);
    return true;
}

}

@Override
public boolean
onOptionsItemSelected(MenuItem item) {
    // Handle action bar item clicks
here. The action bar will
// automatically handle clicks on
the Home/Up button, so long
// as you specify a parent activity
in AndroidManifest.xml.
    int id = item.getItemId();

    //noinspection
SimplifiableIfStatement
    if (id == R.id.action_settings) {
        return true;
    }

    return
super.onOptionsItemSelected(item);
}

@Override
protected void onStart() {
    super.onStart();
    if(mGoogleApiClient != null) {
        mGoogleApiClient.connect();
    }
}

@Override
protected void onResume() {
    super.onResume();
    setUpMapIfNeeded();
    checkPlayServices();
}

@Override
protected void onPause() {
    super.onPause();
}

@Override
public void onConnectionSuspended(int
arg0) {
    mGoogleApiClient.connect();
}

/**
 * Google api callback methods
 */
@Override
public void
onConnectionFailed(ConnectionResult result)
{
    Log.i(TAG, "Connection failed:
ConnectionResult.getErrorCode() = "
+ result.getErrorCode());
}

@Override
public void onConnected(Bundle arg0) {
    // Once connected with google api,
get the location
displayLocation();
}
}

```

```

/**
 * Creating location request object
 * */
protected void createLocationRequest() {
    mLocationRequest = new
LocationRequest();

mLocationRequest.setInterval(UPDATE_INTERVAL
);

mLocationRequest.setFastestInterval(FATEST_
INTERVAL);

mLocationRequest.setPriority(LocationRequest
.PRIORITY_HIGH_ACCURACY);

mLocationRequest.setSmallestDisplacement(DIS
PLACEMENT); // 10 meters
    }

@Override
    public void onLocationChanged(Location
location) {
        // Assign the new location
        mLastLocation = location;

Toast.makeText(getApplicationContext(),
"Location changed!",
        Toast.LENGTH_SHORT).show();

        // Displaying the new location on UI
        displayLocation();
    }
}

```

Listing 36: AlertDialogManager.java

```

package ph.edu.upm.cjgaleng.spotavet;

import android.app.AlertDialog;
import android.content.Context;
import android.content.DialogInterface;

/**
 * Created by CJ on 6/4/2015.
 */
public class AlertDialogManager {
    /**
     * Function to display simple Alert
     Dialog
     * @param context - application context
     * @param title - alert dialog title
     * @param message - alert message
     * @param status - success/failure (used
     to set icon)
     * - pass null if you
     don't want icon
     * */
    public void showAlertDialog(Context
context, String title, String message,
        Boolean
status) {
        AlertDialog alertDialog = new
AlertDialog.Builder(context).create();

        // Setting Dialog Title
        alertDialog.setTitle(title);

```

```

// Setting Dialog Message
        alertDialog.setMessage(message);

        // Setting OK Button
        alertDialog.setButton("OK", new
DialogInterface.OnClickListener() {
            public void
onClick(DialogInterface dialog, int which) {
                }
            });

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

```

Listing 37: ClinicInfoActivity.java

```

package ph.edu.upm.cjgaleng.spotavet;

import android.app.FragmentManager;
import android.app.FragmentTransaction;
import android.app.ProgressDialog;
import android.content.Intent;
import android.os.AsyncTask;
import android.os.Bundle;
import
android.support.v7.app.AppCompatActivity;
import android.util.Log;
import android.view.Menu;
import android.view.MenuItem;
import android.widget.EditText;

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

import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;

public class ClinicInfoActivity extends
AppCompatActivity {
    EditText txtCName, txtCAddress,
txtCSchedule, txtCContactNo,txtCEmail;
    private ProgressDialog pDialog;

    // change to server IP address
    private static final String CLINICS_URL
=
"http://192.168.43.149:80/SpotAVet/resources
/json/clinic-info.php";

    // JSON parser class
    JSONParser jsonParser = new
JSONParser();

    private String clinic_id;

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

```



```

setContentView(R.layout.activity_clinic_info
);

    ListVetsFragment listVetsFragment =
new ListVetsFragment();

    getSupportActionBar().hide();
    FragmentManager fm =
getFragmentManager();
    FragmentTransaction ft =
fm.beginTransaction();
    ft.add(R.id.linearLayout1,
listVetsFragment);
    ft.commit();

    txtCName = (EditText)
findViewById(R.id.txtCName);
    txtCAddress = (EditText)
findViewById(R.id.txtCAddress);
    txtCSchedule = (EditText)
findViewById(R.id.txtCSchedule);
    txtCContactNo = (EditText)
findViewById(R.id.txtCContactNo);
    txtCEmail = (EditText)
findViewById(R.id.txtCEmail);

    Intent intent = getIntent();
    clinic_id =
intent.getStringExtra("clinic_id");

    new ClinicInformation().execute();
}

class ClinicInformation extends
AsyncTask<String, String, String> {

    /**
     * Before starting background thread
     Show Progress Dialog
     */
    boolean failure = false;
    HashMap<String, String> clinicInfo;

    @Override
    protected void onPreExecute() {
        super.onPreExecute();
        pDialog = new
ProgressDialog(ClinicInfoActivity.this);
        pDialog.setMessage("Getting
clinic info...");
        pDialog.setIndeterminate(false);
        pDialog.setCancelable(true);
        pDialog.show();
    }

    @Override
    protected String
doInBackground(String... args) {
        // Check for success tag
        int success;
        try {
            // Building Parameters
            List<NameValuePair> params =
new ArrayList<NameValuePair>();
            clinicInfo = new
HashMap<String, String>();
            params.add(new
BasicNameValuePair("clinic_id", clinic_id));

```

```

        Log.d("request!",
"starting");

        // getting product details
        by making HTTP request
        JSONObject json =
jsonParser.makeHttpRequest(
            CLINICS_URL ,
            "POST", params);

        // check your log for json
        response
        Log.d("Clinic info attempt",
json.toString());

        // json success tag
        success =
json.getInt("success");
        if (success == 1) {
            // Storing each json
            item in variable
            String clinic_id =
json.getString("clinic_id");
            String clinic_name =
json.getString("clinic_name");
            String address =
json.getString("address");
            String schedule =
json.getString("schedule");
            String contact_no =
json.getString("contact_no");
            String email =
json.getString("email");

            // adding each child
            node to HashMap key => value
            clinicInfo.put("clinic_id", clinic_id);
            clinicInfo.put("clinic_name", clinic_name);
            clinicInfo.put("address", address);
            clinicInfo.put("schedule", schedule);
            clinicInfo.put("contact_no", contact_no);
            clinicInfo.put("email",
email);

            return
json.getString("message");
        } else {
            Log.d("No clinic!",
json.getString("message"));
            return
json.getString("message");
        }
    } catch (JSONException e) {
        e.printStackTrace();
    }

    return null;
}

/**
 * After completing background task,
dismiss the progress dialog
 * @param file_url
 */
protected void onPostExecute(String
file_url) {

```

```

        // dismiss the dialog once
        product deleted
        pDialog.dismiss();

txtCName.setText(clinicInfo.get("clinic_name
"));

txtCAddress.setText(clinicInfo.get("address"
));

txtCSchedule.setText(clinicInfo.get("schedul
e"));

txtCContactNo.setText(clinicInfo.get("contac
t_no"));

txtCEmail.setText(clinicInfo.get("email"));

    }

    @Override
    public boolean onCreateOptionsMenu(Menu
menu) {
        // Inflate the menu; this adds items
to the action bar if it is present.

getMenuInflater().inflate(R.menu.menu_clinic
_info, menu);
        return true;
    }

    @Override
    public boolean
onOptionsItemSelected() {
        // Handle action bar item clicks
here. The action bar will
        // automatically handle clicks on
the Home/Up button, so long
        // as you specify a parent activity
in AndroidManifest.xml.
        int id = item.getItemId();

        //noinspection
SimpleifiableIfStatement
        if (id == R.id.action_settings) {
            return true;
        }

        return
super.onOptionsItemSelected(item);
    }
}

```

Listing 38: ConnectionDetector.java

```

package ph.edu.upm.cjgaleng.spotavet;

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.CONNECTIVI
TY_SERVICE);
        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;
        }
    }
}

```

Listing 39: GPlaceAPI.java

```

package ph.edu.upm.cjgaleng.spotavet;

import android.util.Log;

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

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;

/**
 * Created by CJ on 6/4/2015.
 */
public class GPlaceAPI {
    private static final String TAG =
GPlaceAPI.class.getSimpleName();

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

```

```

// Go to
https://console.developers.google.com/ and
get your Android API Key
// example : private static final String
API_KEY = "AIzaSyBdVl-
cTICSwYKrZ95SuvNw7dbMuDt1KG0"
private static final String API_KEY =
"";
public String place_id;

public ArrayList<String> autocomplete
(String input) {
    ArrayList<String> resultList = null;
    place_id = "";

    HttpURLConnection conn = null;
    StringBuilder jsonResults = new
StringBuilder();

    try {
        StringBuilder sb = new
StringBuilder(PLACES_API_BASE +
TYPE_AUTOCOMPLETE + OUT_JSON);
        sb.append("?key=" + API_KEY);
        sb.append("&types=geocode");
        sb.append("&input=" +
URLEncoder.encode(input, "utf8"));

        URL url = new
URL(sb.toString());
        conn = (HttpURLConnection)
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(TAG, "Error processing
Places API URL", e);
        return resultList;
    } catch (IOException e) {
        Log.e(TAG, "Error connecting to
Places API", e);
        return resultList;
    } finally {
        if (conn != null) {
            conn.disconnect();
        }
    }

    try {
        // Log.d(TAG,
jsonResults.toString());

        // 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).getJSONObject("description"));
            place_id =
predsJSONArray.getJSONObject(i).getString("p
lace_id");
        }
    } catch (JSONException e) {
        Log.e(TAG, "Cannot process JSON
results", e);
    }
    return resultList;
}
}

```

Listing 40: JSONParser.java

```

package ph.edu.upm.cjgaleng.spotavet;

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.ClientProtocolExcepti
on;
import
org.apache.http.client.entity.UrlEncodedForm
Entity;
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.DefaultHttpClie
nt;
import org.json.JSONException;
import org.json.JSONObject;

import android.util.Log;

public class JSONParser {
    static InputStream is = null;
    static JSONObject jsonObj = null;
    static String json = "";

    // constructor
    public JSONParser() {
    }

    // function get json from url
    // by making HTTP POST or GET mehtod
    public JSONObject makeHttpRequest(String
url, String method,

```

```

List<NameValuePair> params) {

    // Making HTTP request
    try {

        // check for request method
        if(method.equals("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.equals("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());
    }
}

```

```

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

// return JSON String
return jsonObj;
}
}

```

Listing 41: ListNearestClinicsActivity.java

```

package ph.edu.upm.cjgaleng.spotavet;

import android.app.FragmentManager;
import android.app.FragmentTransaction;
import android.content.Intent;
import
android.support.v7.app.ActionBarActivity;
import android.os.Bundle;
import
android.support.v7.app.AppCompatActivity;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.Button;

public class ListNearestClinicsActivity
extends AppCompatActivity implements
View.OnClickListener {
    private Button btnMapList2, btnFilter2;

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

        setContentView(R.layout.activity_list_neares
t_clinics);
        btnMapList2 = (Button)
findViewById(R.id.btnMapList2);

        btnMapList2.setOnClickListener(this);

        ListNearestClinicsActivityFragment
listNearestClinicsActivityFragment = new
ListNearestClinicsActivityFragment();
        FragmentManager fm =
getManager();
        FragmentTransaction ft =
fm.beginTransaction();

        ft.add(R.id.layoutNearestClinicsList,
listNearestClinicsActivityFragment);
        ft.commit();
    }
}

```

```

    @Override
    public boolean onCreateOptionsMenu(Menu
menu) {
        // Inflate the menu; this adds items
to the action bar if it is present.

getMenuInflater().inflate(R.menu.menu_list_n
earest_clinics, menu);
        return true;
    }

    @Override
    public boolean
onOptionsItemSelected(MenuItem item) {
        // Handle action bar item clicks
here. The action bar will
        // automatically handle clicks on
the Home/Up button, so long
        // as you specify a parent activity
in AndroidManifest.xml.
        int id = item.getItemId();

        //noinspection
SimplifiableIfStatement
        if (id == R.id.action_settings) {
            return true;
        }

        return
super.onOptionsItemSelected(item);
    }

    @Override
    public void onClick(View v) {
        Intent i;
        switch (v.getId()) {
            case R.id.btnMapList2:
                finish();
                break;
            case R.id.btnFilter2:
                break;
            default:
        }
    }
}

```

Listing 42:

ListNearestClinicsActivityFragment.java

```

package ph.edu.upm.cjgaleng.spotavet;

import android.app.ListFragment;
import android.app.ProgressDialog;
import android.content.Intent;
import android.os.AsyncTask;
import android.support.v4.app.Fragment;
import android.os.Bundle;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.ListAdapter;
import android.widget.ListView;
import android.widget.SimpleAdapter;
import android.widget.Toast;

```

```

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

import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;

/**
 * A placeholder fragment containing a
simple view.
 */
public class
ListNearestClinicsActivityFragment extends
ListFragment {
    private ProgressDialog pDialog;
    private Button btnMapList2, btnFilter2;

    public
ListNearestClinicsActivityFragment() {
    }

    @Override
    public View onCreateView(LayoutInflater
inflater, ViewGroup container,
Bundle
savedInstanceState) {
        View myInflatedView =
inflater.inflate(R.layout.fragment_list_near
est_clinics, container,false);

        return myInflatedView;
    }

    @Override
    public void onActivityCreated (Bundle
savedInstanceState){
super.onActivityCreated(savedInstanceState);
        btnMapList2 = (Button)
getActivity().findViewById(R.id.btnMapList2)
;
        btnFilter2 = (Button)
getActivity().findViewById(R.id.btnFilter2);
    }

    public void onResume() {
        // TODO Auto-generated method stub
super.onResume();

        //loading the comments via AsyncTask
new GetClinicList().execute();
    }

    @Override
    public void onItemClick(ListView l,
View v, int position, long id) {
        Intent nextScreen = new
Intent(getActivity().getApplicationContext()
, ClinicInfoActivity.class);
        nextScreen.putExtra("clinic_id",
MainActivity.clinicsList.get(position).get("
clinic_id"));

        startActivity(nextScreen);
    }
}

```

```

class GetClinicList extends
AsyncTask<String, String, String> {

    /**
     * Before starting background thread
     Show Progress Dialog
     * */

    @Override
    protected void onPreExecute() {
        super.onPreExecute();
        btnMapList2.setEnabled(false);

        progressDialog = new
ProgressDialog(getActivity());
        progressDialog.setMessage("Loading
clinics list...");
        progressDialog.setIndeterminate(false);
        progressDialog.setCancelable(false);
        progressDialog.show();
    }

    @Override
    protected String
doInBackground(String... args) {
        // Check for success tag
        return null;
    }

    /**
     * After completing background task,
     dismiss the progress dialog
     * @param file_url
     */
    protected void onPostExecute(String
file_url) {
        super.onPostExecute(file_url);
        // dismiss the dialog once
        product deleted
        progressDialog.dismiss();
        btnMapList2.setEnabled(true);

        ListAdapter adapter = new
SimpleAdapter(getActivity(),
MainActivity.clinicsList,
R.layout.fragment_list_nearest_clinics_item,
new
String[]{"clinic_name", "address", },
new
int[]{R.id.txtListCName,
R.id.txtListCAddress});

        setListAdapter(adapter);
    }
}

```

Listing 43: PlacesAutoCompleteAdapter.java

```

package ph.edu.upm.cjgaleng.spotavet;

import android.content.Context;
import android.view.LayoutInflater;
import android.view.View;

```

```

import android.view.ViewGroup;
import android.widget.ArrayAdapter;
import android.widget.Filter;
import android.widget.Filterable;
import android.widget.ImageView;
import android.widget.TextView;

import java.util.ArrayList;

/**
 * Created by CJ on 6/4/2015.
 */
public class PlacesAutoCompleteAdapter
extends ArrayAdapter<String> implements
Filterable {
    ArrayList<String> resultList;

    Context mContext;
    int mResource;

    GPlaceAPI mPlaceAPI = new GPlaceAPI();

    public PlacesAutoCompleteAdapter(Context
context, int resource) {
        super(context, resource);

        mContext = context;
        mResource = resource;
    }

    @Override
    public int getCount() {
        // Last item will be the footer
        return resultList.size();
    }

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

    @Override
    public Filter getFilter() {
        Filter filter = new Filter() {
            @Override
            protected FilterResults
performFiltering(CharSequence constraint) {
                FilterResults filterResults
= new FilterResults();

                return filterResults;
            }

            @Override
            protected void
publishResults(CharSequence constraint,
FilterResults results) {
            }
        };

        return filter;
    }

    @Override
    public View getView(int position, View
convertView, ViewGroup parent) {
        View view;

        LayoutInflater inflater =
(LayoutInflater)

```

```

mContext.getSystemService(Context.LAYOUT_INFLATER_SERVICE);

        if (position != (resultList.size() - 1))
            view =
inflater.inflate(R.layout.autocomplete_list_item, null);
            else
                view =
inflater.inflate(R.layout.autocomplete_google_logo, null);

        if (position != (resultList.size() - 1)) {
            TextView autoCompleteTextView =
(TextView)
view.findViewById(R.id.autocompleteText);

autoCompleteTextView.setText(resultList.get(position));
        }
        else {
            ImageView imageView =
(ImageView)
view.findViewById(R.id.imageView);
        }

        return view;
    }
}

```

Listing 44: SearchMapActivity.java

```

package ph.edu.upm.cjgaleng.spotavet;

import android.os.Bundle;
import android.os.Handler;
import android.os.HandlerThread;
import android.os.Looper;
import android.os.Message;
import
android.support.v7.app.AppCompatActivity;
import android.text.Editable;
import android.text.TextWatcher;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.AdapterView;
import
android.widget.AdapterView.OnItemClickListener;
import android.widget.AutoCompleteTextView;
import android.widget.Toast;

import java.util.ArrayList;

public class SearchMapActivity extends
AppCompatActivity implements
OnItemClickListener, TextWatcher {
    private static String TAG =
SearchMapActivity.class.getSimpleName();
    private PlacesAutoCompleteAdapter
mAdapter;
    private String place_id1;

    HandlerThread mHandlerThread;

```

```

        Handler mThreadHandler;

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

setContentView(R.layout.activity_search_map);
        }

        mAdapter = new
PlacesAutoCompleteAdapter(this,
R.layout.autocomplete_list_item);
        mAdapter.setNotifyOnChange(false);

        if (mThreadHandler == null) {
            // Initialize and start the
HandlerThread
            // which is basically a Thread
with a Looper
            // attached (hence a
MessageQueue)
            mHandlerThread = new
HandlerThread(TAG,
android.os.Process.THREAD_PRIORITY_BACKGROUND);
            mHandlerThread.start();

            // Initialize the Handler
mThreadHandler = new
Handler(mHandlerThread.getLooper()) {
                @Override
                public void
handleMessage(Message msg) {
                    if (msg.what == 1) {
                        ArrayList<String>
results = mAdapter.resultList;

                            if (results != null
&& results.size() > 0) {

mAdapter.notifyDataSetChanged();
                            }
                            else {

mAdapter.notifyDataSetChanged();
                            }
                        }
                    }
                };

            autoCompleteTextView
autoCompleteView = (AutoCompleteTextView)
findViewById(R.id.autocomplete);

autoCompleteView.addTextChangedListener(this);

autoCompleteView.setOnItemClickListener(this);
            autoCompleteView.setThreshold(3);

autoCompleteView.setAdapter(mAdapter);
        }

        @Override
        public void
beforeTextChanged(CharSequence s, int start,
int count, int after) {

```

```

    }

    @Override
    public void onTextChanged(CharSequence s, int start, int before, int count) {
        final String value = s.toString();

        // Remove all callbacks and messages
        mHandler.removeCallbacksAndMessages(null);

        // Now add a new one
        mHandler.postDelayed(new Runnable() {

            @Override
            public void run() {
                // Background thread

                mAdapter.resultList =
                mAdapter.mPlaceAPI.autocomplete(value);
                place_id1 =
                mAdapter.mPlaceAPI.place_id;

                // Footer
                if
                (mAdapter.resultList.size() > 0)
                mAdapter.resultList.add("footer");

                // Post to Main Thread
                mHandler.sendEmptyMessage(1);
            }
            }, 1000);
        }

        @Override
        public void afterTextChanged(Editable s)
        {

        }

        @Override
        public boolean onCreateOptionsMenu(Menu menu) {
            // Inflate the menu; this adds items
            to the action bar if it is present.

            getMenuInflater().inflate(R.menu.menu_search_map, menu);
            return true;
        }

        @Override
        public boolean onOptionsItemSelected(MenuItem item) {
            // Handle action bar item clicks
            here. The action bar will
            // automatically handle clicks on
            the Home/Up button, so long
            // as you specify a parent activity
            in AndroidManifest.xml.
            int id = item.getItemId();

            //noinspection SimplifiableIfStatement
            if (id == R.id.action_settings) {
                return true;
            }
        }
    }
}

```

```

    }

    return
    super.onOptionsItemSelected(item);
}

@Override
public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
    // Get data associated with the
    specified position
    // in the list (AdapterView)
    String description = (String)
    parent.getItemAtPosition(position);

    Toast.makeText(getApplicationContext(),
    place_id1, Toast.LENGTH_SHORT).show();
}

@Override
public void onDestroy() {
    super.onDestroy();

    // Get rid of our Place API Handlers
    if(mHandler != null) {
        mHandler.removeCallbacksAndMessages(null);
        mHandlerThread.quit();
    }
}
}

```

Listing 45: SearchMapActivityFragment.java

```

package ph.edu.upm.cjgaleng.spotavet;

import android.support.v4.app.Fragment;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;

/**
 * A placeholder fragment containing a
 * simple view.
 */
public class SearchMapActivityFragment
extends Fragment {

    public SearchMapActivityFragment() {
    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
    Bundle savedInstanceState) {
    }
}

```



```

        return
inflater.inflate(R.layout.fragment_search_ma
p, container, false);
    }
}

```

```

        return true;
    }

    return
super.onOptionsItemSelected(item);
}
}

```

Listing 46: VetInfoActivity.java

```

package ph.edu.upm.cjgaleng.spotavet;

import android.app.FragmentManager;
import android.app.FragmentTransaction;
import
android.support.v7.app.ActionBarActivity;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;

public class VetInfoActivity extends
ActionBarActivity {

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

setContentView(R.layout.activity_vet_info);

        VetInfoFragment vetInfoFragment =
new VetInfoFragment();

        FragmentManager fm =
getFragmentManager();
        FragmentTransaction ft =
fm.beginTransaction();
        ft.add(R.id.layoutVetActivity,
vetInfoFragment);
        ft.commit();

    }

    @Override
    public boolean onCreateOptionsMenu(Menu
menu) {
        // Inflate the menu; this adds items
to the action bar if it is present.

getMenuInflater().inflate(R.menu.menu_vet_in
fo, menu);
        return true;
    }

    @Override
    public boolean
onOptionsItemSelected(MenuItem item) {
        // Handle action bar item clicks
here. The action bar will
        // automatically handle clicks on
the Home/Up button, so long
        // as you specify a parent activity
in AndroidManifest.xml.
        int id = item.getItemId();

        //noinspection
SimplifiableIfStatement
        if (id == R.id.action_settings) {

```

Listing 47: VetInfoFragment.java

```

package ph.edu.upm.cjgaleng.spotavet;

import android.app.Activity;
import android.app.ProgressDialog;
import android.net.Uri;
import android.os.AsyncTask;
import android.os.Bundle;
import android.app.Fragment;
import android.text.Html;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ListAdapter;
import android.widget.ListView;
import android.widget.SimpleAdapter;
import android.widget.TextView;

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

import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;

public class VetInfoFragment extends
Fragment {

    // change to server IP address
    private static final String LOGIN_URL =
"http://192.168.43.149:80/SpotAVet/resources
/json/vet-info.php";
    TextView txtCardVName, txtCardVLicense,
txtCardVEmail, txtCardVContact,
        txtCardVSpecialization,
txtCardVSchedule;

    private ListView listViewVets;
    ArrayList<HashMap<String, String>>
vetsList;

    private ProgressDialog pDialog;

    // JSON parser class
    JSONParser jsonParser = new
JSONParser();

    public VetInfoFragment() {
        // Required empty public constructor
    }

    @Override

```

```

    public View onCreateView(LayoutInflater inflater, ViewGroup container,
        Bundle savedInstanceState) {
        // Inflate the layout for this fragment

        // -- inflate the layout for this fragment
        View myInflatedView =
            inflater.inflate(R.layout.fragment_vetinfo,
                container, false);

        // Set the Text to try this out
        txtCardVName = (TextView)
            myInflatedView.findViewById(R.id.txtCardVName);
        txtCardVLicense = (TextView)
            myInflatedView.findViewById(R.id.txtCardVLicense);
        txtCardVEmail = (TextView)
            myInflatedView.findViewById(R.id.txtCardVEmail);
        txtCardVContact = (TextView)
            myInflatedView.findViewById(R.id.txtCardVContact);
        txtCardVSpecialization = (TextView)
            myInflatedView.findViewById(R.id.txtCardVSpecialization);
        txtCardVSchedule = (TextView)
            myInflatedView.findViewById(R.id.txtCardVSchedule);

        txtCardVName.setText(getActivity().getIntent()
            .getExtras().getString("name"));
        String license_no =
            getActivity().getIntent().getExtras().getString("license_no");

        txtCardVLicense.setText(Html.fromHtml("<strong>License No. : </strong>&nbsp;" +
            license_no));
        String email =
            getActivity().getIntent().getExtras().getString("email");

        txtCardVEmail.setText(Html.fromHtml("<strong>Email : </strong>&nbsp;" + email));
        String contact_no =
            getActivity().getIntent().getExtras().getString("contact_no");

        txtCardVContact.setText(Html.fromHtml("<strong>Contact No. : </strong>&nbsp;" +
            contact_no));

        return myInflatedView;
    }

    public void onResume() {
        // TODO Auto-generated method stub
        super.onResume();

        //loading the comments via AsyncTask
        new GetVetInfo().execute();
    }

```

```

class GetVetInfo extends
    AsyncTask<String, String, String> {
    private String specialization = "";
    private String schedule = "";

    /**
     * Before starting background thread
     * Show Progress Dialog
     */
    @Override
    protected void onPreExecute() {
        super.onPreExecute();
        pDialog = new
            ProgressDialog(getActivity());
        pDialog.setMessage("Loading vet
            info...");
        pDialog.setIndeterminate(false);
        pDialog.setCancelable(false);
        pDialog.show();
    }

    @Override
    protected String
    doInBackground(String... args) {
        // Check for success tag
        int success;

        try {
            // Building Parameters
            List<NameValuePair> params =
                new ArrayList<NameValuePair>();
            params.add(new
                BasicNameValuePair("clinic_id",
                    getActivity().getIntent().getExtras().getString("clinic_id")));
            params.add(new
                BasicNameValuePair("vet_id",
                    getActivity().getIntent().getExtras().getString("vet_id")));

            Log.d("request!",
                "starting");

            // getting product details
            by making HTTP request
            JSONObject json =
                jsonParser.makeHttpRequest(
                    LOGIN_URL, "POST",
                    params);

            // check your log for json
            response
                Log.d("Get vetinfo attempt",
                    json.toString());

            // json success tag
            success =
                json.getInt("success");
            if (success == 1) {
                specialization =
                    json.getString("specializations");
                schedule =
                    json.getString("schedule");

                return
                    json.getString("message");
            } else {

```

```

        Log.d("Vet not found!",
json.getString("message"));
        return
json.getString("message");
    }
    } catch (JSONException e) {
        e.printStackTrace();
    }

    return null;
}

/**
 * After completing background task,
dismiss the progress dialog
 * @param file_url
 */
protected void onPostExecute(String
file_url) {
    super.onPostExecute(file_url);
    // dismiss the dialog once
product deleted
    pDialog.dismiss();

txtCardVSpecialization.setText(specializatio
n);

txtCardVSchedule.setText(Html.fromHtml(sched
ule));
}
}
}

```

Listing 48: ListVetsFragment.java

```

package ph.edu.upm.cjgaleng.spotavet;

import android.app.Activity;
import android.app.FragmentTransaction;
import android.app.ProgressDialog;
import android.content.Intent;
import android.os.AsyncTask;
import android.os.Bundle;
import android.app.ListFragment;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ArrayAdapter;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.SimpleAdapter;
import android.widget.Toast;

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

import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;

```

```

public class ListVetsFragment extends
ListFragment {

    // change to server IP address
    private static final String LOGIN_URL =
"http://192.168.43.149:80/SpotAVet/resources
/json/clinic-vets.php";
    private ListView listViewVets;
    ArrayList<HashMap<String, String>>
vetsList;

    private ProgressDialog pDialog;

    // JSON parser class
    JSONParser jsonParser = new
JSONParser();

    /**
 * Mandatory empty constructor for the
fragment manager to instantiate the
 * fragment (e.g. upon screen
orientation changes).
 */
    public ListVetsFragment() {
    }

    @Override
    public View onCreateView(LayoutInflater
inflater, ViewGroup container, Bundle
savedInstanceState) {
        return
inflater.inflate(R.layout.fragment_listvets,
container, false);
    }

    @Override
    public void onActivityCreated(Bundle
savedState) {
        super.onActivityCreated(savedState);

registerContextMenu(getListView());
    }

    public void onResume() {
        // TODO Auto-generated method stub
super.onResume();

        vetsList = new
ArrayList<HashMap<String, String>>();
        //loading the comments via AsyncTask
new GetVets().execute();
    }

    @Override
    public void onItemClick(ListView l,
View v, int position, long id) {
        VetInfoFragment vetInfoFragment =
new VetInfoFragment();

        Intent intent = new
Intent(getActivity().getApplicationContext()
, VetInfoActivity.class);

        intent.putExtra("name",
vetsList.get(position).get("name"));
        intent.putExtra("license_no",
vetsList.get(position).get("license_no"));
        intent.putExtra("email",
vetsList.get(position).get("email"));
    }
}

```

```

        intent.putExtra("contact_no",
vetsList.get(position).get("contact_no"));
        intent.putExtra("vet_id",
vetsList.get(position).get("vet_id"));
        intent.putExtra("clinic_id",
getActivity().getIntent().getExtras().getString("clinic_id"));

        startActivity(intent);
    }

    class GetVets extends AsyncTask<String,
String, String> {

        /**
         * Before starting background thread
         Show Progress Dialog
         */
        boolean failure = false;
        private String loginRole;

        @Override
        protected void onPreExecute() {
            super.onPreExecute();
            progressDialog = new
ProgressDialog(getActivity());
            progressDialog.setMessage("Loading
users list...");
            progressDialog.setIndeterminate(false);
            progressDialog.setCancelable(false);
            progressDialog.show();
        }

        @Override
        protected String
doInBackground(String... args) {
            // Check for success tag
            int success;

            JSONArray vetsArray = null;

            try {

                // Building Parameters
                List<NameValuePair> params =
new ArrayList<NameValuePair>();
                params.add(new
BasicNameValuePair("clinic_id",

getActivity().getIntent().getExtras().getString("clinic_id")));

                Log.d("request!",
"starting");

                // getting product details
                by making HTTP request
                JSONObject json =
jsonParser.makeHttpRequest(
                LOGIN_URL, "POST",
                params);

                if
                (json.has("veterinarians")) {
                    vetsArray =
                    json.getJSONArray("veterinarians");

                    // check your log for
                    json response

```

```

                Log.d("Get vets
attempt", json.toString());

                // json success tag
                success =
                json.getInt("success");
                if (success == 1) {
                    // looping through
                    All messages
                    for (int i = 0; i <
                    vetsArray.length(); i++) {
                        JSONObject c =
                        vetsArray.getJSONObject(i);

                        // Storing each
                        json item in variable
                        String vet_id =
                        c.getString("vet_id");
                        String
                        license_no = c.getString("license_no");
                        String name =
                        c.getString("first_name") + " " +
                        c.getString("last_name");
                        String email =
                        c.getString("email");
                        String
                        contact_no = c.getString("contact_no");

                        // creating new
                        HashMap
                        HashMap<String,
                        String> map = new HashMap<String, String>();

                        // adding each
                        child node to HashMap key => value
                        map.put("vet_id", vet_id);

                        map.put("license_no", license_no);
                        map.put("name",
                        name);
                        map.put("email",
                        email);

                        map.put("contact_no", contact_no);

                        // adding
                        HashMap to ArrayList
                        vetsList.add(map);
                    }

                    return
                    json.getString("message");
                } else {
                    Log.d("No result!",
                    json.getString("message"));
                    return
                    json.getString("message");
                }
            } catch (JSONException e) {
                e.printStackTrace();
            }

            return null;
        }

        /**

```

```

        * After completing background task,
dismiss the progress dialog
        * @param file_url
        */
protected void onPostExecute(String
file_url) {
    // dismiss the dialog once
product deleted
    pDialog.dismiss();

    ListAdapter adapter = new
SimpleAdapter(getActivity(), vetsList,
R.layout.fragment_listvets_item,
                new String[]{"name",
"email", "license_no", },
                new int[]{R.id.txtVName,
R.id.txtVEmail, R.id.txtVLicense});

    setListAdapter(adapter);
}
}
}

```

Listing 49: res/layout/activity_clinic_info.xml

```

<LinearLayout
xmlns:android="http://schemas.android.com/ap
k/res/android"

xmlns:tools="http://schemas.android.com/tool
s" android:layout_width="match_parent"
    android:layout_height="wrap_content"
android:paddingLeft="@dimen/activity_hORIZON
tal_margin"

android:paddingRight="@dimen/activity_hORIZON
tal_margin"

android:paddingTop="@dimen/activity_VERTICAL
_margin"

android:paddingBottom="@dimen/activity_VERTI
cal_margin"
    android:orientation="vertical"

tools:context="ph.edu.upm.cjgaleng.ClinicInf
oActivity">

    <android.support.v7.widget.CardView

xmlns:card_view="http://schemas.android.com/
apk/res-auto"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        card_view:contentPadding="8dp"
        card_view:cardElevation="2dp"
        card_view:cardCornerRadius="5dp" >

        <LinearLayout

android:layout_width="match_parent"

android:layout_height="wrap_content"
            android:orientation="vertical">

```

```

        <TextView

style="@style/Base.TextAppearance.AppCompat.
Headline"

        android:layout_width="match_parent"

        android:layout_height="wrap_content"

        android:textAppearance="?android:attr/textAp
pearanceLarge"
            android:text="Clinic
Information"
            android:id="@+id/textView2"

        android:background="#ffcc5d00" />

        <LinearLayout

android:layout_width="fill_parent"

android:layout_height="wrap_content"

        android:orientation="horizontal" >

            <TextView

                android:layout_width="0dp"

                android:layout_height="wrap_content"

                android:layout_weight="0.2"
                    android:text="Clinic
Name :"

                android:layout_gravity="center" />
            <EditText

                style="@style/Base.TextAppearance.AppCompat.
Subhead"

                android:layout_width="0dp"

                android:layout_height="wrap_content"

                android:layout_weight="0.8"

                android:id="@+id/txtCName"
                    android:editable="false"

                android:clickable="false"

                android:focusable="false" />
            </LinearLayout>

            <LinearLayout

                android:layout_width="fill_parent"

                android:layout_height="wrap_content"

                android:orientation="horizontal"

                android:layout_gravity="center">

                <TextView

                    android:layout_width="0dp"

                    android:layout_height="wrap_content"

```

```

android:layout_weight="0.2"
        android:text="Address :
" />
        <EditText
style="@style/Base.TextAppearance.AppCompat.
Subhead"
android:layout_width="0dp"
android:layout_height="wrap_content"
android:layout_weight="0.8"
android:id="@+id/txtCAddress"
        android:editable="false"
android:clickable="false"
android:focusable="false" />
        </LinearLayout>
        <LinearLayout
android:layout_width="fill_parent"
android:layout_height="wrap_content"
android:orientation="horizontal" >
                <TextView
android:layout_width="0dp"
android:layout_height="wrap_content"
android:layout_weight="0.2"
        android:text="Schedule :
" />
        <EditText
style="@style/Base.TextAppearance.AppCompat.
Subhead"
android:layout_width="0dp"
android:layout_height="wrap_content"
android:layout_weight="0.8"
android:id="@+id/txtCSchedule"
        android:editable="false"
android:clickable="false"
android:focusable="false" />
        </LinearLayout>
        <LinearLayout
android:layout_width="fill_parent"
android:layout_height="wrap_content"
android:orientation="horizontal" >
                <TextView
android:layout_width="0dp"

```

```

android:layout_height="wrap_content"
android:layout_weight="0.2"
        android:text="Contact #
:" />
        <EditText
style="@style/Base.TextAppearance.AppCompat.
Subhead"
android:layout_width="0dp"
android:layout_height="wrap_content"
android:layout_weight="0.8"
android:id="@+id/txtCContactNo"
        android:editable="false"
android:clickable="false"
android:focusable="false" />
        </LinearLayout>
        <LinearLayout
android:layout_width="fill_parent"
android:layout_height="wrap_content"
android:orientation="horizontal" >
                <TextView
android:layout_width="0dp"
android:layout_height="wrap_content"
android:layout_weight="0.2"
        android:text="Email : "
/>
        <EditText
style="@style/Base.TextAppearance.AppCompat.
Subhead"
android:layout_width="0dp"
android:layout_height="wrap_content"
android:layout_weight="0.8"
android:id="@+id/txtCEmail"
        android:editable="false"
android:clickable="false"
android:focusable="false" />
        </LinearLayout>
        </android.support.v7.widget.CardView>
        <android.support.v7.widget.CardView
xmlns:card_view="http://schemas.android.com/
apk/res-auto"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        card_view:contentPadding="8dp"
        card_view:cardElevation="2dp"

```

```

        card_view:cardCornerRadius="5dp" >
<LinearLayout
    android:id="@+id/linearLayout1"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal">

</LinearLayout>

</android.support.v7.widget.CardView>
</LinearLayout>

```

Listing 50: res/layout/activity_list_nearest_clinics.xml

```

<FrameLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">
<LinearLayout
    android:orientation="horizontal"
    android:layout_width="fill_parent"
    android:layout_height="465dp"

android:id="@+id/layoutNearestClinicsList">
</LinearLayout>

<LinearLayout
    android:orientation="horizontal"
    android:layout_width="fill_parent"
    android:layout_height="45dip"
    android:weightSum="2"

android:layout_gravity="center_horizontal|bottom">
<Button
    android:layout_width="0dip"
    android:layout_height="fill_parent"
    android:layout_weight="1"
    android:text="Map"
    android:id="@+id/btnMapList2"

android:background="@drawable/buttonshape"
    android:enabled="false" />
<Button
    android:layout_width="0dip"
    android:layout_height="fill_parent"
    android:layout_weight="1"
    android:text="List"
    android:id="@+id/btnFilter2"

android:background="@drawable/buttonshape"
    android:enabled="false" />
</LinearLayout>
</FrameLayout>

```

Listing 51: res/layout/activity_main.xml

```

<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:tools="http://schemas.android.com/tools"

xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity" >

<fragment
xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:map="http://schemas.android.com/apk/res-auto"

xmlns:tools="http://schemas.android.com/tools" tools:context=".MainActivity"
    android:id="@+id/map"
    android:layout_width="match_parent"
    android:layout_height="match_parent"

android:name="com.google.android.gms.maps.MapFragment"/>

<ImageButton
    android:id="@+id/btnShowLocation"
    android:src="@drawable/location"
    android:layout_width="60dp"
    android:layout_height="60dp"
    android:scaleType="fitCenter"

android:layout_above="@+id/linearLayout"

android:layout_toRightOf="@+id/btnChangeAddress"

android:layout_toEndOf="@+id/btnChangeAddress" />

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"

android:textAppearance="?android:attr/textAppearanceSmall"
    android:text="Latitude : x,
Longitude : y"
    android:id="@+id/textView_latLng"
    android:gravity="center|top"
    android:typeface="sans"

android:layout_toLeftOf="@+id/btnSearchClinic"
    android:layout_alignParentTop="true"

android:layout_alignParentLeft="false"

android:layout_alignParentStart="true"
    android:textColor="#ffcc5d00"
    android:layout_marginTop="10dp"
    android:textSize="11sp" />

```

```

        <Button
            android:id="@+id/btnChangeAddress"
            android:layout_width="wrap_content"
            android:layout_height="60dp"
            android:layout_alignParentTop="true"
android:layout_toLeftOf="@+id/btnSearchClinic"
android:layout_toStartOf="@+id/btnSearchClinic"
android:layout_alignParentStart="true"
android:layout_alignParentLeft="true"
android:background="@drawable/buttonshape"
            android:textAlignment="center" />
        <ImageButton
            android:layout_width="60dp"
            android:layout_height="60dp"
            android:src="@drawable/search2"
            android:id="@+id/btnSearchClinic"
            android:scaleType="fitCenter"
android:layout_alignParentRight="true"
            android:layout_alignParentEnd="true"
            android:layout_alignParentTop="true"
/>
        <LinearLayout
            android:orientation="horizontal"
            android:layout_width="fill_parent"
            android:layout_height="45dip"
            android:weightSum="2"
            android:layout_gravity="bottom"
android:layout_alignParentBottom="true"
android:layout_alignParentLeft="false"
android:layout_alignParentStart="true"
            android:id="@+id/linearLayout"
android:layout_alignParentRight="false"
android:layout_alignWithParentIfMissing="false">
            <Button
                android:layout_width="0dip"
android:layout_height="fill_parent"
                android:layout_weight="1"
                android:text="Map"
                android:id="@+id/btnMapList"
android:background="@drawable/buttonshape"
                android:enabled="false"
                android:textColor="#ff000000" />
            <Button
                android:layout_width="0dip"
android:layout_height="fill_parent"
                android:layout_weight="1"
                android:text="List"
                android:id="@+id/btnFilter"
android:background="@drawable/buttonshape"
                android:enabled="false"

```

```

                android:textColor="#ff000000" />
        </LinearLayout>
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
android:textAppearance="?android:attr/textAppearanceMedium"
            android:text="This is my address"
            android:id="@+id/textView_address"
android:layout_below="@+id/textView_latLng"
android:layout_alignParentLeft="true"
android:layout_alignParentStart="true"
android:layout_toLeftOf="@+id/btnSearchClinic"
android:layout_toStartOf="@+id/btnSearchClinic"
            android:typeface="serif"
            android:textAlignment="center"
            android:maxLines="1"
            android:gravity="center|bottom"
            android:textStyle="bold"
android:layout_alignBottom="@+id/btnChangeAddress"
            android:layout_marginBottom="8dp"
            android:textSize="16sp"
            android:ellipsize="end" />
</RelativeLayout>

```

Listing 52: res/layout/activity_search_map.xml

```

<fragment
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools" android:id="@+id/fragment"
    android:name="ph.edu.upm.cjgaleng.spotavet.SearchMapActivityFragment"
    tools:layout="@layout/fragment_search_map"
    android:layout_width="match_parent"
    android:layout_height="match_parent" />

```

Listing 53: res/layout/ activity_vet_info.xml

```

<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:id="@+id/layoutVetActivity"

```



```
tools:context="ph.edu.upm.cjgaleng.VetInfoActivity">
</RelativeLayout>
```

Listing 54:
res/layout/autocomplete_google_logo.xml

```
<ImageView
xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/imageView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:src="@drawable/google" />
```

Listing 55:
res/layout/autocomplete_list_item.xml

```
<?xml version="1.0" encoding="utf-8"?>
<TextView
xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:padding="10dp"
    android:id="@+id/autocompleteText" />
```

Listing 56:
res/layout/fragment_list_nearest_clinics.xml

```
<LinearLayout
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:orientation="vertical"

tools:context="ph.edu.upm.cjgaleng.spotavet.ListNearestClinicsActivityFragment">

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"

        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="Nearest clinics : "
        android:id="@+id/textView"
        android:background="#ffcc5d00" />

    <ListView
        android:id="@android:id/list"
```

```
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
    />
</LinearLayout>
```

Listing 57:
res/layout/fragment_list_nearest_clinics_item.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="fill_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical" >

    <TextView
        android:id="@+id/txtListCName"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:paddingTop="8dip"
        android:paddingLeft="8dip"
        android:paddingBottom="4dip"
        android:textSize="24dip"
        android:textStyle="bold" />

    <TextView
        android:id="@+id/txtListCAddress"
        android:layout_height="wrap_content"
        android:layout_width="wrap_content"
        android:paddingLeft="8dip"
        android:paddingBottom="6dip"
        android:textSize="16dip"

        android:layout_below="@id/txtListCName"/>
</RelativeLayout>
```

Listing 58:
res/layout/fragment_listvets.xml

```
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical"

tools:context="ph.edu.upm.cjgaleng.ListVetsFragment">

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
```

```

android:textAppearance="?android:attr/textAppearanceLarge"
    android:text="Veterinarians : "
    android:id="@+id/textView"
    android:background="#ffcc5d00" />

    <ListView
        android:id="@android:id/list"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
    />
</LinearLayout>

```

Listing 59: res/layout/fragment_listvets_item.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="fill_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical" >

    <TextView
        android:id="@+id/txtVName"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:paddingTop="8dip"
        android:paddingLeft="8dip"
        android:paddingBottom="4dip"
        android:textSize="20dip"
        android:textStyle="bold" />

    <TextView android:id="@+id/txtVEmail"
        android:layout_height="wrap_content"
        android:layout_width="wrap_content"
        android:paddingLeft="8dip"
        android:paddingBottom="6dip"
        android:textSize="15dip"

    android:layout_below="@id/txtVName"/>

    <TextView android:id="@+id/txtVLicense"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"

    android:layout_alignParentRight="true"
        android:padding="8dip"/>/

</RelativeLayout>

```

Listing 60: res/layout/fragment_search_map.xml

```

<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:paddingLeft="@dimen/activity_horizontal_margin"

    android:paddingRight="@dimen/activity_horizontal_margin"

    android:paddingTop="@dimen/activity_vertical_margin"

    android:paddingBottom="@dimen/activity_vertical_margin"

    tools:context="ph.edu.upm.cjgaleng.spotavet.SearchMapActivityFragment">

```

```

    <!-- City and Country Selector -->
    <AutoCompleteTextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/autocomplete"
        android:hint="Type in your Location"
    />

    <Button

    android:id="@+id/clearable_button_clear"
        android:layout_width="30dip"
        android:layout_height="30dip"

    android:background="@android:drawable/ic_menu_close_clear_cancel"

    android:layout_alignBottom="@+id/autocomplete"

    android:layout_alignRight="@+id/autocomplete"

    android:layout_alignEnd="@+id/autocomplete"
        android:layout_marginRight="11dp"
        android:layout_marginBottom="5dp"
        android:layout_alignParentTop="true"

    android:layout_alignParentLeft="false"

    android:layout_alignParentBottom="false"

    android:layout_alignParentRight="true" />
</RelativeLayout>

```

Listing 61: res/layout/fragment_vetinfo.xml

```

<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"

    xmlns:tools="http://schemas.android.com/tools"

    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:paddingLeft="8dp"

```



```
        android:layout_height="wrap_content"
            android:layout_marginTop="8dp"
            android:text="Schedule :"
            android:textStyle="bold" />
        <TextView
            style="@style/Base.TextAppearance.AppCompat.
            Body1"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:layout_marginLeft="10dp"
            android:id="@+id/txtCardVSchedule"
            android:text="Schedule" />
    </LinearLayout>
</android.support.v7.widget.CardView>
</LinearLayout>
```

XI. Acknowledgement

Yehey! It's my turn now to say "Finally"! Yehey dahil siyempre FINALLY tapos ko na lahat ng comsci subjects ko, konti na lang graduation na weee.. after many years. Also happy ako kasi freestyle ang page na to, ubos na ubos na kasi English ko from the previous pages. ☺
My Special Problem is indeed an accomplishment for me. Para sa tulad ko na hindi magaling sa write-ups, natapos ko ito.. So happy.

Firstly, I would like to give my big thanks to my adviser Ma'am Sheila, sobrang thank you po sa paulit ulit niyong pagtanggap sa akin, kahit minsan isang taon bago ako magpakita ulit sa inyo. Hehe. Isa po kayo sa nakapagpa push sa akin na tapusin ito. Yung mga words niyo po nakakatulong, though mahina pa rin po talaga ang loob ko sa ganito. Ayuuun tapos na po ang lahat ng '08 advisees niyo ☺

Sa blockmates ko, salamat sa friendship and everything, nung undergrad years natin or should I say nung may klase pa tayo. Naenjoy ko ang college life dahil sa inyo. ♥ I admit na hindi ako masyadong nagpapakita after our supposedly graduation year. Miss na miss ko na kayong lahat. Pagkatapos nito, sana naaalala niyo pa din ako. Dahil... sasama na ko sa mga outings, YAY. Tagal ko na naiinggit gusto ko talaga sumama ☺ Thank you mommy JJ, Auradee lobster, sa paulit ulit na pag-aya sa akin kahit ang madalas kong sagot ay no. Tina, Hii Rialabs!!!! Kayo yung mga paulit ulit nangangamusta sakin kahit di ako nakakapagreply. See you soon. Eulahhhpot, uwi ka na ulit dito sa Pinas. Hehe

Ate Naji, JM, Gela, Patrick, Luwi, Ian, magkikita tayo after this.. I insist. I need to catch up with you guys. Pero sagot niyo muna food and "tea?", bigtime na kayo e :D

Keisha, Lorielyn, Axl, thank you din sa pag motivate sa aking tapusin to, I will meet you all soon. ☺

Mica, piggie frieeend, thank you ng marami sa morale support, food support, words support sa pagsusulat ng papers, haha lahat na. You've been there for me from the start of my final topic (haha) and never left me, kahit ilang breakdown na at pagpapanic na ang nangyari sa akin. Your friendship means so much to me. Ililibre kita after this, woo.

And last, but not the least, to my dearest family..

Gillian, (yehess naman special mention) thank you sa pagpapahiram mo ng laptop, sobrang tagal na din pala to nasakin. At kita mo na, pati yung laptop ang lawak na ng ngiti for me. Ipapaayos ko na lang to pag kaya pang ayusin. Goodluck sa remaining years mo sa BS Nursing. Mag grad ka ng regular ah, sumablay ka sa 2017, wag mo ko tularan hehe. Make us proud! Love you, Gillian :P

Mama, Tatay, eto na po, pagkatapos ng ilang taon niyong pangungulit, this is for you. Kahit nagtampo po ako nung una kasi pinagresign niyo ako para tapusin ang SP, ehh gusto ko pa nga magwork. Pero narealize ko naman para sakin din talaga yun, para magkaroon ako ng diploma. Salamat po sa lahat, sa pagpapaaral, sa allowances, sa trust. Mama, sorry kung wala ako masyado time makipag-usap sa bahay lately. Pero thank you sa paglalaba ng ilan kong gamit lately, dahil wala na akong time maglaba. I love you mama. ♥ Be strong, and always be healthy, Mama. Masusuot mo na yung mga binili mong damit, mama donya. Advanced happy birthday gift ko muna sa'yo to. Next time na iphone okay? Heehhe

Tay, I think this is the best gift I could give you on Father's day. Lolol. Eto sana yung birthday gift ko sa'yo.. kaso naudlot. Hehe ang mahalaga naman po diba tapos na? Diba po? Hehe. Finally eto na po nakapagpatapos na kayo. Eto na po yung dream niyo na makagraduate. I love you, tatay. ♥