**VetPet**

**Software Design**

**CSCI-P465/565 (Software Engineering I)**

**Project Team**

**Amey Avinash Tarfe**

**Christian Gonzalez**

**Jainendra Kumar**

**Xinran Dai**



# 1. Introduction

This section introduces the design approach to the software system.

## 1.1 System Description

VetPet intends to streamline the treatment of animals by local veterinary doctors through a centralized web portal which acts as a medium of communication between the two parties. In addition, the system will support insurance plans for the animals by connecting the pet owners with multiple insurance plan providers. The system will be unique in a way that very few existing systems currently exists in the country which centrally govern the treatment of animals.

## 1.2 Design Evolution

This section is intended to document the rationale behind the selected design solution.

**1.2.1 Design Issues**

Application only needs a Web browser to be accessed.

### 1.2.2 Candidate Design Solutions

We have decided to build most of the components in the system using JSP as the backend. We are looking to deploy the application on the Amazon Web Services server.

### 1.2.3 Design Solution Rationale

Java Server Pages (**JSP**) is a server-side programming technology that enables the creation of dynamic, platform-independent method for building Web-based applications. **JSP** have access to the entire family of Java APIs, including the JDBC API to access enterprise databases.

**Amazon Web Services** offers reliable, scalable, and inexpensive cloud computing services. There are sources available in the aws.amazon.com to assist in hosting.

## 1.3 Design Approach

### 1.3.1 Methods

Until now we have designed the Login and User registration feature for which the approach used is that along with the normal registration and login process, we will encrypt the user’s password and store it in the DB. Also, there will be a Duo authentication for all the users every time they login. Forgot password feature will consist of two options to reset the password i.e. either using Security Question or Duo authentication.

For testing, we would be using unit and regression testing. Unit tests are for stability and code predictability, while regression testing is for evaluating of how users interact with the code.

### 1.3.2 Standards

We are adhering to the JSP standards and practices as mentioned in the official documentation.

Apart from these, we are using JavaScript and HTML5. Also, CSS3 and Bootstrap will be used to provide better look and feel of the GUI

### 1.3.3 Tools

* Editors – Eclipse, Sublime Text 2
* Webserver- Tomcat
* Database - MySQL
* Hosting- Amazon Web Services

# 2. System Architecture

## 2.1 System Design

At a higher level, our system will have classes/functions for each component implementing a specific feature. At a lower level, each function will have its own Logic and Unit Tests.

We have used a basic template for our HTML pages having the same header and footer. This template will be inherited for all the HTML pages that we develop.

## 2.2 External Interfaces

The system is using OAuth using Google API. The user can login using his/her Google credentials. For the Hospital locator, Google Maps API is being used.

**3. Component Design**

**Component Name** (1)

Login and registration

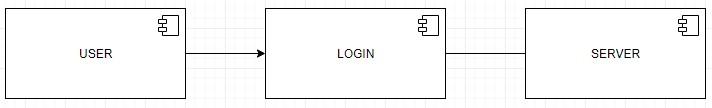
# Component Description

The Login form allows the users to login with the credentials. The users of the system are of three types viz. Patients, Doctors and the insurance providers. Accordingly, there are three different logins for the same. When providing the correct credentials, the user will be receiving Duo authentication mail containing a One Time Password (OTP). Then the user must authenticate for the second time using this OTP and once they are verified, they will be logged in to the website.

# Responsible Development Team Member

All the team members had divided this major component into smaller chunks and then integrated it together.

# Component Diagram



# Component Objects

User Component: A User who will be connect to the URL of the website and then will either register themselves if they will be new user or else, they will log in to the site if they are existing users.

Login Component: This component consists of multiple objects which are given below

Login – Authenticate the User based on provided credentials

Registration – Form to register new user along with the security questions.

Duo Authentication – Send an OTP to the user in email and authenticate once they have provided correct OTP.

Forgot Password – Forgot password has 2 options, Security question and OTP. Security Question will have 2 questions, if user answers them correctly, they will be able to reset the password. Or an OTP will be sent to their registered email, if they authenticate themselves using OTP, they will be able to reset their password.

Database Component: This component will store all the User’s data and credentials and will be used to authenticate the users based on the Database calls made to the server.

# Component Interfaces (internal and external)

The component will make calls to the Database to validate the credentials of users trying to access the system. The component will check with the database to validate the credentials.

# Component Error Handling

If a user is trying to login with invalid credentials such as Invalid username or Invalid password, they are handled in the component.

**Component Name** **(2)**

Search Insurance provider component

# Component Description

The Search Insurance provider component allows the users to look out for various insurance providers with various quotes available and choose the best among them. Also, there is a recommender system based on the reviews and statistics available. After the insurance package is chosen, the user have to complete the payment of the package through the payment gateway.

# Responsible Development Team Member

All the team members had divided this major component into smaller chunks and then integrated it together.

# Component Objects

User Component: A User who will be connect to the URL of the website and then will either register themselves if they will be new user or else, they will log in to the site if they are existing users.

Search Insurance provider component: This component consists of:

Insurance provider component – Select the specific insurance provider from the list of providers.

Insurance package component – Select the type of insurance package from the selected insurance provider based on the needs of the patient.

Recommender component – Return a list of insurance packages depending upon the statistics such as the amount charged by the provider, amount covered under the package, etc.

Payment component – Based on the insurance package bought, the total cost of the same is calculated and the user has to input the payment information such as card details and some basic information, and complete the payment.

Database Component: This component will store all the User’s data and credentials and will be used to authenticate the users based on the Database calls made to the server.

# Component Interfaces (internal and external)

The component will make calls to the Database to validate the credentials of users trying to access the system. The component will check with the database to validate the credentials.

# Component Error Handling

If a user is trying to input invalid credentials such as unknown packages or unknown providers, they are handled in the component.

**Component Name** **(3)**

Appointment component

# Component Description

The Appointment component allows the patients to search doctors according to the specialization, see the available appointment times for that particular doctor and schedule the appointment for the desired time. On the doctor side (view), the doctor can set the available dates for the appointments and see the list of already scheduled appointments.

# Responsible Development Team Member

All the team members had divided this major component into smaller chunks and then integrated it together.

# Component Objects

User Component: A User who will be connect to the URL of the website and then will either register themselves if they will be new user or else, they will log in to the site if they are existing users.

Appointment component: This component consists of:

Doctor component – Select the specific doctor according to the pet-type and the doctor specialization.

Date component – Select from the available dates. Doctor can set these dates.

Appointment component – Schedule the appointment for the date selected. The doctor can see the scheduled dates for all the patients.

Database Component: This component will store all the User’s data and credentials and will be used to authenticate the users based on the Database calls made to the server.

# Component Interfaces (internal and external)

The component will make calls to the Database to validate the credentials of users trying to access the system. The component will check with the database to validate the credentials.

# Component Error Handling

If a user is trying to input invalid dates such as unavailable dates, they are handled in the component.

**Component Name (4)**

Doctor information component

# Component Description

The Doctor information component allows the users/patients to look out for the information of various doctors such as Doctor name, address, Doctor specialization, star rating and reviews or feedback from previous patients.

# Responsible Development Team Member

Most of the part of this component was implemented by Jainendra Kumar and the UI part was implemented by Amey Tarfe.

# Component Objects

User Component: A User who will be connect to the URL of the website and then will either register themselves if they will be new user or else, they will log in to the site if they are existing users.

Doctor information component: This component consists of:

Doctor component – Display the doctor name, his/her picture and specialization.

Rating component - Display the star rating for the doctor according to previous reviews out of five.

Review component – Patient can add feedback to the previous appointment with a particular doctor.

Database Component: This component will store all the User’s data and credentials and will be used to authenticate the users based on the Database calls made to the server.

# Component Interfaces (internal and external)

The component will make calls to the Database to validate the credentials of users trying to access the system. The component will check with the database to validate the credentials.

# Component Error Handling

If a user is trying to rate the doctor which was different than doctor they were appointed to, they are handled in the component.

**Component Name (5)**

Chat component

# Component Description

The users (Patients, Doctors and the Insurance providers) can chat with each other LIVE when each of the user is online.

# Responsible Development Team Member

Most of the part of this component was implemented by Christian Gonzalez.

# Component Objects

User Component: A User who will be connect to the URL of the website and then will either register themselves if they will be new user or else, they will log in to the site if they are existing users.

Chat component: This component consists of:

Chatbox component – Display the online users and text message the online user one-on-one.

Database Component: This component will store all the User’s data and credentials and will be used to authenticate the users based on the Database calls made to the server.

# Component Interfaces (internal and external)

The component will make calls to the Database to validate the credentials of users trying to access the system. The component will check with the database to validate the credentials.

# Component Error Handling

If a user is trying to chat with someone who is offline, they are handled in the component.

# Revision History

|  |  |  |
| --- | --- | --- |
| **Revision** | **Date** | **Change Description** |
| Initial Version | 9/30/2018 |  |
| Version 2 | 10/14/2018 | Changes for the SPRINT2. |
| Version 3 | 10/28/2018 | Changes for the SPRINT3. |
| Version 4 | 11/11/2018 | Changes for the SPRINT4. |
| Version 5 | 2/12/2018 | Changes for the SPRINT5. |
|  |  |  |