

A Project Report submitted  
to  
**Software Engineering Lab (IT-3098)**  
on  
**Online Multipurpose Healthcare System (OMHS)**

Submitted By:

Abhishek Acharya	VI-Semester	1405272
Animesh Acharya	VI-Semester	1405283
Ashish Kumar Behera	VI-Semester	1405296
Dinesh Panda	VI-Semester	1405299
Kunja Bihari Mishra	VI-Semester	1405309

Under the Guidance of

Dr. Prachet Bhuyan  
Associate Professor & Associate Dean(T&P)



**School of Computer Engineering  
KiiT University  
Bhubaneswar-24  
Academic Year : 2016-17**

## ACKNOWLEDGEMENT

Apart from our efforts, the success of this project depends largely on the encouragement and guidelines of many others. We take this opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project.

We take this immense pleasure in thanking and warmly acknowledging the continuous encouragement, invaluable supervision, timely suggestions and inspired guidance offered by our project mentor **Dr. Prachet Bhuyan, Associate Professor & Associate Dean (T&P), School of Computer Engineering, KiiT University**, in bringing this project to a successful completion.

We are grateful to **Dr. Samresh Mishra, Dean, School of Computer Engineering**, for permitting us to make use of the facilities available in the department to carry out the project successfully.

We also like to express our gratitude to **Purnendu Sir and Manorama Ma'am** for their assistance in the project completion. We also thank our subject teacher **Mr. Somenath Bera, Assistant Professor, School of Computer Engineering**, for clearing our doubts regarding UML diagrams.

We also express our sincere thanks to our friends, from whom we got new ideas for our project. We would also like to express our heartfelt thanks to each of our beloved parents for their blessings, helps, motivations and wishes for the successful completion of the project.

Finally we extend our gratefulness to one and all who are directly or indirectly involved in the successful completion of this project work.

Abhishek Acharya	VI Sem	1405272
Animesh Acharya	VI Sem	1405283
Ashish Kumar Behera	VI Sem	1405296
Dinesh Panda	VI Sem	1405299
Kunja Bihari Mishra	VI Sem	1405309

# CONTENTS

<b>Sl. No</b>	<b>Topic</b>	<b>Page Number</b>
<b>1</b>	<b>Introduction</b>	<b>(02-03)</b>
<b>1.1</b>	Methodology	02
<b>2</b>	<b>Requirement and Analysis</b>	<b>(04-06)</b>
<b>2.1</b>	System Requirement	04
<b>2.2</b>	Non-Functional Requirement	04
<b>2.3</b>	Functional Requirement	05
<b>3</b>	<b>Design</b>	<b>(07-19)</b>
<b>3.1</b>	MVC Architecture	07
<b>3.2</b>	Three Tier Architecture	08
<b>3.3</b>	E-R Diagram	09
<b>3.4</b>	UML	11
<b>4</b>	<b>Implementation</b>	<b>(20-33)</b>
<b>4.1</b>	Home Page	20
<b>4.2</b>	Contact Us	20
<b>4.3</b>	For Hospital	21
<b>4.4</b>	For Patient	26
<b>4.5</b>	For Doctor	30
<b>5</b>	<b>Testing</b>	<b>(34-39)</b>
<b>5.1</b>	Testing Strategy	34
<b>5.2</b>	Unit Testing	34
<b>5.3</b>	System and Integration Testing	34
<b>5.4</b>	Beta Testing	36
<b>5.5</b>	Browser Testing	36
<b>5.6</b>	Environment Requirement	36
<b>5.7</b>	Network	37
<b>5.8</b>	Features to be Tested	37
<b>6</b>	<b>Future Scope</b>	<b>(40-40)</b>
<b>7</b>	<b>Conclusion</b>	<b>(41-41)</b>
<b>8</b>	<b>References</b>	<b>(42-42)</b>

## **ABSTRACT**

The Online Multipurpose Healthcare System aims in bringing together Hospitals, Doctors and Patients into one platform. OMHS, a web application, is the perfect solution to one of the few areas not digitalized in India. OMHS tackles the chaotic system of healthcare head-on and provides a three-way portal which is not only patient friendly but also doctor friendly. OMHS allows a patient to select appointments as per convenience with only a few clicks. It also provides a patient to store all the prescriptions online. OMHS provides flexibility to doctors by allowing them to select the number of patients they want to examine in a day. Finally, even hospitals benefit from OMHS as it equips the hospitals to manage doctors, providing a level of transparency like no other.

**Key Words:** patient, hospital, doctor, healthcare, prescription, online, appointment.

# 1. Introduction

## 1.1 Methodology

### AGILE METHODOLOGIES

#### **What Is Agile?**

Not a methodology! The Agile movement seeks alternatives to traditional project management. Agile approaches help teams respond to unpredictability through incremental, iterative work cadences and empirical feedback.

#### **What is Scrum?**

Scrum is the most popular way of introducing Agility due to its simplicity and flexibility. Because of this popularity, many organizations claim to be “doing Scrum” but aren’t doing anything close to Scrum’s actual definition. Scrum emphasizes empirical feedback, team self-management, and striving to build properly tested product increments within short iterations. Doing Scrum as it’s actually defined usually comes into conflict with existing habits at established non-Agile organizations

### AGILE PRACTICES USED:

- Development process and practices facilitated by a dedicated role (Scrum master).
- Short daily meeting to resolve current issues.
- Sprint review meeting to present completed work.
- Setting up prototypes before the coding stage.
- Distributing the whole into mates for efficiency.
- Simplicity in design.
- Sprint retrospective to learn from previous sprint.

## **Risks involved:**

### **1) Operational Risk**

- ❖ Number of users of the product.
- ❖ Amount and quality of product documentation that must be produced and delivered to the customer.
- ❖ Government constraints in the construction of the product.
- ❖ Is the customer technically sophisticated in the product area?

### **2) Technical risk**

- ❖ Estimated size of the product.
- ❖ The size of the database created or used by the product.
- ❖ Number of changes according to the requirement of the product?
- ❖ Number of other products/systems with which this product must be interoperable.
- ❖ Speed of data access from remote database.
- ❖ Do analysis and design tools deliver method that are appropriate for the product to be built?

### **3) Schedule risk**

- ❖ Wrong time estimation
- ❖ Resources not tracked properly.
- ❖ Unexpected project scope expansion

## **2. Requirement and Analysis**

### **2.1 System Requirements**

#### **MINIMUM HARDWARE REQUIREMENTS**

- Microprocessor: Intel i3 and above
- RAM: 2GB

#### **TECHNOLOGIES REQUIRED**

- Java Enterprise Edition (EE) 1.7
- Operating System (Primarily Used): Windows 8.1 and Windows 10
- Front-End: HTML5, JavaScript, CSS3
- Back-End: MySQL
- Connector: MySQL-Java Connector (version 5.0.8)
- API: Java-Mail 1.4.7
- Server: Tomcat 8.0 / Glassfish 4.1

### **2.2 NON-FUNCTIONAL REQUIREMENTS**

1. 24X7 availability of the server (Tomcat/Glassfish) on which the application is running.
2. Securely store the confidential data, i.e., user details.
3. Browser testing and support for Google Chrome, Opera.
4. Create a detailed UML diagram (Use Case, Class, Activity, Sequence, Collaboration, Component, Deployment, Object, State Chart) for the system and it's sub-components.

## **2.3 FUNCTIONAL REQUIREMENTS**

### **Patients**

- + have the ability to make appointment in their desired hospital and choose their desired day and doctor.
- + can also view their upcoming appointments.
- + can view their health history, i.e., prescriptions of their previous appointments.
- + can create appointments on behalf of someone.
- + can change their registration details.
- + can change their password.

### **Hospitals**

- + can add doctors in their profile with the number of desired appointment slots.
- + can view upcoming appointments.
- + can change doctor details and password.
- + can change their registration details and password.

### **Doctors**

- + can view upcoming appointments.
- + can view today's appointments.
- + can write prescriptions.
- + can view patient health history, i.e., previous prescriptions.

## User Interface

- + *Login Interface*: prompts the user to enter the credentials. Gives error message “Enter Valid Credentials” on incorrect information.
- + *Home Page Interface*: allows users to contact developers.

## Signup/Login Interface

- + Front-End and Back-End data validation
- + *Requirement 1 : Interface Requirements*
  - Username field can accept alphanumeric characters up to 15 characters.
  - Username field cannot be left blank.
  - Password Field must contain at least one alphabet, one digit and one special character.
- + *Requirement 2 : User Verification*
  - One Time Password : creation of random number
  - Time out : token generated will expire after a certain time.

## Prescription Interface

- + *Requirement 1 : Interface Requirement*
  - Dynamic form field creation.
- + *Requirement 2 : Back-End Requirement*
  - Efficient way of passing n number of medicines to the database.

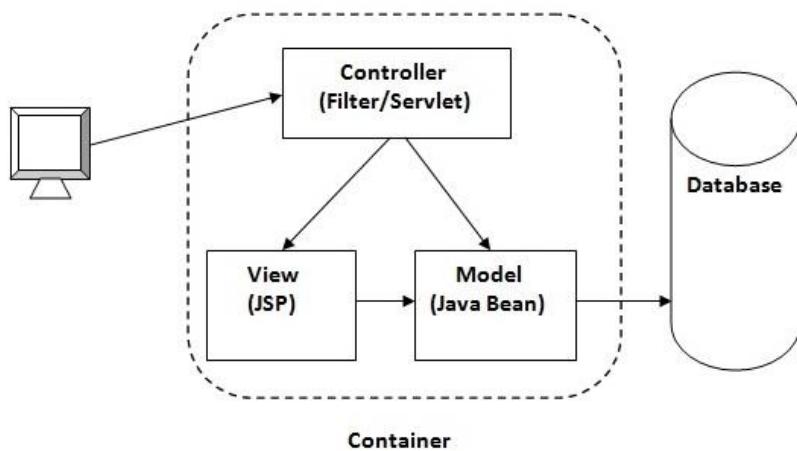
# 3. Design

For making the project “**OMHS**” the team had to learn various concepts.

## 3.1 MVC ARCHITECTURE

**Model View Controller** or **MVC** as it is popularly called, is a software design pattern for developing web applications. It is a design pattern that separates business logic, presentation logic and data. A Model View Controller pattern is made up of the following three parts:

- **Model** – Represents the state of the application, i.e., data. It can also have business logic.
- **View** – Represents the presentation layer, i.e., UI (User Interface).
- **Controller** – Acts as an interface between view and model. The controller intercepts all the incoming requests.



MVC is popular as it isolates the application logic from the user interface layer and supports separation of concerns. Here the controller receives all requests for the application and then works with the Model to prepare any data needed by the View. The View then

uses the data prepared by the Controller to generate final presentable response.

## 3.2 THREE-TIER ARCHITECTURE

Three-tier architecture is a client–server software architecture pattern in which the user interface (presentation), functional process logic ("business rules"), computer data storage and data access are developed and maintained as independent modules, most often on separate platforms. It was developed by John J. Donovan in Open Environment Corporation (OEC), a tools company he founded in Cambridge, Massachusetts.

Apart from the usual advantages of modular software with well-defined interfaces, the three-tier architecture is intended to allow any of the three tiers to be upgraded or replaced independently in response to changes in requirements or technology. For example, a change of operating system in the presentation tier would only affect the user interface code.

Presentation Layer	<ul style="list-style-type: none"><li>● CSS(Cascaded Style Sheet)</li><li>● HTMLv5</li><li>● Bootstrap</li><li>● JavaScript</li></ul>
Business Layer	<ul style="list-style-type: none"><li>● JSP/Servlet(Java1.7)</li></ul>

Database Layer

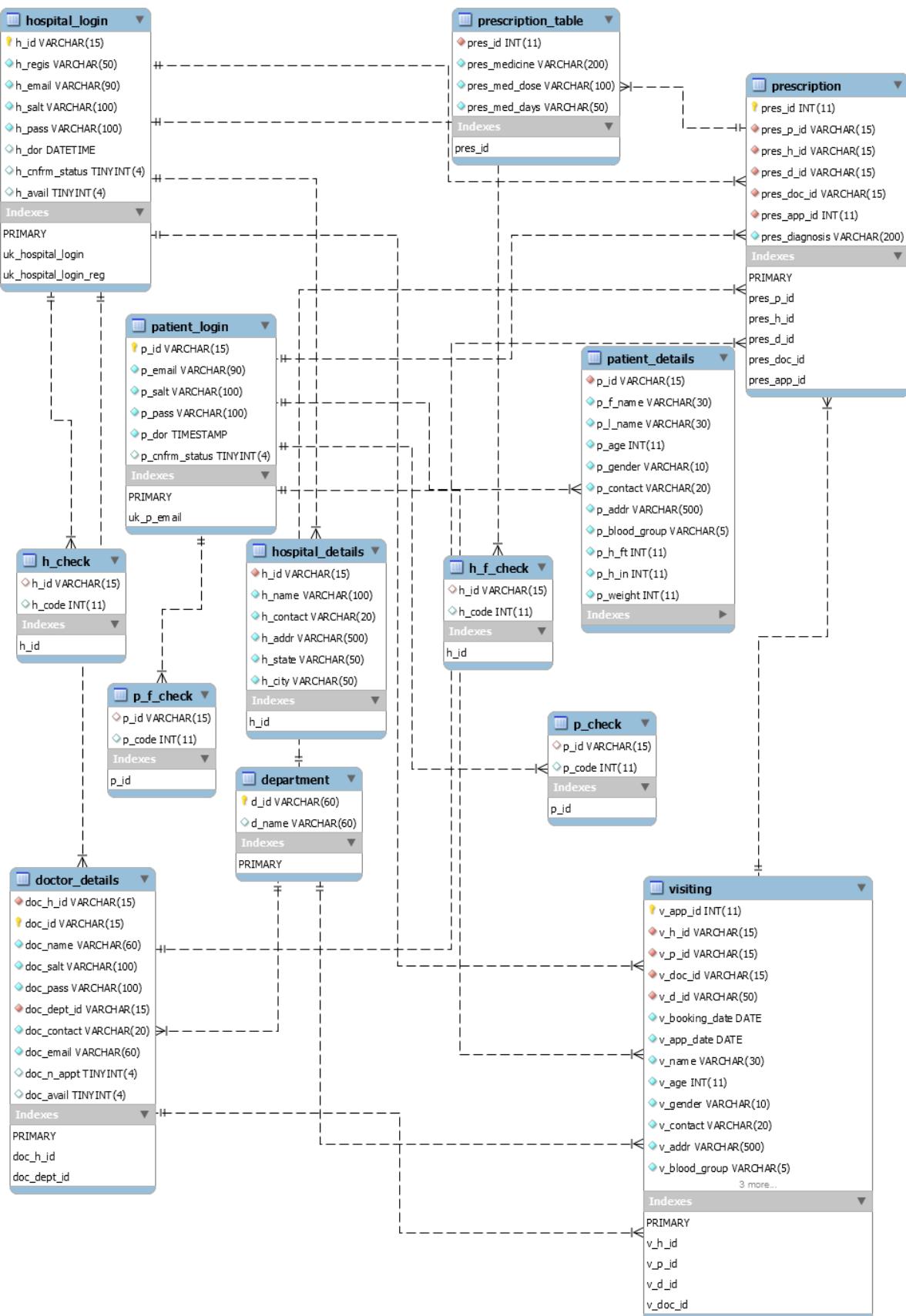
• MySQL

### 3.3 ENTITY-RELATIONSHIP DIAGRAM

An entity-relationship diagram (ERD) is a graphical representation of an information system that shows the relationship between people, objects, places, concepts or events within that system. An ERD is a data modeling technique that can help define business processes and can be used as the foundation for a relational database.

While useful for organizing data that can be represented by a relational structure, an entity-relationship diagram can't sufficiently represent semi-structured or unstructured data, and an ERD is unlikely to be helpful on its own in integrating data into a pre-existing information system.

Three main components of an ERD are the entities, which are objects or concepts that can have data stored about them, the relationship between those entities, and the cardinality, which defines that relationship in terms of numbers.



## **3.4 UNIFIED MODELLING LANGUAGE**

The Unified Modelling Language (UML) is the industry-standard language for specifying, visualizing, constructing and documenting the artifacts of software systems, as well as other non-software systems. The **U** in UML stands for unified because the UML is a unification and standardization of earlier modeling notations of Booch, Rumbaugh, Jacobson, Mellor, Shlaer, Coad, and Wirf-Brock, among others. The UML most closely reflects the combined work of Rumbaugh, Jacobson, and Booch – sometimes called the three amigos. The UML has been accepted as a standard by the Object Management Group (OMG).

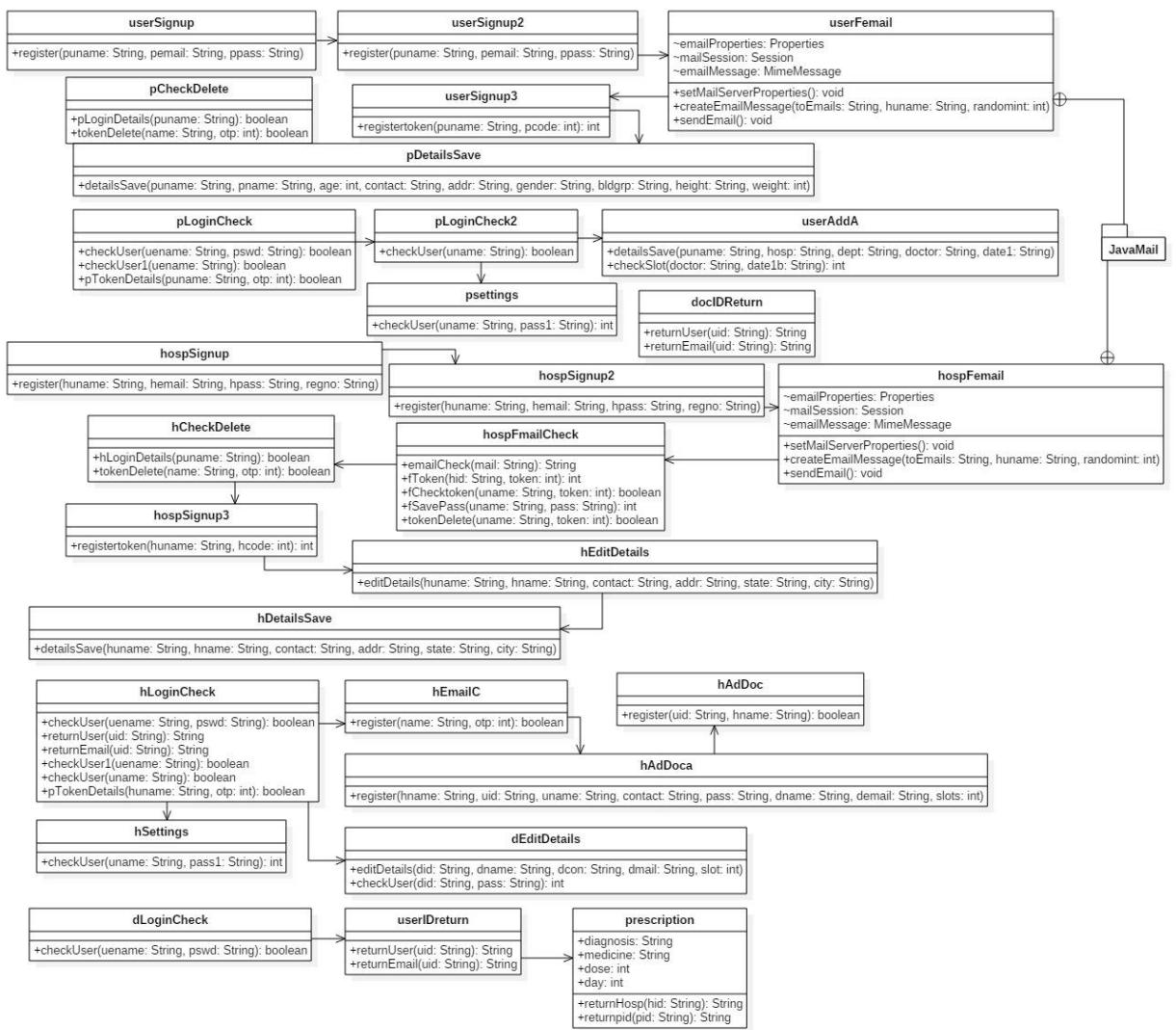
UML simplifies the complex process of software design, making a “blueprint” for construction, and is now the standard notation for software architecture. UML provides both the structural views and behavioral views of the system. A set of programs with different graphical elements is the core part as well as the most expressive presentation in UML. The UML includes nine kinds of diagrams.

There are nine major components of Unified Modelling Language. These can be listed as below.

- Class Diagram
- Object Diagram
- Use-Case Diagram
- Activity Diagram
- Sequence Diagram
- Collaboration Diagram
- State Chart Diagram
- Deployment Diagram
- Component Diagram

### 3.4.1 Class diagram

These diagrams depict the behavioral pattern of the system, i.e. how each and every class is interrelated to the other one, which relationship exists among each of the classes, etc. There would be only one class diagram possible for a single system. Class diagrams of one system can be linked to the class diagrams of another system, provided, there is a multi-system requirement.

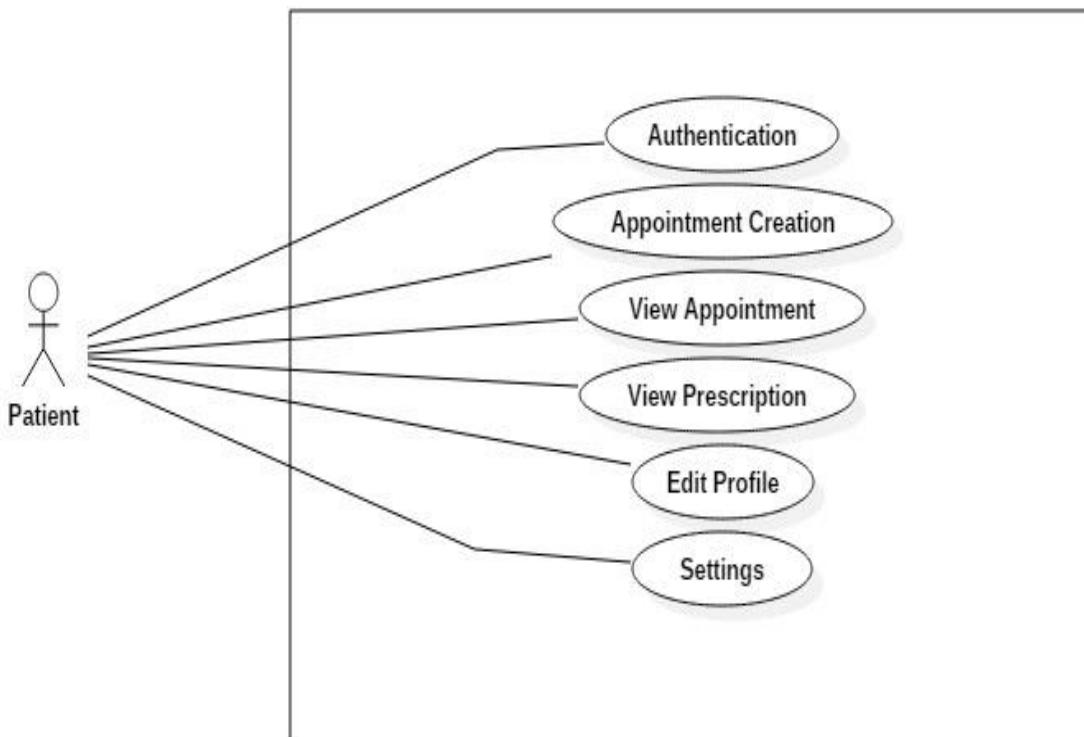


### **3.4.2 Object diagram**

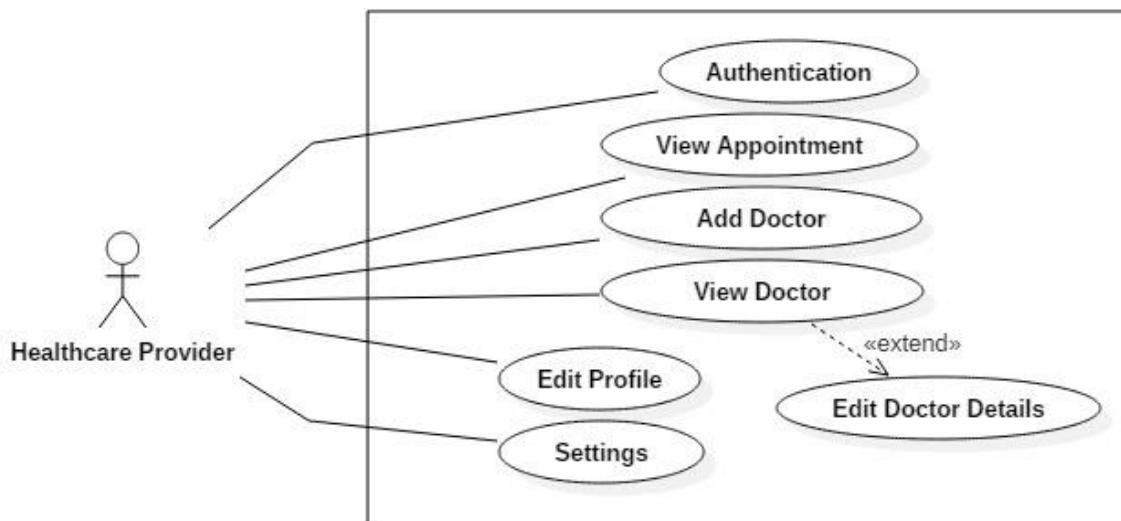
Object diagram is similar to the above mentioned class diagram and is said to be a real entity or an instance of the Class used to mention the extra properties of an entity in addition to the properties depicted by the class.

### **3.4.3 Use case diagram**

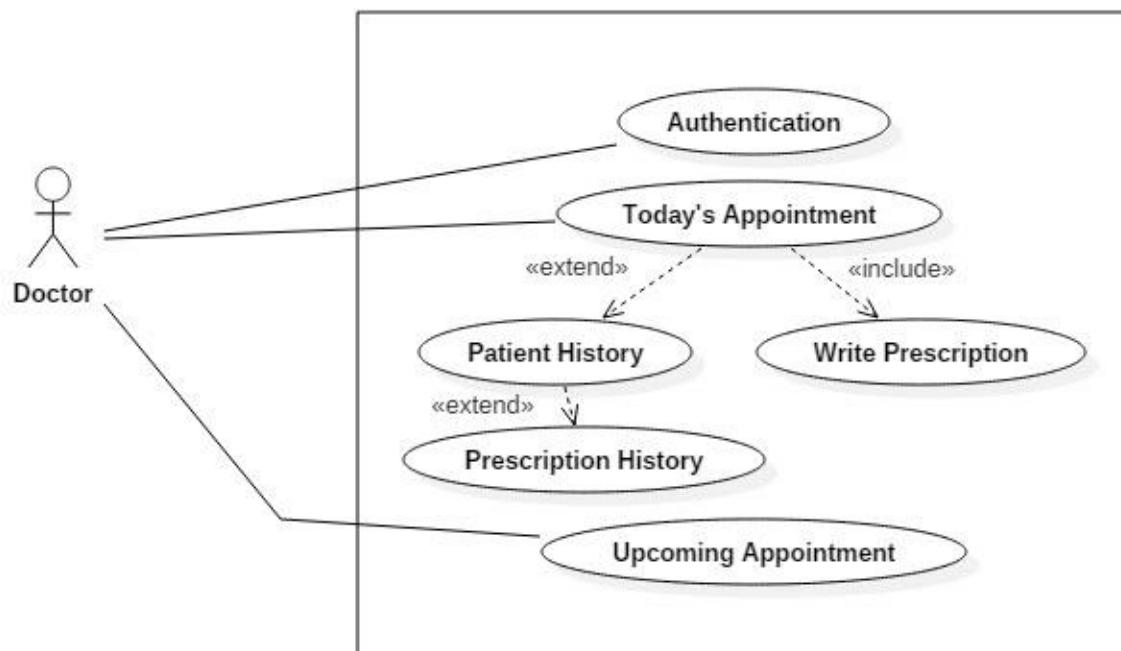
Use case diagram comprises of use cases and actors such that there would be various kinds of relationships among the use cases and the actors. A use case diagram shows all the actions that a particular actor needs to perform throughout the system at any point of time. There would be only one use case diagram per each system.



## **USE CASE DIAGRAM OF PATIENT**



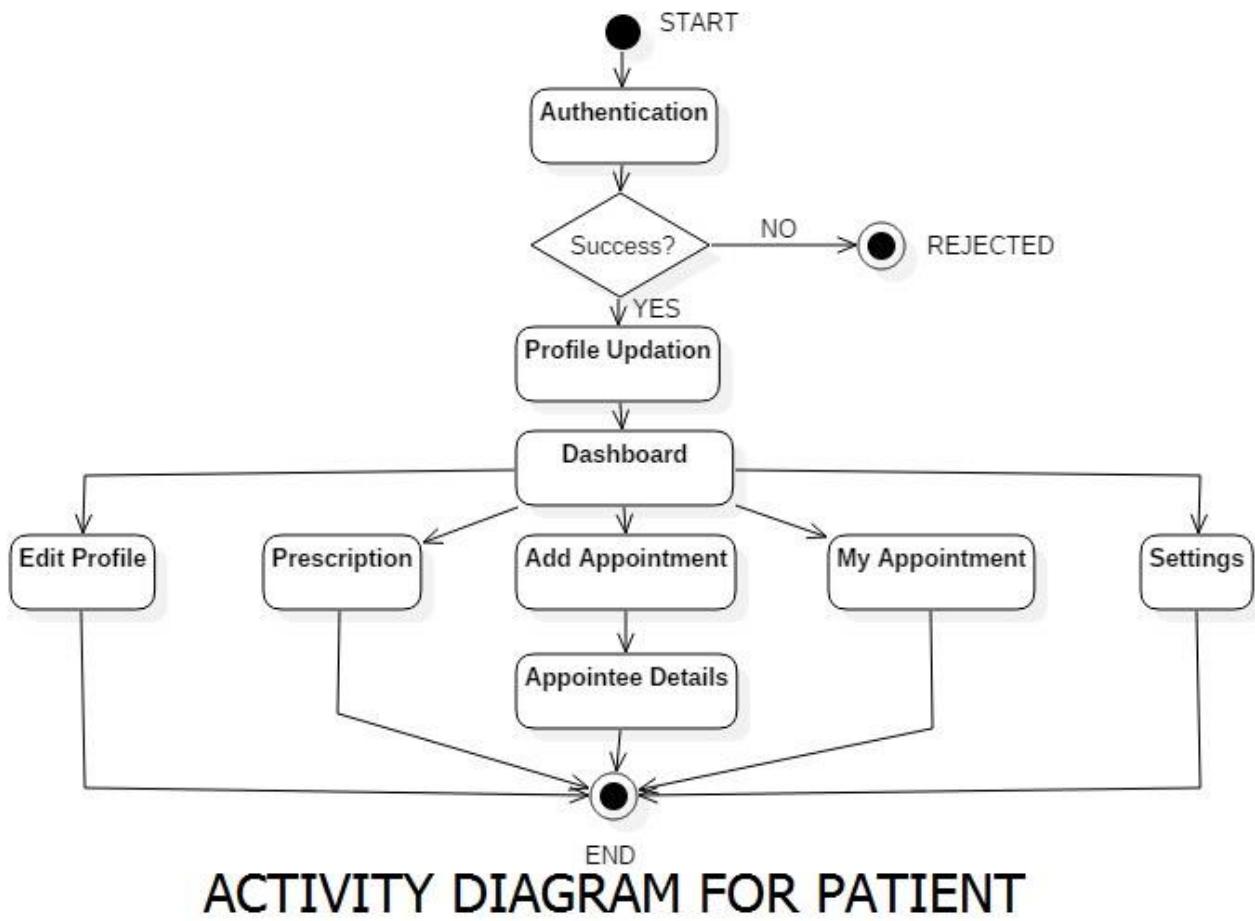
## USE CASE DIAGRAM FOR HEALTHCARE PROVIDER

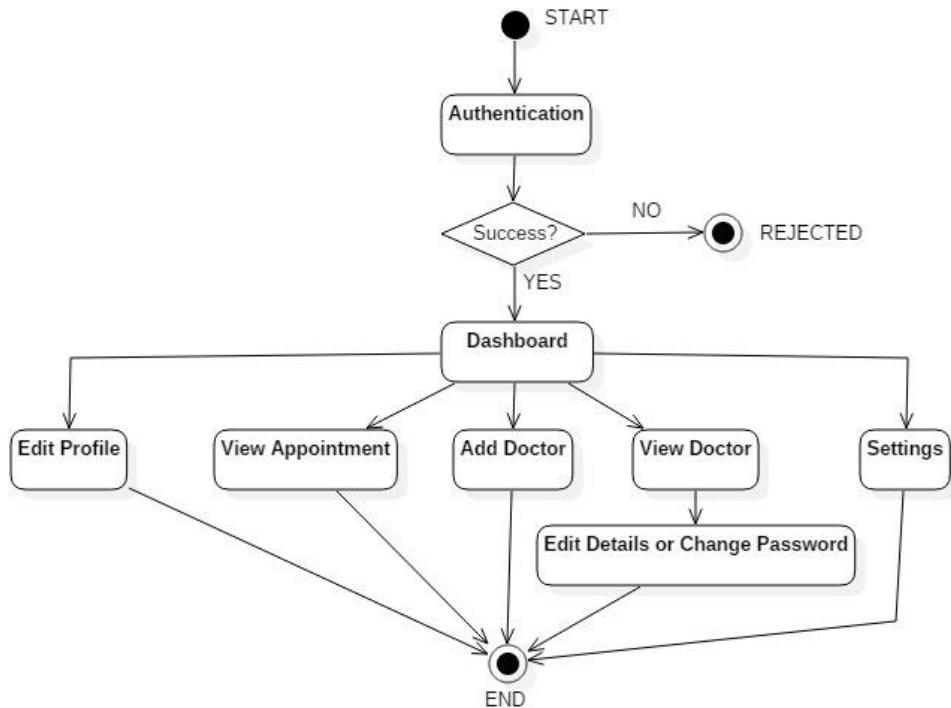


## USE CASE DOCTOR

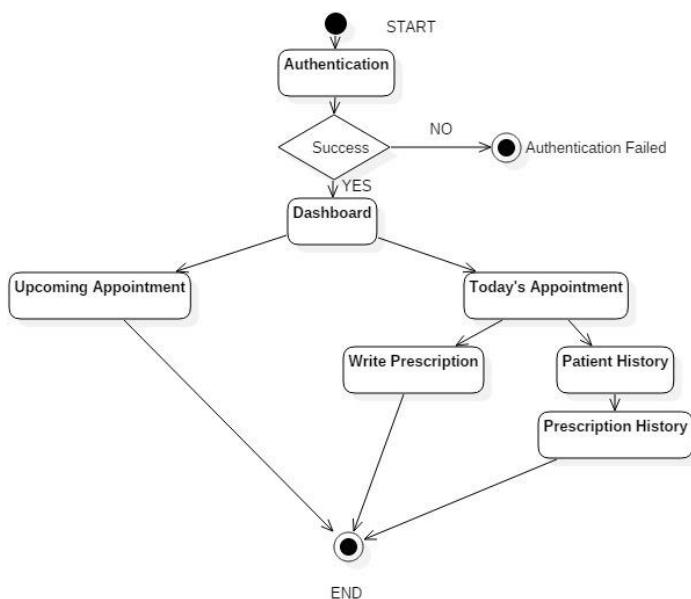
### 3.4.4 Activity diagram

This diagram denotes the structural flow of the activities in the form of flow chart with decision boxes enhanced and hence is also used for troubleshooting like raising exceptions when a particular action is done and the alternative to be done when something abnormal is done. There can be only one activity diagram for the entire system including all the activities that a system can perform.





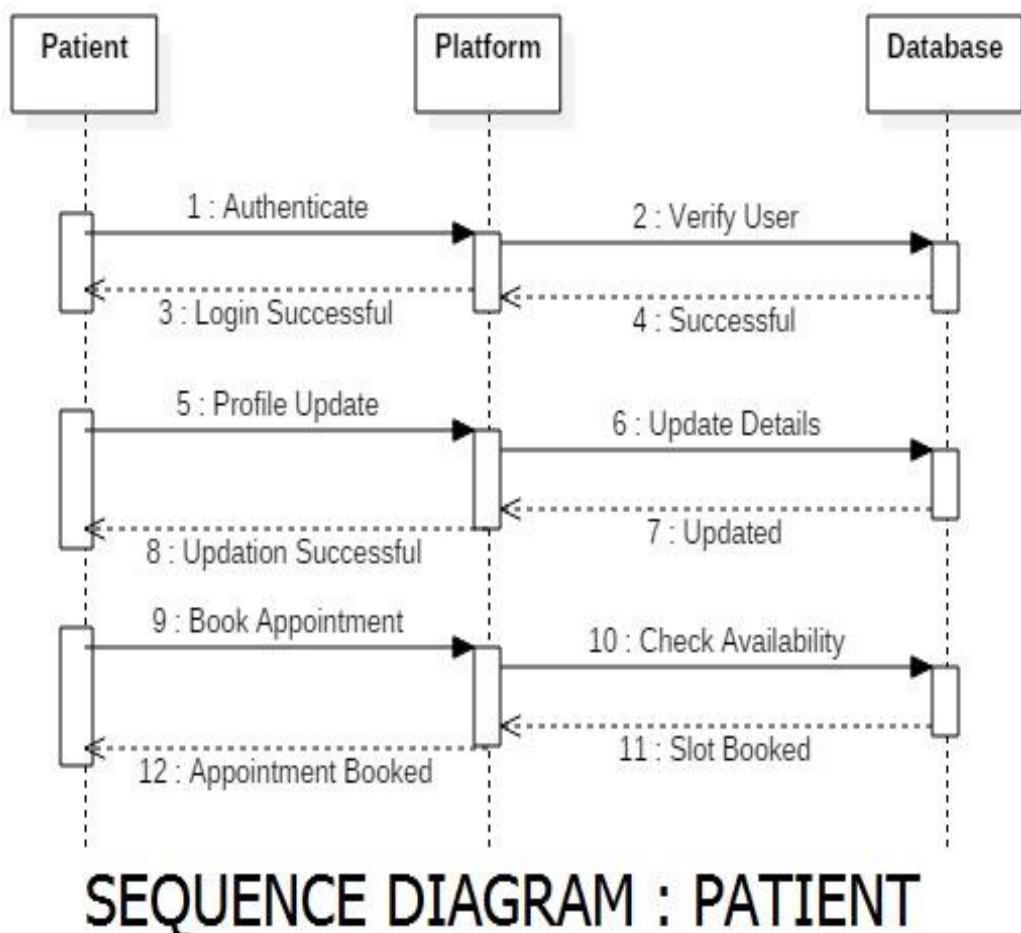
## ACTIVITY DIAGRAM FOR HEALTHCARE PROVIDER

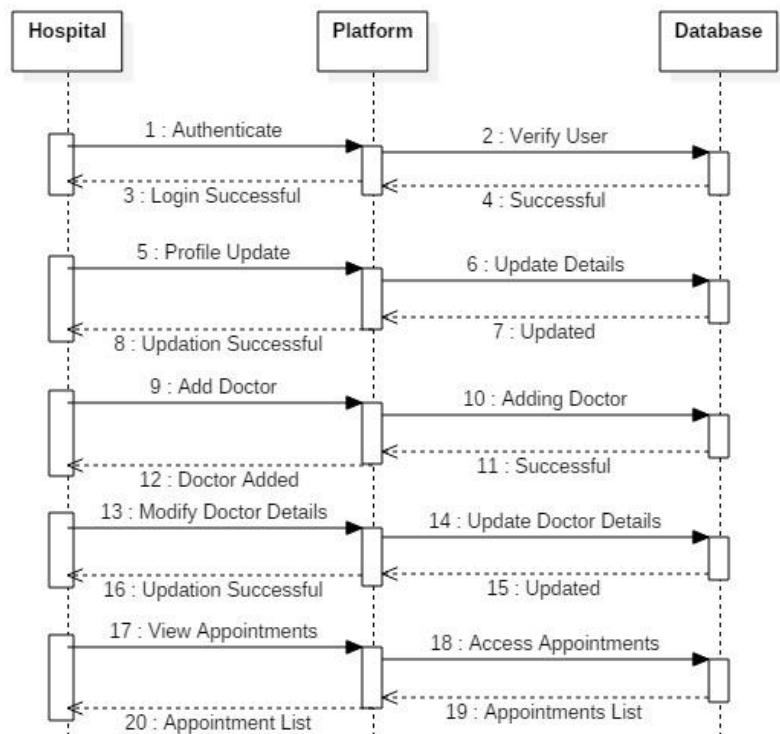


## ACTIVITY DIAGRAM FOR DOCTOR

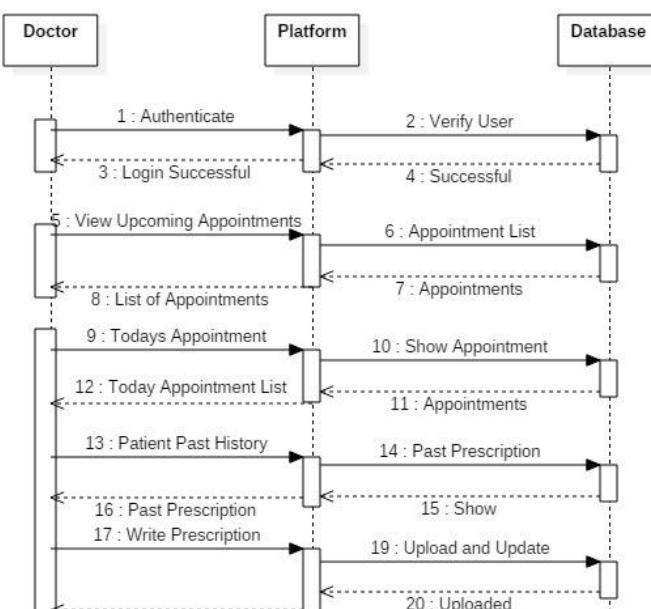
### 3.4.5 Sequence diagram

Sequence diagram, as the name suggests, contains the sequence of flow of actions that are processed through a system and the life lines of the entities, when and how are they accessed. It also contains the security like which entity can process which entity and which one is visible, etc. There can be many number of sequence diagrams per each activity being done.





**SEQUENCE DIAGRAM : HOSPITAL**



**SEQUENCE DIAGRAM : DOCTOR**

### **3.4.6 Collaboration diagram**

This diagram is a polymorphic form of the sequence diagram in which the representation is different but application is the same. If we are able to create one sequence diagram, then it's very simple to create its collaboration diagram with a single key click that varies from software to software. There can be many number of collaboration diagrams per each activity being done because there can be many number of sequence diagrams.

### **3.4.7 State chart diagram**

This diagram is a polymorphic form of the activity diagram in which the representation is different but application is the same. It looks similar to a finite state machine state transition diagrams.

### **3.4.8 Deployment diagram**

Deployment diagram is employed when we need to deploy the application we developed. A single deployment diagram is possible for a single system.

- They model the hardware platform for a system.
- Identify hardware capabilities that affects performance planning and software configuration.

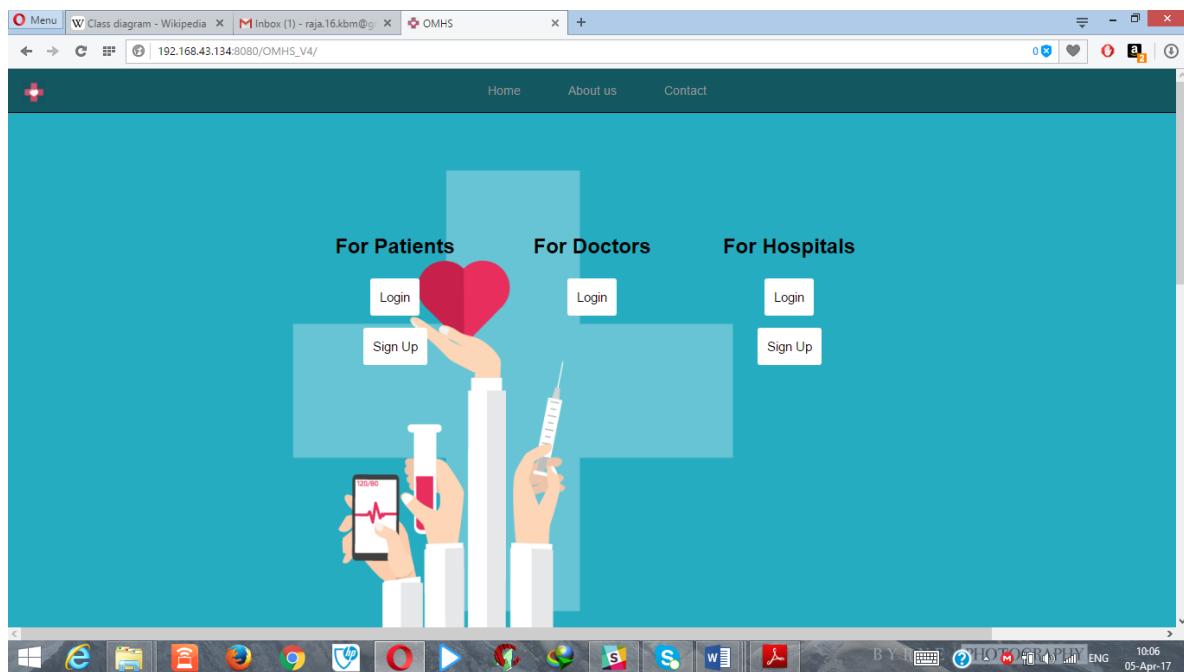
### **3.4.9 Component diagram**

Component diagram represents the components in which the particular application needs to be installed or implemented on. It also shows the type of relation that exists among the various components that are represented. Hence, only a single component diagram representing all the components and their relations is needed for the entire system.

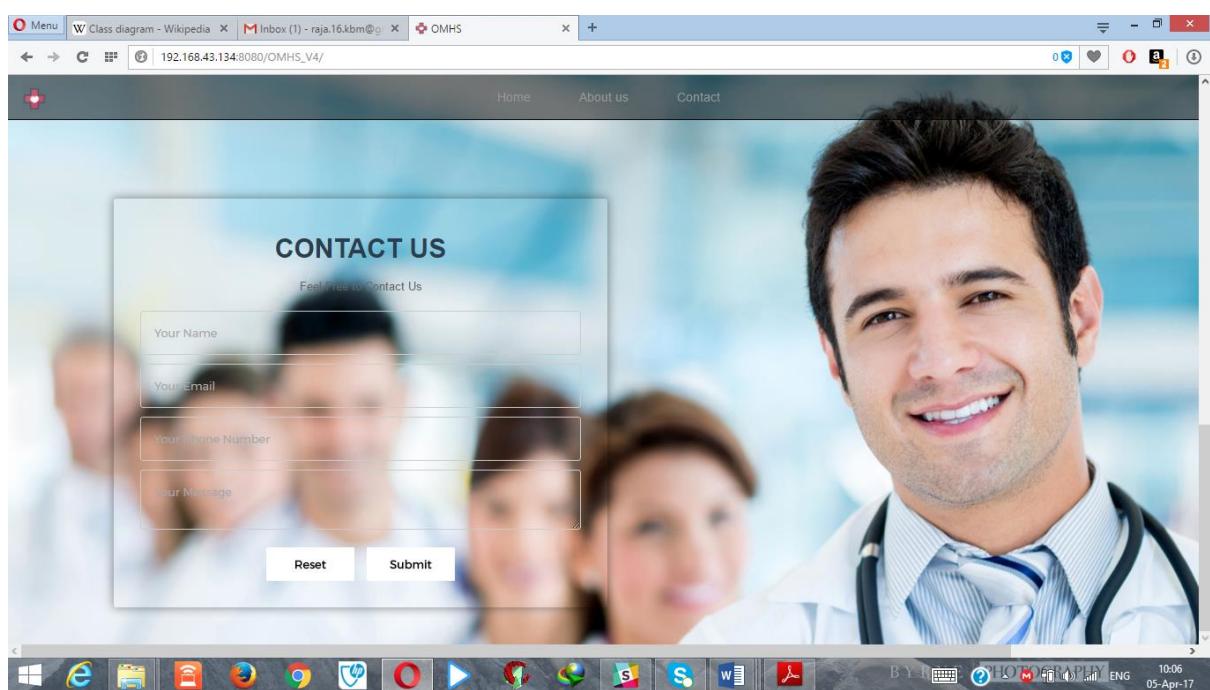
- Model the real software in the implementation environment.
- They provide an accurate picture of existing system prior to making changes or enhancements.

# 4. Implementation (UI)

## 4.1 Home Page

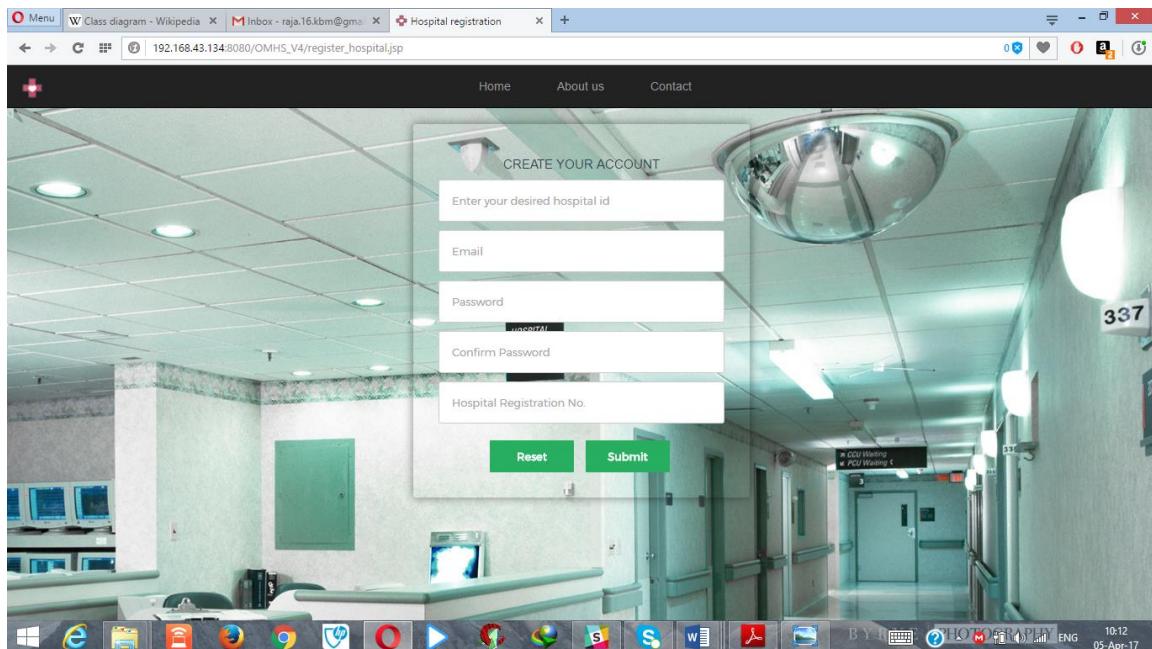


## 4.2 Contact Us

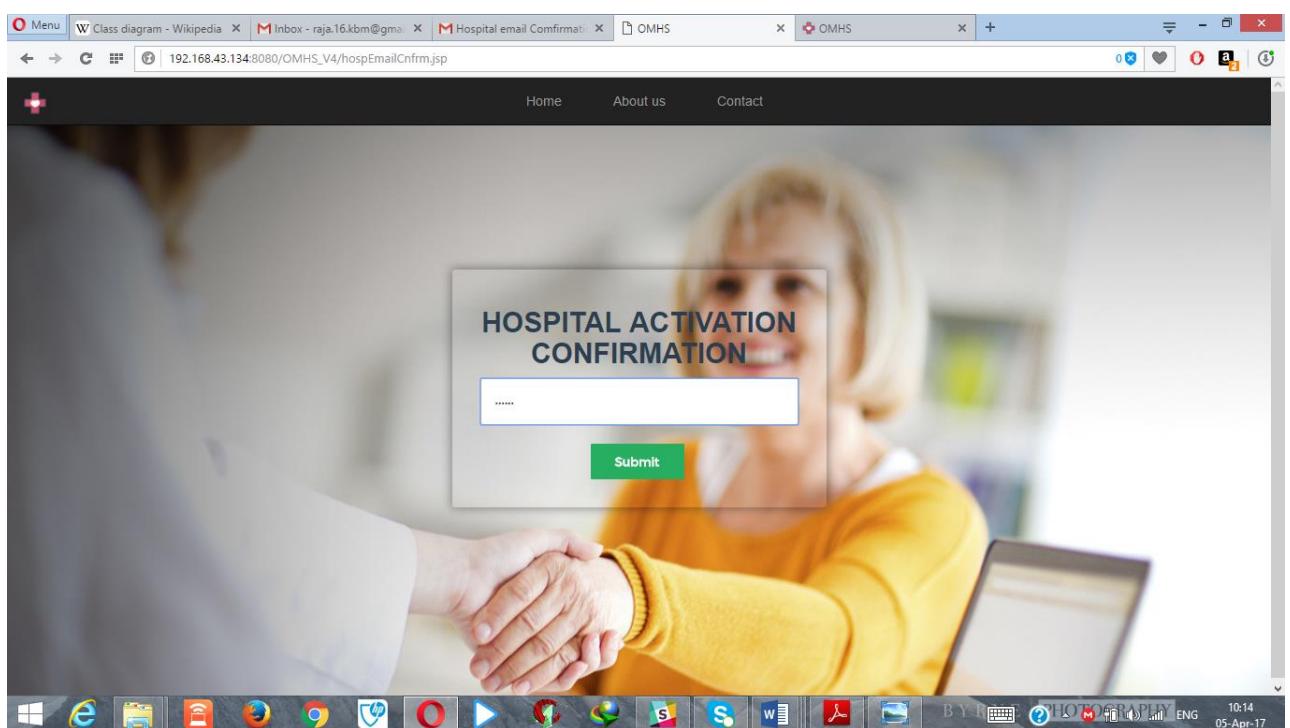


# 4.3 For Hospitals

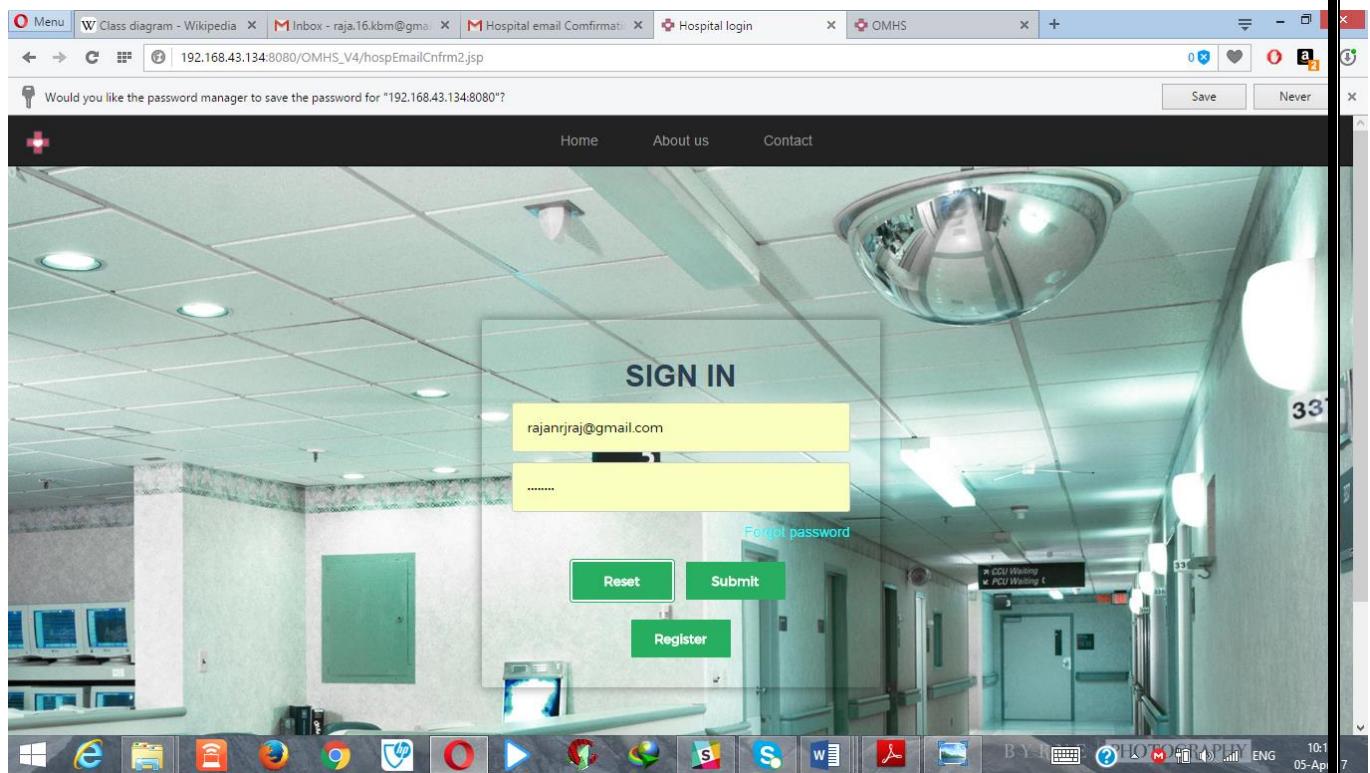
## 4.3.1 Signup



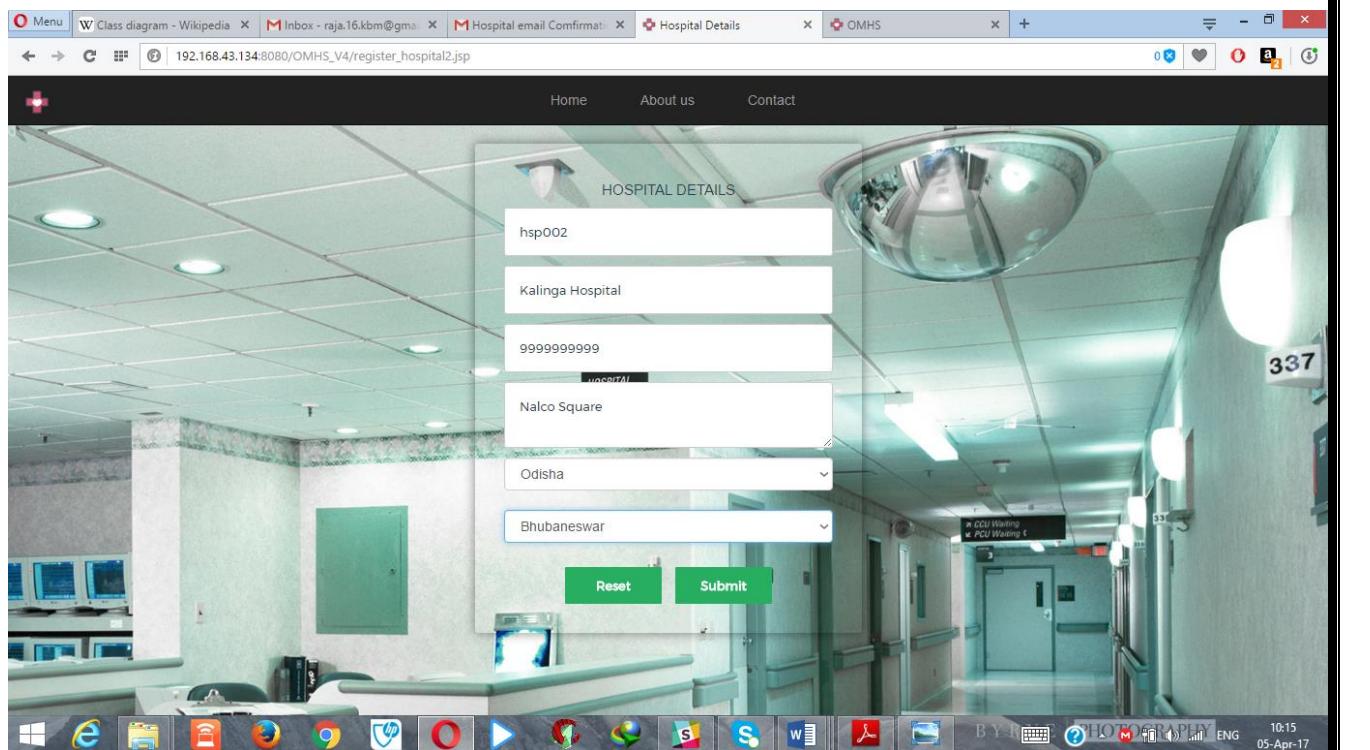
## 4.3.2 OTP Verification



### 4.3.3 Login



### 4.3.4 Edit Profile



## 4.3.5 Dashboard

The screenshot shows a web browser window with the URL [192.168.43.134:8080/OMHS\\_V4/reg\\_hospital2.jsp](http://192.168.43.134:8080/OMHS_V4/reg_hospital2.jsp). The title bar includes tabs for 'Menu', 'Class diagram - Wikipedia', 'Inbox - raja.16.kbm@gmail.com', 'Hospital email Confirmation', 'OMHS-hsp002', 'OMHS', and a '+' button. The main content area has a red header 'hsp002' and a sidebar with a red cross icon. The sidebar menu includes 'Dashboard' (selected), 'Appointments', 'Add Doctor', 'View Doctor', 'Edit Profile', and 'Settings'. The main content area displays the 'Hospital Profile' section with a welcome message and a list of actions:

Welcome to Online Multipurpose Healthcare System.

This is the dashboard. Here you can-

- View the appointments.
- Add a doctor.
- Edit your profile.

The taskbar at the bottom shows various application icons, and the system tray indicates the date as 05-Apr-17 and the time as 10:15.

## 4.3.6 Add Doctor

The screenshot shows a web browser window with the URL [192.168.43.134:8080/OMHS\\_V4/hospAdDoc.jsp](http://192.168.43.134:8080/OMHS_V4/hospAdDoc.jsp). The title bar includes tabs for 'Menu', 'Class diagram - Wikipedia', 'Inbox - raja.16.kbm@gmail.com', 'Hospital email Confirmation', 'OMHS-hsp002', 'OMHS', and a '+' button. The main content area has a sidebar with 'Appointments' (selected), 'Add Doctor' (selected), 'View Doctor', 'Edit Profile', and 'Settings'. The main content area displays the 'Add Doctor to your organisation' form with the following fields:

doc001
Ashish Kumar Behera
8888888888
ashishkumarbehera296@gmail.com
*****
*****
Cardiology
5

**Add Doctor**

The taskbar at the bottom shows various application icons, and the system tray indicates the date as 05-Apr-17 and the time as 10:16.

### 4.3.7 View Doctor

The screenshot shows a web browser window with the URL [192.168.43.134:8080/OMHS\\_V4/hospVDoc.jsp](http://192.168.43.134:8080/OMHS_V4/hospVDoc.jsp). The title bar includes tabs for 'Menu', 'Class diagram - Wikipedia', 'Inbox - raja.16.kbm@gmail.com', 'Hospital email Confirmation', 'OMHS-hsp002', 'OMHS', and a '+' icon. The main content area has a red header 'hsp002' and a sidebar with a red cross icon. The sidebar menu items are: Dashboard, Appointments, Add Doctor, **View Doctor** (highlighted in green), Edit Profile, and Settings. The main content area is titled 'View Doctor' and shows a dropdown menu with 'Cardiology'. Below it is a table listing three doctors:

Doctor Id	Doctor Name	Email	Contact
doc001	Ashish Kumar Behera	ashishkumarbehera296@gmail.com	8888888888
doc002	Abhishek Acharya	1405272@kiit.ac.in	7777777777
doc003	Dinesh Panda	pandadinesh68@gmail.com	9898989898

### 4.3.8 Change Password or Registration Details

The screenshot shows a web browser window with the URL [192.168.43.134:8080/OMHS\\_V4/hospSett.jsp](http://192.168.43.134:8080/OMHS_V4/hospSett.jsp). The title bar includes tabs for 'Menu', 'Class diagram - Wikipedia', 'Inbox - raja.16.kbm@gmail.com', 'Hospital email Confirmation', 'OMHS-hsp002', 'OMHS', and a '+' icon. The main content area has a red header 'hsp002' and a sidebar with a red cross icon. The sidebar menu items are: Dashboard, Appointments, Add Doctor, View Doctor, Edit Profile, and **Settings** (highlighted in green). The main content area is titled 'Settings' and shows a message 'Welcome to Settings'. A modal dialog box is open with the title 'Change your Password/Hospital Registration No.' It contains five input fields: 'hsp002', 'hsp0002', 'rajanraj@gmail.com', 'Old Password' (empty), and 'New Password' and 'Confirm New Password' (both empty). At the bottom are 'Reset' and 'Submit' buttons.

### 4.3.9 Edit Doctor Details

The screenshot shows a web browser window with the URL [192.168.43.134:8080/OMHS\\_V4/hospVDoc3.jsp](http://192.168.43.134:8080/OMHS_V4/hospVDoc3.jsp). The page title is "Manage Selected Doctor". On the left, there is a sidebar menu with options: Dashboard, Appointments, Add Doctor, View Doctor (selected), Edit Profile, and Settings. The main content area contains two forms. The top form is titled "Change Doctor Details" and includes fields for Doctor ID (doc001), Name (Ashutosh Kumar Behera), Password (SSSSSSSS), Email (ashutoshkumarbehera296@gmail.com), and Number of Slots (dropdown menu). A green "Change!" button is at the bottom. Below it is another form titled "Change Doctor Password" with fields for Doctor Password and Confirm Doctor Password, also featuring a green "Change!" button.

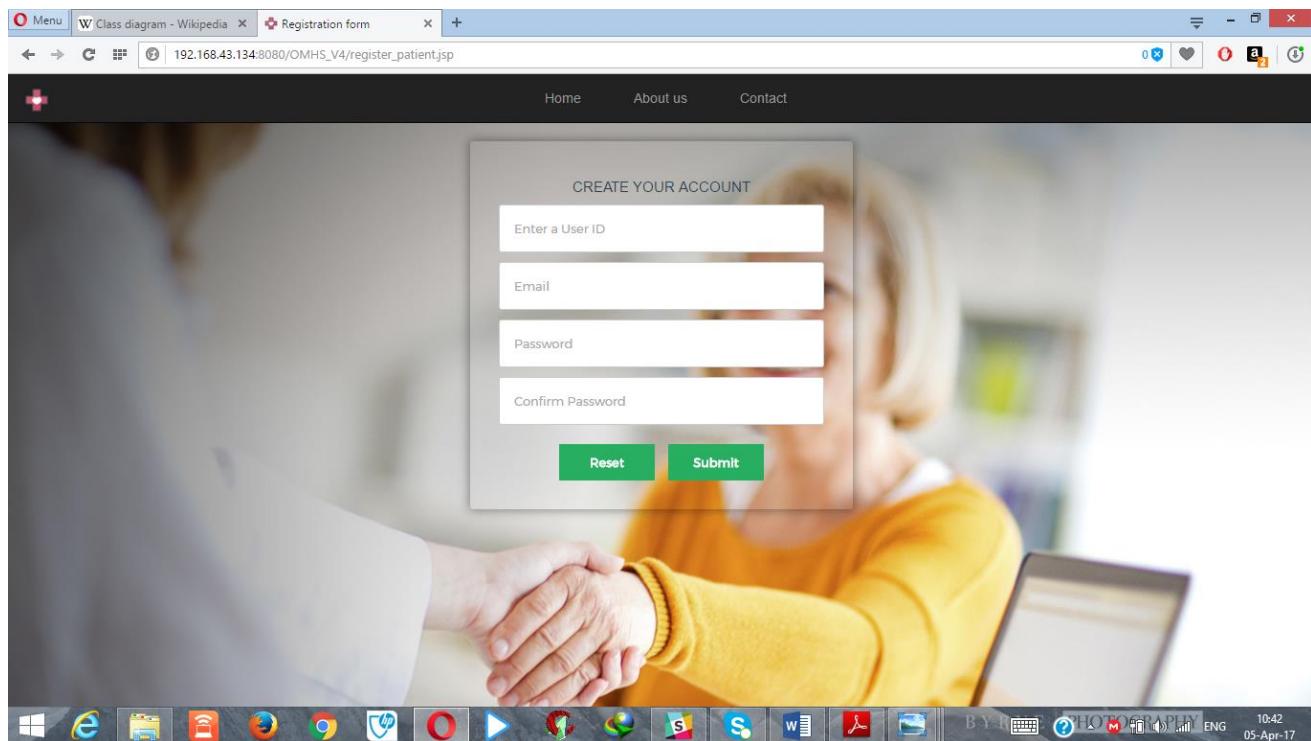
### 4.3.10 View Appointment

The screenshot shows a web browser window with the URL [192.168.43.134:8080/OMHS\\_V4/hospApp2.jsp](http://192.168.43.134:8080/OMHS_V4/hospApp2.jsp). The page title is "OMHS-hsp001". The sidebar menu is identical to the previous screenshot. The main content area has a red header "Hospital Profile". Below it, a section titled "See Appointments" displays a table of patient information:

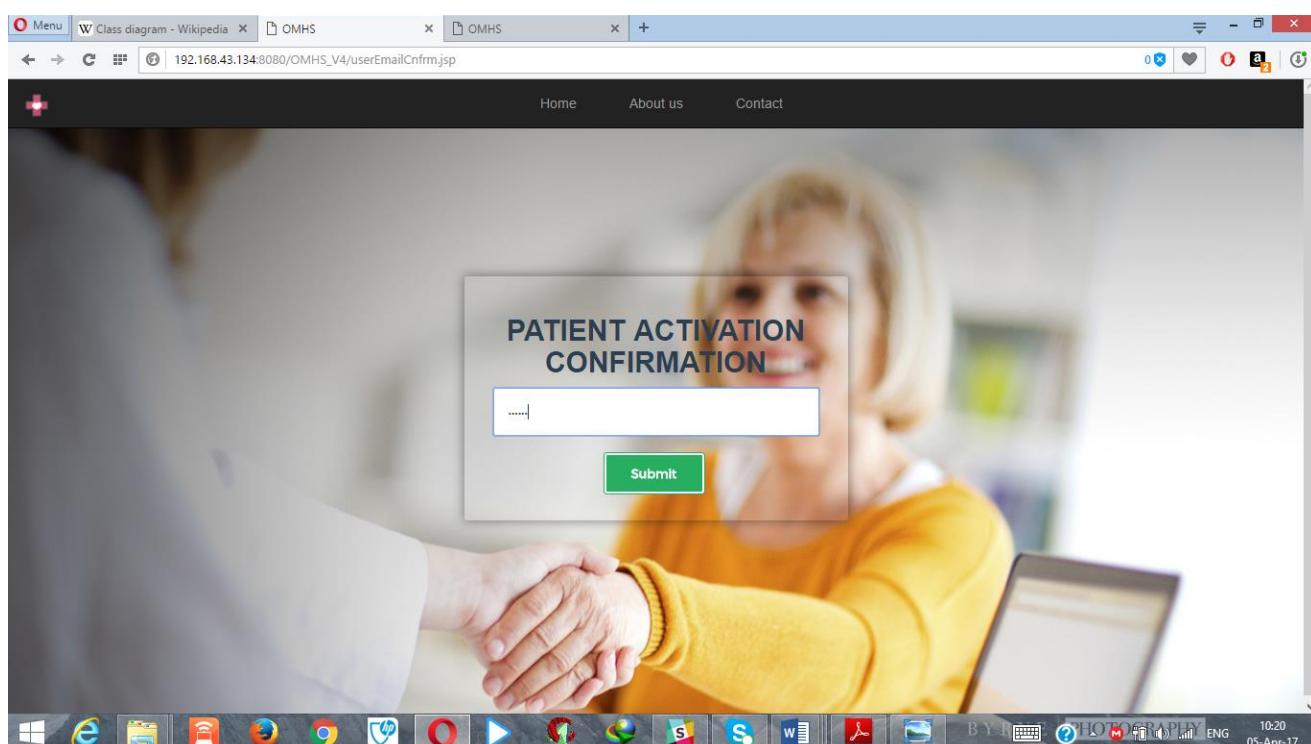
Appointment Id	Patient Id	Doctor Id	Patient Name	Patient Age	Patient Gender	Patient Contact	Patient Address
1002	pat001	dsp002	Ashutosh Panda	22	Male	9797979797	KiIT University

## 4.4 For Patients

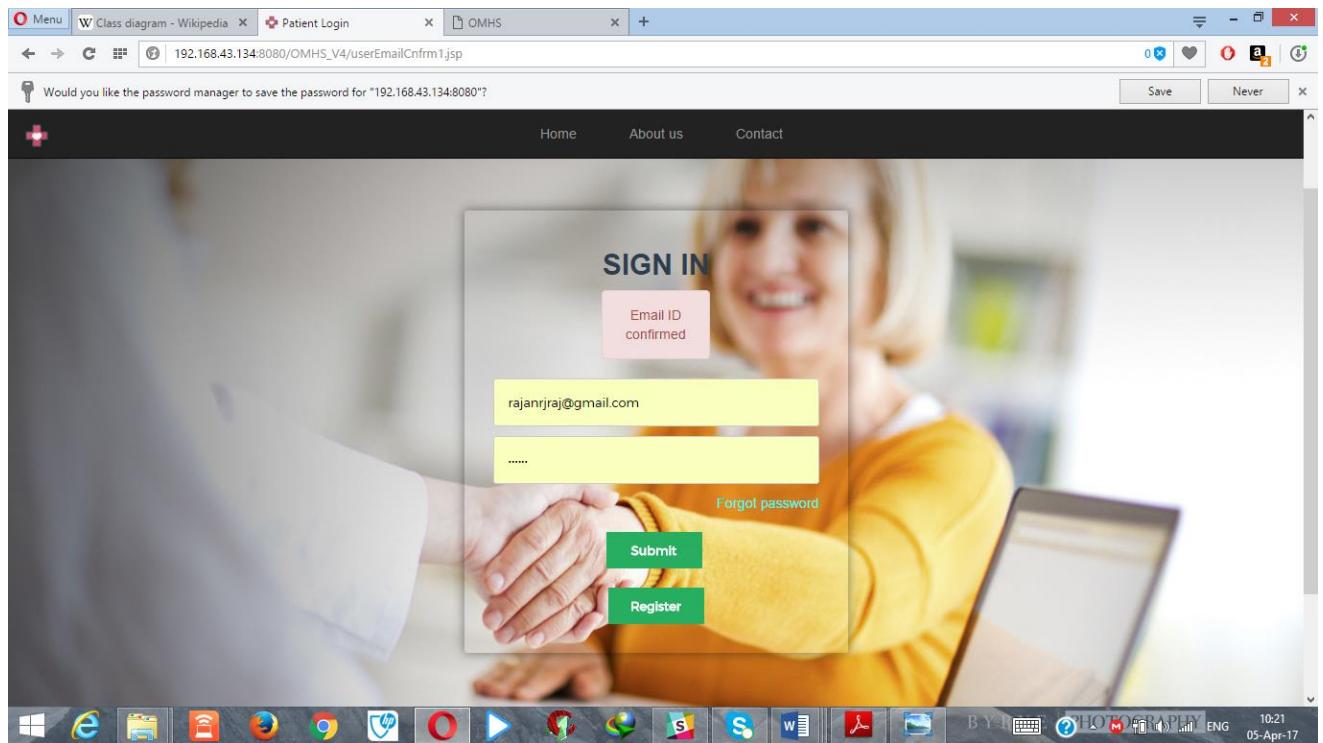
### 4.4.1 Signup



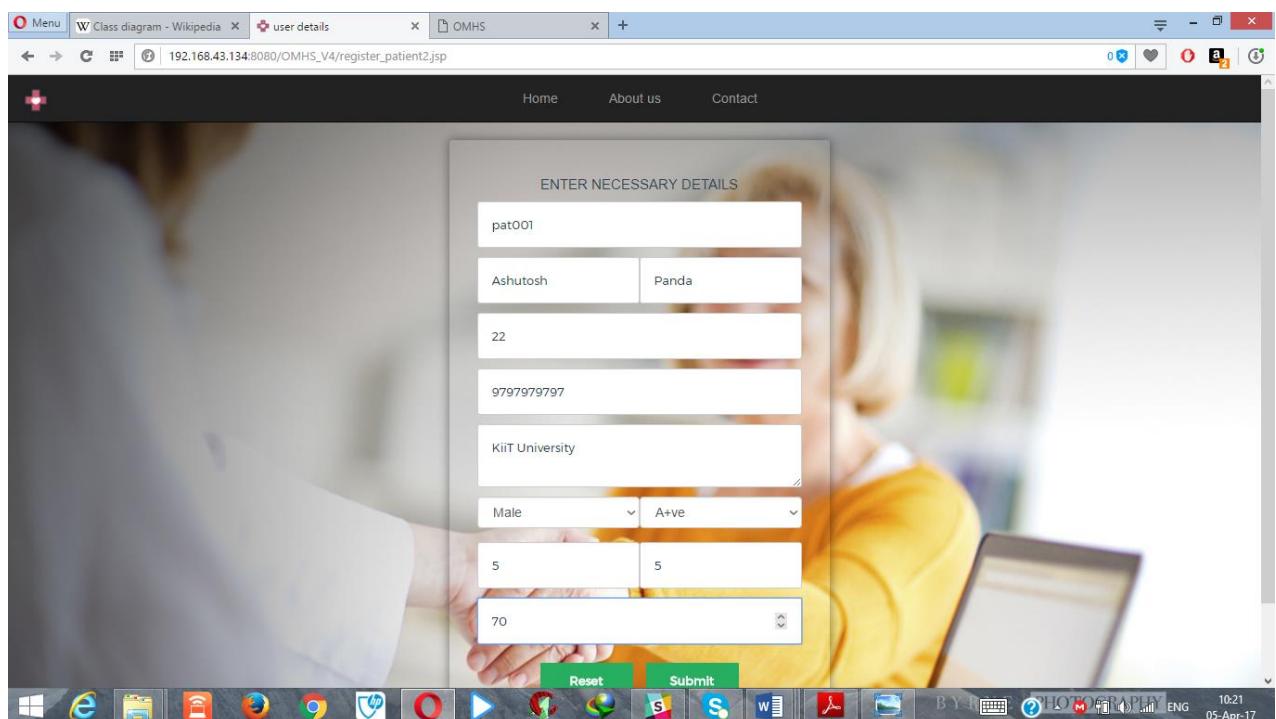
### 4.4.2 Patient Activation



### 4.4.3 Login



### 4.4.4 Enter Details



#### 4.4.5 Make an Appointment

Add Appointment

Welcome to the add appointment page.

Make an Appointment

Odisha

Bhubaneswar

AMRI Hospital

Orthopaedics

Kaiser Williams

06-04-2017

Reset Submit

#### 4.4.6 Edit Patient Details (where one can book for family)

Edit Necessary Details

Ashutosh Panda

22

9797979797

KiiT University

Male A+ve

5 70

Reset Submit

## 4.4.7 My Appointments

The screenshot shows a web-based application interface for managing medical appointments. The top navigation bar includes tabs for 'Menu', 'Class diagram - Wikipedia', 'OMHS-pat001', 'OMHS', and a search bar. A red header bar displays the user ID 'pat001'. On the left, a sidebar menu lists 'Dashboard', 'Add Appointment', 'My Appointments' (which is selected and highlighted in green), 'Prescriptions', 'Edit Profile', and 'Settings'. The main content area is titled 'My Appointments' and displays a table of scheduled appointments. The table columns are: Date of Appointment (yyyy-mm-dd), Hospital Name, Hospital Address, Hospital Contact, Department, Doctor Name, Date of Booking (yyyy-mm-dd), and Appointment for. The data in the table is as follows:

Date of Appointment (yyyy-mm-dd)	Hospital Name	Hospital Address	Hospital Contact	Department	Doctor Name	Date of Booking (yyyy-mm-dd)	Appointment for
2017-04-06	AMRI Hospital	Khandagiri, Bhubaneswar, Odisha	9393939393	Orthopaedics	Kaiser Williams	2017-04-05	Ashutosh Panda
2017-04-07	Kalinga Hospital	Nalco Square, Bhubaneswar, Odisha	9999999999	Cardiology	Ashish Kumar Behera	2017-04-05	Ashutosh Panda
2017-04-07	Kalinga Hospital	Nalco Square, Bhubaneswar, Odisha	9999999999	Cardiology	Abhishek Acharya	2017-04-05	Ashutosh Panda
2017-04-07	Kalinga Hospital	Nalco Square, Bhubaneswar, Odisha	9999999999	Cardiology	Dinesh Panda	2017-04-05	Ashutosh Panda

The taskbar at the bottom of the screen shows various application icons.

## 4.4.8 Your Prescription

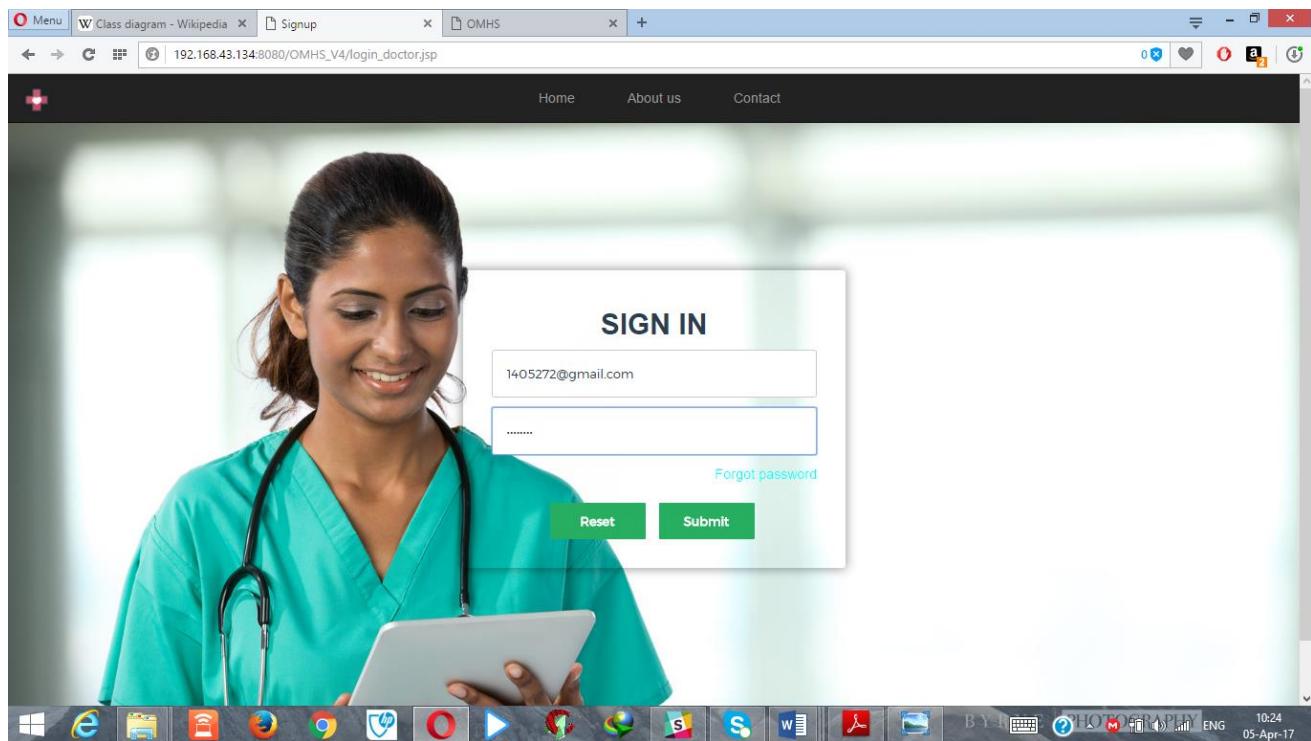
The screenshot shows a web-based application interface for viewing prescriptions. The top navigation bar includes tabs for 'Menu', 'Class diagram - Wikipedia', 'OMHS', and a search bar. A red header bar displays the user ID 'pat001'. On the left, a sidebar menu lists 'Dashboard', 'Add Appointment', 'My Appointments', 'Prescriptions' (which is selected and highlighted in green), 'Edit Profile', and 'Settings'. The main content area is titled 'Your Prescriptions' and displays a table of prescriptions. The table columns are: Medicine, Dosage, and Days. The data in the table is as follows:

Medicine	Dosage	Days
Senarest 500	2	10

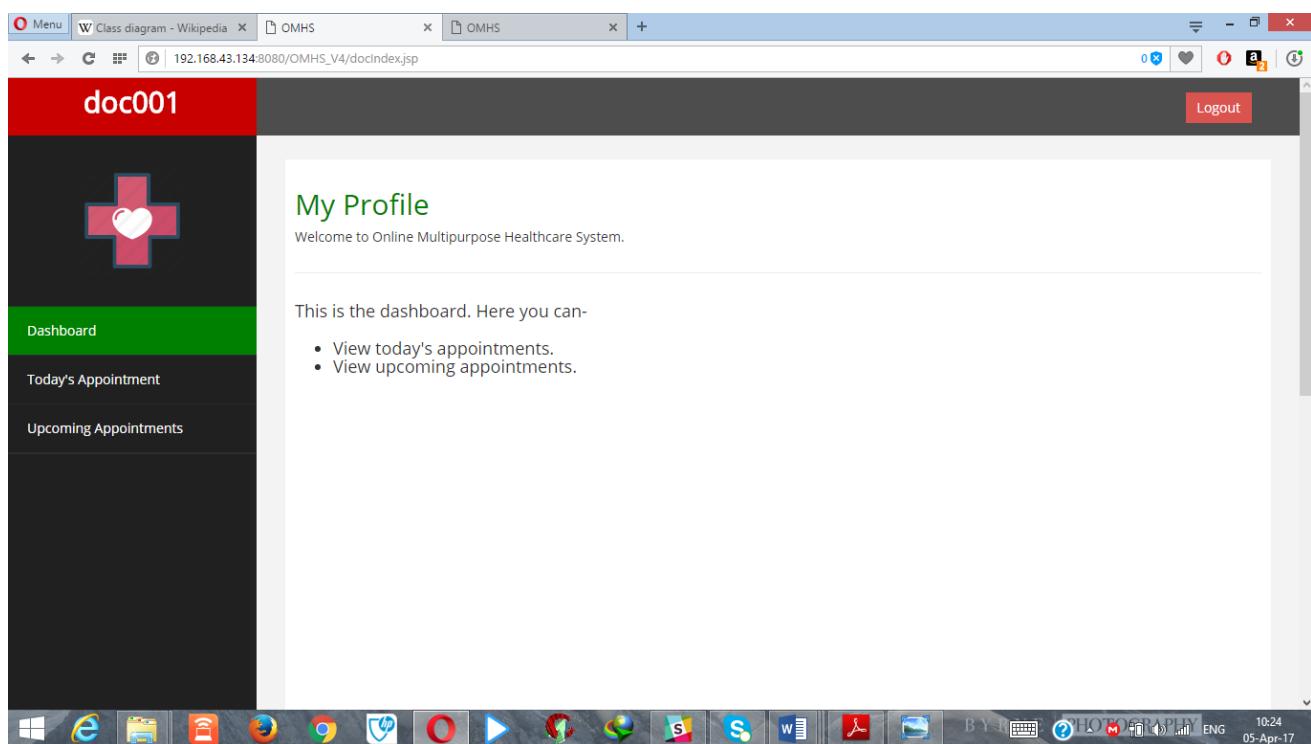
The taskbar at the bottom of the screen shows various application icons.

# 4.5 For Doctor

## 4.5.1 Login



## 4.5.2 Dashboard



### 4.5.3 Upcoming Appointment

The screenshot shows a web browser window with the URL [192.168.43.134:8080/OMHS\\_V4/docUpApp.jsp](http://192.168.43.134:8080/OMHS_V4/docUpApp.jsp). The title bar includes tabs for 'Menu', 'Class diagram - Wikipedia', 'OMHS', and 'OMHS'. A red header bar displays 'doc001'. On the left, a sidebar menu has 'Upcoming Appointments' selected. The main content area is titled 'My Profile' and 'View Upcoming Appointments'. A table lists one appointment:

Appointment Id	Date	Patient Name
1001	2017-04-07	Ashutosh Panda

The taskbar at the bottom shows various application icons.

### 4.5.4 Today's Appointment

The screenshot shows a web browser window with the URL [192.168.43.134:8080/OMHS\\_V4/docTodApp.jsp](http://192.168.43.134:8080/OMHS_V4/docTodApp.jsp). The title bar includes tabs for 'Menu', 'Class diagram - Wikipedia', 'OMHS', and 'OMHS'. A red header bar displays 'dsp002'. On the left, a sidebar menu has 'Today's Appointment' selected. The main content area is titled 'Today's Appointments' and 'See your today's appointments.' A table lists one appointment:

Appointment Id	Patient Name
1002	Ashutosh Panda

The taskbar at the bottom shows various application icons.

## 4.5.5 Write Prescription

Write Prescription for Appointment ID:  
1002

dsp002

Ashutosh

22

A+ve	5	5	70
------	---	---	----

Flu

Senarest	2	10	<span style="color: green;">+</span>
Medicine name	Dosage(in mg)	Days	<span style="color: red;">-</span>

Reset Submit

## 4.5.6 Patient History

Doctor Profile

View Patient History

Date	Department	Prescription For
2017-04-04	Cardiology	Animesh Acharya
2017-04-05	Cardiology	Animesh Acharya
2017-04-05	Dentistry	Animesh Acharya

#### 4.5.7 Prescription History

The screenshot shows a web browser window titled "OMHS" with two tabs open. The active tab is "localhost:8084/OMHS\_V4/docTodApp\_Pres3.jsp?var=1". The page has a header "Doctor Profile" and a sidebar with a red cross icon. The main content area is titled "Prescription History" and displays a single prescription entry:

Medicine	Dosage	Days
Flu	2	10
Sinarest		

# **5. Testing**

Testing section describes what is being tested, such as all the functions of a specific product, its existing interfaces, and integrations of all functions.

## **5.1 Testing Strategy**

It describes the overall approach to testing. For each major group of features or feature combinations, the approach which will ensure that these feature groups are adequately tested are specified. The major activities, techniques, and tools which are used to test the designated groups of features are specified.

## **5.2 Unit Testing**

Unit testing is a software testing method by which individual units of source code, sets of one or more computer program modules together with associated control data, usage procedures, and operating procedures are tested to determine whether they are fit for use. Each test case is independent from the others. Substitutes such as method subs, mock objects, fakes and test harnesses can be used to assist testing a module in isolation.

## **5.3 System and Integration Testing**

System Integration Testing (SIT) involves the overall testing of a complete system of many subsystem components or elements. The system under test may be composed of hardware, or software, or hardware with embedded software, or hardware/software. The

beginning of system integration testing is often the first time that an entire system has been assembled such that it can be tested as a whole. In order to make system testing most productive, the many constituent assemblies and subsystems will have typically gone through a subsystem test and successfully verified that each subsystem meets its requirements at the subsystem interface level.

Following procedures are performed during System and Integration Testing:

#### For Patients

HOME-> REGISTER-> LOGIN-> ADD APPOINTMENT-> EDIT DETAILS-> BOOKING SUCCESSFUL

HOME-> LOGIN-> VIEW APPOINTMENT

HOME -> LOGIN -> VIEW PRESCRIPTION LIST -> VIEW PARTICULAR PRESCRIPTION

#### For Hospitals

HOME-> REGISTER-> LOGIN-> ADD DOCTOR

HOME-> LOGIN-> VIEW APPOINTMENTS

HOME-> VIEW DOCTOR-> EDIT DOCTOR DETAILS

HOME-> VIEW DOCTOR-> CHANGE PASSWORD

#### For Doctors

HOME-> LOGIN-> VIEW UPCOMING APPOINTMENT

HOME -> LOGIN -> TODAY'S APPOINTMENT -> SELECT APPOINTMENT-> WRITE PRESCRIPTION

## **5.4 Beta Testing**

Beta testing is also known as user testing takes place at the end users site by the end users to validate the usability, functionality, compatibility and reliability testing. Beta testing adds value to the software development life cycle as it allows the “real” customer an opportunity to provide inputs into the design, functionality and usability of a product. These inputs are not only critical to the success of the product but also an investment into future products when the gathered data is managed effectively.

## **5.5 Browser Testing**

Cross Browser Testing is a process to test web applications across multiple browsers. Cross browser testing involves checking compatibility of the application across multiple web browsers and ensures that the web application works correctly across different web browsers.

Unit Testing and System and Integration Testing we performed on different browsers. The browsers on which the web application is tested on are as follows:

- Google Chrome (version 57.0.2987.133)
- Opera (version 43.0.2442.1144)

## **5.6 Environment Requirements**

A testing environment is a setup of software and hardware on which the testing team is going to perform the testing of the newly built software product. This setup consists of the physical setup which includes hardware, and logical setup or any other software components required to run the software product.

## 5.7 Network

Network with internet or without internet, with in the same network or private network for testing, so that the congestion that is occurred during testing doesn't hamper the other members involved in development.

## 5.8 Features to be Tested

All web application features and combinations of the platform that has been tested are:

- ❖ Form Submission
- ❖ SQL Injection Testing
- ❖ Dynamic Form Field Creation
- ❖ Website Responsiveness
- ❖ OTP Authentication
- ❖ Automated OTP Notification

Test Case	Test Procedure	PreCondition
Check the fields	Check various input for front-end validation	Usage of HTMLv5
Check for buttons	View the buttons in the login	Usage of Bootstrap
Check Email Notifiaction	Check by signing up with different email	Usage of JavaMail

### Testing OMHS Login Module

Expected Result	Reference to Detailed Design
Wrong input should show Notification	Login Module
Should change color while hover	Login Module
Should send OTP to the email of user	Notification

### Testing OMHS Login Module

Test Case	Test Procedure	PreCondition
Check the dashboard	Check the hyperlinks	Usage of HTML5, Bootstrap
Create appointment	Create appointments for different hospitals	Hospitals should have created doctor accounts
Prescription History	Show Prescription History	Patient must have visited Doctor
Change Settings	Change Fields	Account should be present

## TESTING PATIENT MODULE

Expected Result	Reference to Detailed Design
All pages should open	Dashboard
Can create appointments	Appointment
Show Diagnosis and Medicines	Prescription
Details must change	Change Settings

## PATIENT TESTING MODULE

Test Case	Test Procedure	PreCondition
Check the dashboard	Check the hyperlinks	Usage of HTML5, Bootstrap
View appointment	Check appointments	Patients should have made appointment
Create Doctor Account	Create Doctor Account	Doctor Id should be Unique
Change Settings	Change Fields	Account should be present

## TESTING HOSPITAL MODULE

Expected Result	Reference to Detailed Design
All pages should open	Dashboard
Can view appointments	Appointment
Can create Account for doctors	Create Doctors
Details must change	Change Settings

## TESTING HOSPITAL MODULE

Test Case	Test Procedure	PreCondition
Check the dashboard	Check the hyperlinks	Usage of HTML5, Bootstrap
View today appointment	Check appointments	Patients should have made appointment
View upcoming appointment	Check appointments	Patients should have made appointment

## TESTING DOCTOR MODULE

Expected Result	Reference to Detailed Design
All pages should open	Dashboard
Can view today appointments	Appointment
Can view upcoming appointments	Appointment

## TESTING DOCTOR MODULE

## **6. Future Scope**

This web application can be developed in the future. Various functions and features can be implemented which will make it more useful and dynamic. The additional features that can be implemented are as follows:

- Upload Prescription: The patient can easily upload his/her past medication details by uploading it in his/her profile. It will be easier for the user to track every past medications.
- Better hospital verification to avoid fraud.
- Introducing payment portal to encourage cashless transaction.

## **7. Conclusion**

OMHS is the result of a small effort from us to create a viable software to cater the needs of people in the Health Sector. OMHS is a unique blend of all facets of the Health Sector in one go. However, this project has a lot of potential for scaling. Things like accurate appointment time, payment gateway, and unique identification system can be implemented in the future.

During the implementation process of OMHS, we came across many hurdles which gave us valuable experience and knowledge about how to overcome issues while developing an application. It also gave us a lot of invaluable knowledge about the concepts of Object Oriented Programming, Database Management and Software Engineering.

As is often the case, no application is perfect, and there is always a scope for improvement. We definitely are open to feedbacks of all sorts which can allow us to make this application a success.

**Help us become better!**

**Stay Connected**

## 8. References

- <http://www.javatpoint.com/jsp-tutorial> (For JSP/Servlet)
- <https://dev.mysql.com/doc/workbench/en/wb-creating-eer-diagram.html> (For E-R Diagram Creation)
- <http://www.javatpoint.com/ajax-tutorial> (For AJAX Learning)
- <http://www.javatpoint.com/jquery-ui-tutorial> (For JQUERY)
- <http://stackoverflow.com/questions/19344977/jquery-ui-cant-get-datepicker-to-work> (JQUERY Data packer)
- <http://bootstrapdocs.com/v3.0.3/docs/css/> (For Bootstrap)
- <http://stackoverflow.com/questions/19924236/query-comparing-dates-in-sql> (SQL Dates)
- <http://stackoverflow.com/questions/2099425/when-should-we-use-a-preparedstatement-instead-of-a-statement> (Prepared Statement)
- <http://stackoverflow.com/questions/8263371/how-can-prepared-statements-protect-from-sql-injection-attacks> (For Prevent SQL Injection)
- <http://www.journaldev.com/2532/javamail-example-send-mail-in-java-smtp> (Java Mail Integration)