ONLINE MEDICAL ADVICE SERVICE

By

Syed Faisal Abbas

Muhammad Khaliq ur Rehman

Muhammad Raziullah Tiwana

Submitted to the Faculty of Computing Software Engineering, National University of Sciences and Technology, Islamabad in partial fulfillment for the requirement of a
B.E. Degree in Computer Software Engineering

May 2016

CERTIFICATE FOR CORRECTNESS AND APPROVAL

Certified that work contained in the thesis- **Online Medical Advice Service** carried out by Syed Faisal Abbas, Muhammad Raziullah Tiwana and Muhammad Khaliq ur Rehman under supervision of Dr. Hammad Afzal for partial fulfilment of Degree of Bachelor of Software Engineering is correct and approved.

Approved by

_____

(Dr. Hammad Afzal)

Department of Computer Software Engineering

Military College of Signals, Rawalpindi

Dated: _____

ABSTRACT

The world is moving rapidly towards online platforms, every basic need nowadays is available via the Internet. From online book stores to online food courts every business is making its way towards the Internet. Keeping that in mind the project “Online Medical Advice Service” or “O-MAS” is a step to move the medical assistance/services online so that they are accessible widely all over.

A Platform on which Patient will be able to ask a medical questions and using some tags they may be able to categorize the questions with the related questions and sitting on the other end Doctors and paramedical staff will answer the question of their respective field. The Questions will be entertained with respect to first come first serve strategy and another option will be incorporated i.e. if the patient does not wish to wait for the answer then he may also be able to pay to get an instant answer and he may also be entitled to a chat with the doctor. And if he is willing to pay sufficient enough he may also get a prescription. The patient will be able to sign up on the site and also will be able to ask a question and when that is the case the patient will provide his email on which his the notification will be send when the question is answered. After the question is answered the doctors will be rated the patient but that will be only allowed to the patient who have signed up.

DECLARATION

No portion of the work presented in this dissertation has been submitted in support of another award or qualification either at this institution or elsewhere.

DEDICATION

In the name of Allah, the Most Merciful, the Most Beneficent to our parents, without whose unflinching support and unstinting cooperation, a work of this magnitude would not have been possible.

ACKNOWLEDGEMENTS

To begin with, there is no greater guide than **ALLAH (SWT)** Himself and we feel blessed that He gave us enough strength to complete this project well in time.
In addition to this, we all would deeply and genuinely like to thank Sir **Hammad Afzal** for his persistent guidance and continuous support. Sir you are an exceptional supervisor and without you we could not have come this far.
.

Table of Contents

Chapter 1: Introduction **1**

Chapter 2: Literature Review **3**

Chapter 3: Software Requirement Specification 5

3. Introduction: 5

3.1.3.3 Project Supervisor: 6

3.1.3.4 Project Testers: 6

3.1.3.5 Up gradation Engineers: 6

3.1.3.6 End Users: 6

3.2 Overall Description 6

3.2.1 Product Perspective: 6

3.2.2 Product Functions: 7

3.2.4 Operating Environment: 7

3.2.4.1 Hardware requirements: 7

3.2.4.2 Software requirements: 8

3.2.5 Design and Implementation Constraints: 8

3.2.6 User Documentation: 8

3.2.7 Assumptions and Dependencies: 8

3.3 External Interface Requirements 8

3.3.1 User Interfaces: 8

3.4 Functional Requirements 9

3.5 Nonfunctional Requirements 10

Chapter 4: Design and Development 11

4.1 Introduction 11

4.2 System Architecture Description 11

4.2.1 Overview of Modules/Components 12

4.2.2 Structure and Relationships 13

4.2.2.1 System Block Diagram 13

4.2.2.2 Database Design 14

4.2.2.2.1 Database Design: 14

4.2.2.3 Sequence diagram 27
4.2.2.4 Activity Diagram 30
4.2.2.5 Implementation View (Class Diagram) 31
Chapter5:Testing…… 33
5.2 Test Items …… 33
5.3 Subsystems for test…… 33
5.4 FunctionalityTest…… 34
5.5 Pperformance Test …… 34

Chapter 1: Introduction

1.1. Overview:

The project aims to provide a platform on which patients and doctors can interact; which as a result will minimize the need of going to the doctor at a clinic physically. A doctor is virtually able to make an online presence and patients will be able to reach him due to this.

1.2. Problem Statement

The project intends to investigate the benefits of Online Advise service. Imagine you went to a doctors clinic and when you get there either there are no free slots for appointment or the doctor is not there yet or may have left .Now you are left with the option to go to another clinic or go home either way you have suffered loss in terms of time and money. The project intends to minimize the loss.

1.3. Scope of Work

The scope of work is limited to making a web site and an Android app which will incorporate the features discussed further in the document.

1.4. Objective:

I. ***Academic Objective*****:**

Web technologies will be utilized, Networking especially Peer to Peer will be applied in the case when an individual meeting is booked (which in fact will be only paid or if the specific Doctor agrees it to be free).

II. ***Application /End Goal Objective*****:**

To create a platform on which people will be able to get medical advice for free and also be able to chat with the Doctor but that would be charged and even get a prescription.

1.5. **Deliverables**

Sr.	Tasks	Deliverables
1	Literature Review	Literature Survey
2	Requirements Gathering	SRS Document
3	Application Design	Design Document (SDS)
4	Implementation	Implementation on computer with a live test to show the accuracy and ability of the project
5	Testing	Evaluation plan and test document
6	Deployment	Complete application along with necessary documentation

Table 1: Deliverables

Chapter 2: Literature Review

2.1. Doctors Lounge:

Provides physicians, students and allied professionals with clinical and technical information through multimedia tutorials and study aids

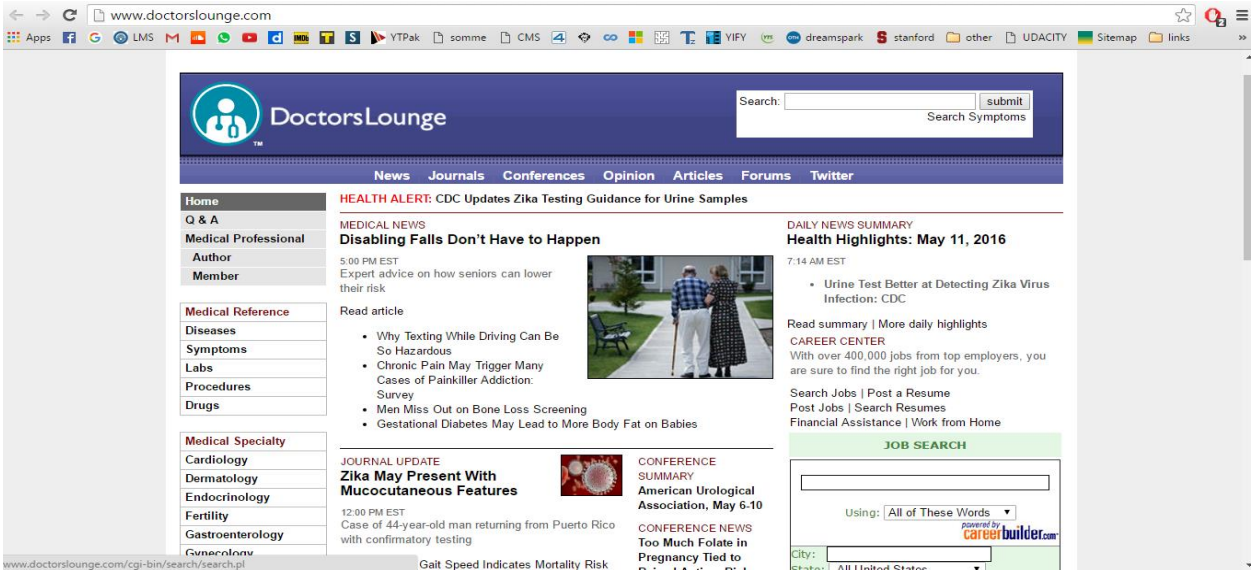


Figure 1: Docyors Lounge

2.2. Net Doctor:

A comprehensive guide to health on the Internet, with reviews and links to hundreds of online resources.

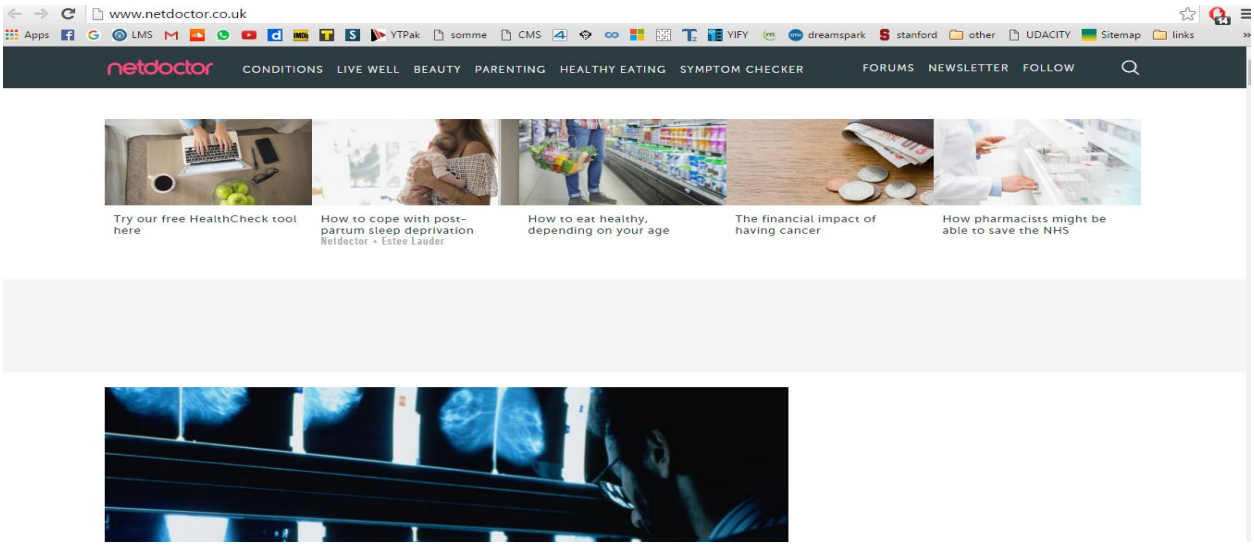


Figure 2 Net Doctor:

2.3. Doctor Spring:

Doctor Spring is an online platform where patients can consult licensed physicians' online - from anywhere, at any time. It help people in getting medical advice.

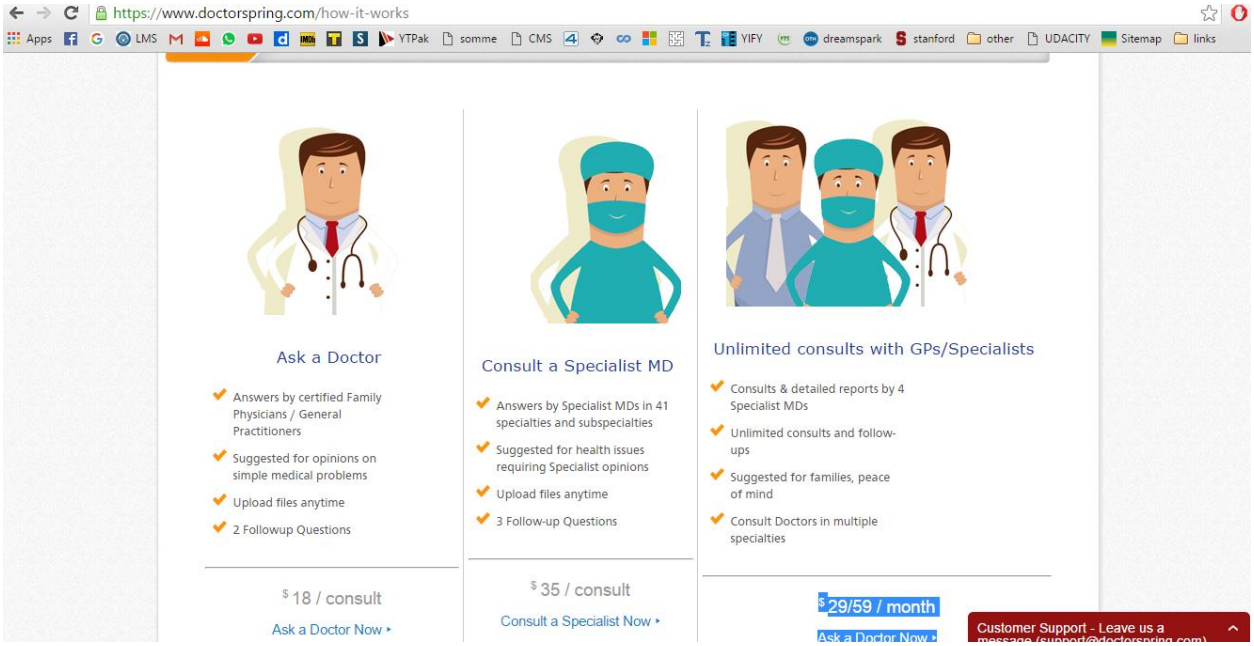


Figure 3: Doctors Spring

The main problem of these exiting sites is that:

- Most of them are paid not free and
- Even when some are free they don't provide a friendly interface.
- None of them are doctors from this country.

Chapter 3: Software Requirement Specification

3. Introduction:

This section covers the basic questions like what, why and how about the project.

3.1.1. Purpose:

The purpose of this document is to provide an overview of the requirements of O-MAS (Online Medical Advice Services). It will describe the purpose and requirements, functional and non-function both, of the system, the interfaces and how the system will react to external environment.

3.1.2 Document Conventions:

The conventions used to prepare the document is given bellow

1. Font – Times New Roman, size 12
2. Main headings, Bold size 18
3. Sub headings, Bold size 14
4. Sub-sub headings, Bold size 12

3.1.3 Intended Audience and Reading Suggestions:

This document is primarily intended for the evaluators and the supervisor of the project. This document will provide guidelines to system developers and testers. Any third party who needs a basic understanding of the system may find this document helpful.

3.1.3.1 Examiners/Evaluators:

The document will provide the FYP evaluators with the scope, requirements and details of the project to be built. It will also be used as basis for the evaluation of the implementation and final project.

3.1.3.2 Developers:

The document will provide guidance to the developers to determine what the requirements are and how they should continue with the project.

3.1.3.3 Project Supervisor:

This document will be used by the project supervisor to check whether all the requirements have been understood and in the end whether the requirements have been implemented properly and completely.

3.1.3.4 Project Testers:

Project testers can use this document as a base for their testing strategy as some bugs are easier to find using a requirements document. It will help in building up test cases for the testing process. This way testing becomes more methodically organized.

3.1.3.5 Up gradation Engineers:

Up gradation engineers can review projects capabilities and more easily understand where their efforts should be targeted to improve or add more features to it. It sets the guidelines for future developments.

3.1.3.6 End Users:

This document can be read by the end users if they wish to know what the project is about and what requirements have been fulfilled in this project.

3.1.4 Product Scope:

The aim of the product is to assist patients and connect them with renowned doctors and physicians who review their medical condition, answer their questions and provide a detailed report. The product is mainly a web product with a mobile application in Android.

3.2 Overall Description

3.2.1 Product Perspective:

The product is developed to help the patients get a quick access to medical facilities and get treatment from the physicians. The product is basically an interface between the patients and doctors.

It has 2 versions; a free version and a paid version. The free version would provide basic facilities to the users while the paid version would provide basic facilities plus some more functionality. They will be discussed later in the document.

3.2.2 Product Functions:

A patient could use 3 ways to get the treatment; he could post a question about his problem and get it answered (free version), he could chat with the doctor (paid version) or he could have a video chat with the doctor (paid version).

A patient could also upload an image of his disease in the question answer forum (free version). He could also ask for appointments with the doctor (paid version). Every doctor would have a profile in which his degree, his experience and other relevant information would be mentioned. In addition to that each doctor would have a rating, which the patients will provide, which will show the doctor's capability. The product will also mention the cases the doctor has attended to and its results. Similarly a patient would also have a profile in which his medical history would be mentioned which will only be available for the doctors and him in order to keep privacy. Doctors could also provide prescriptions for the ailment (paid version).

3.2.3 User Classes and Characteristics:

The intended user will be a member of general public who is interested in getting the medical treatment. As mentioned earlier a user could get treatment via 3 ways. Public user can view the doctor's profile. Paid users will have privileged access to the facilities that the free user would not have. Users may be technical or non-technical.

3.2.3.1 Tester

Tester will also use this project to check for bug finding. They will also use this to check it is in accordance to the SRS document.

3.2.3.2 Project Evaluator/Supervisor

Project supervisor/Evaluator will also use the product to evaluate. They will use this product to find the accuracy and error in the output.

3.2.4 Operating Environment:

The operating environment required for this project is:

3.2.4.1 Hardware requirements:

The product is a mainly a web application with a mobile application in Android. It requires an always online server, which will manage all the communications between the users. Further hardware requirements will be added in the later versions as the project progresses.

3.2.4.2 Software requirements:

The software required for the product are ASP.NET (web app) and Android (mobile app). Other than that the database used will be SQL Server. Operating System required will be Windows or MAC, Linux etc.

3.2.5 Design and Implementation Constraints:

User will not have an access to other users' profiles in order to keep privacy. The conversation between a doctor and user will only be between them unless a user wants to share it with any other doctor but the doctor's name will not be provided with whom the conversation was made.

3.2.6 User Documentation:

For the user documentation, a user manual will be provided with the system. It will include the details of the system's working. Help documents will also be a part of the system. The project report will also be available for the users which will highlight the system features, working and procedures.

3.2.7 Assumptions and Dependencies:

An assumption is made that the doctor would be able to provide a prescription online.

3.3 External Interface Requirements

3.3.1 User Interfaces:

The interfaces for the product are user friendly for both the web app and mobile app. They are simple and easy to understand. Controls which allow the user to interact with the application are clear and imply their functionality within the application. Error notifications are incorporated within the application, presenting the user with appropriate messages which describe the error that has taken place. If applicable, error messages should suggest possible solutions to the problem.

3.4 Functional Requirements

System features are discussed in detail. This heading mainly covers the functional requirements of the project.

1. Every user (patient / doctor) should first make an account which will further make his profile.
2. Every doctor must have an authentication key which will be given to him after the verification of his degrees.
3. Every user must have a unique ID and password, later he must update his profile with the required information.
4. The System should be able to notify invalid username and password if it is not found in database.
5. If the invalid username or invalid password is entered, the system should generate an error message.
6. The patient's profile will have his related information along with the medical history.
7. The doctor's profile will have his related information along with their medical degrees and certificates, ratings, clinic/hospital location etc.
8. The patients will also provide their payment methods while making their account for paid version.
9. The standard payment methods will be used.
10. After login the patient can choose his problem from the available categories.
11. The list of specialist doctors, related to the problem (e.g. ENT specialists if the problem is related to Throat), will then be shown to the patient and he can choose to take appointments.
12. In free version of the software the patient can only post questions about his problem.
13. The doctors will answer these questions from their account.
14. If the question is already posted by someone else, the system can suggest the answers that are provided to the similar questions. The patient can choose from the available answers or he can continue to post a fresh question.
15. When the question is answered a notification must be sent to the patient.
16. The patient will be able to interact with the doctor when the appointment is started.
17. The patient can chat or video call the doctor for consultation.

18. The patient will get prescriptions from the doctor.
19. The doctor can reschedule his time table.
20. If the doctor cancels any appointment an alert/ notification must be sent to the patient about the cancellation.

3.5 Nonfunctional Requirements

Non-Functional Requirements are very critical in success of a project. This section enlists the non-functional requirements for our project.

4.5.1. Reliability

The software will be managed regularly and meet all of the functional requirements without any unexpected behavior. The system will not crash or malfunction. It will be managed so that it is known to be reliable.

4.5.2. Availability

The software will be available at all times on each device as long as the server is working properly. The availability will depend upon the external services such as internet access and condition of the user device. If those services are unavailable, the user should be alerted.

4.5.3. Security

User's medical history and data will not be shared with the public. Only his name and picture will be shared.

4.5.4. Maintainability

The software should be written clearly and concisely. The code will be well documented. Particular care will be taken to design the software modularly to ensure that maintenance is easy.

4.5.5. Portability

The application is a multi-platform application that can be accessed from anywhere the internet is available.

Chapter 4: Design and Development

4.1 INTRODUCTION

The world is moving rapidly towards online platforms, every basic need nowadays is available via the Internet. From online book stores to online food courts every business is making its way towards the Internet. Keeping that in mind the project "Online Medical Advice Service" or "O-MAS" is a step to move the medical assistance/services online so that they are accessible widely all over.

4.2 SYSTEM ARCHITECTURE DESCRIPTION

1. **Overview of modules / components** shows the main component of the application and their interrelationships
2. **Structure and Relationships** shows the higher level details of system working by the means of system block diagram, activity diagram, state transition diagram and use case diagram and lower level details by the means of class diagram, chen's entity relationship diagram, sequence diagram and structure chart
3. **User Interface Issues** shows the basic design of the user interfaces of the application and possible issues related with them

4. **Detailed description of components** show the working of modules with very low level details. It shows the purpose, function, subordinates, dependencies, interfaces, resources, processing and dat2a of the components and their relationships with each other.
5. **Reuse and relationships to other products** shows information about work done in the same project before and any reuse of the same work.
6. **Design decisions and trade off** shows the architecture style and design pattern of the application
7. **Pseudo Code for components** shows the pseudo codes for the components of the system.

4.2.1 OVERVIEW OF MODULES/COMPONENTS

O-MAS is made in ASP.NET MVC 5 architecture. It comprises of following modules:

i. View (Application UI)

- Patient UI
- Doctor UI
- Home UI

ii. Control (Data Control)

- Home Controller
- Patient Controller
- Doctor Controller
- OMAS_Chat_Hub
- OMAS_News_Feed_Hub

iii. Model (Data Storage)

- OMAS_DB

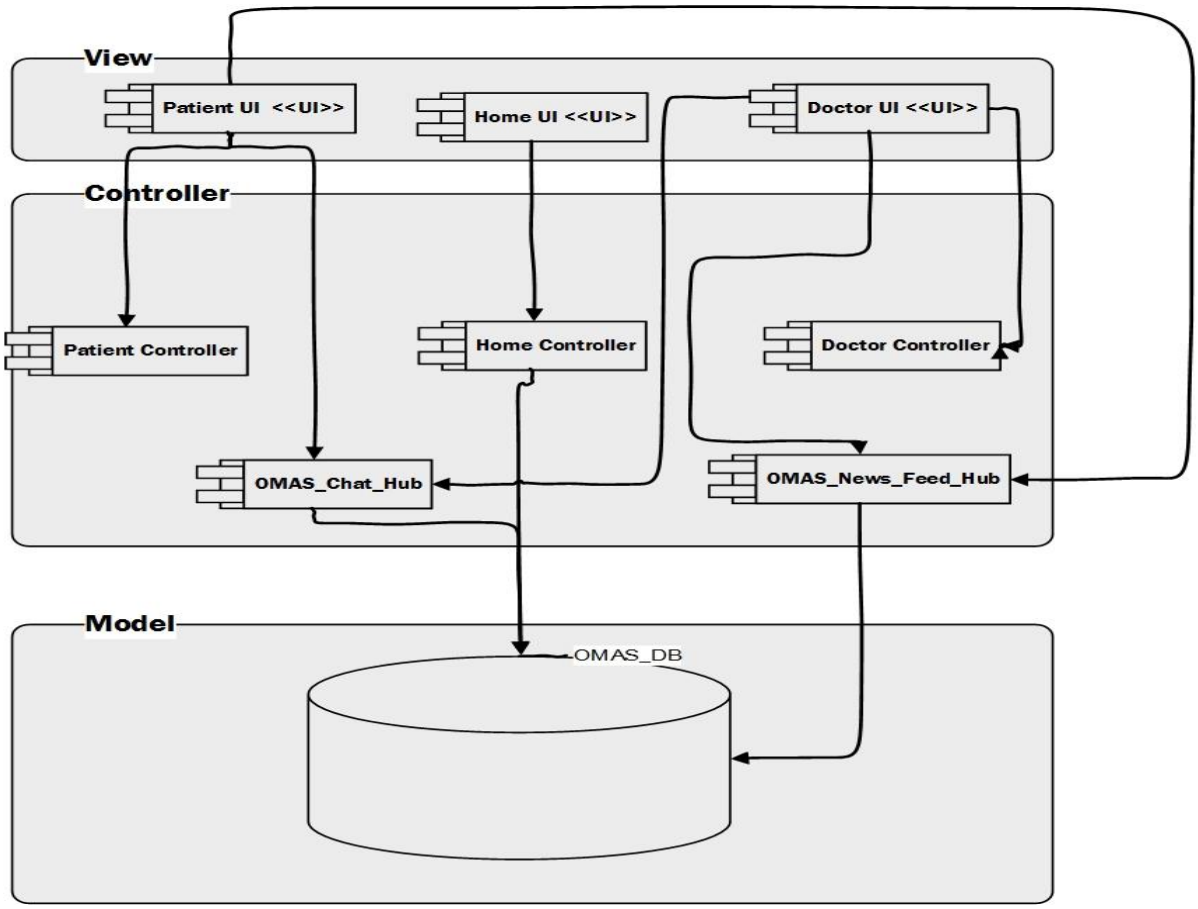


Figure 4: *Block diagram for FTt*

4.2.2 STRUCTURE AND RELATIONSHIPS

This section will cover the overall technical description of **O-MAS** which will show the working of website in perspective of different point of views and will show the relationships between different components.

4.2.2.1 System Block Diagram

This diagram shows the higher level description of the application. It shows the generic working of the Web application and interaction with the user.

In this web application, User will interact with the UI of the application; UI will interact with the Controller which will then interact with Model.

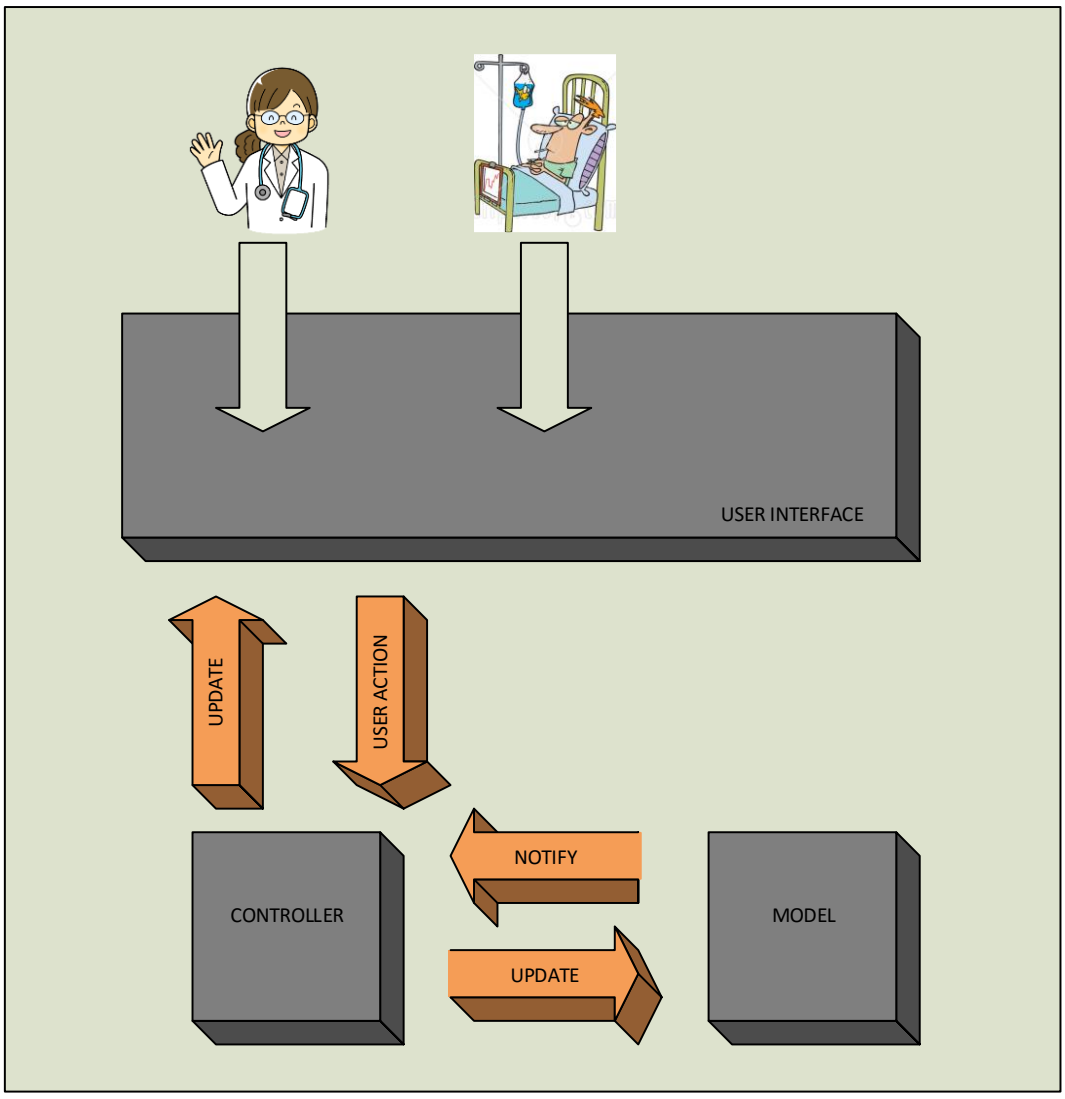


Figure 5: System Block Diagram

4.2.2.2 DATABASE DESIGN

O-MAS require user's username and password, personal information and his respective progress along with other data related to him e.g questions posted, appointments etc to be saved in database. There are eight different tables in database structure which will manage all the requirements mentioned above. Following are the tables: Doctors, Patients, Question, Answers, Follow, SignUpT, Appointments and Chat_Messages.

4.2.2.2.1 DATABASE DESIGN:

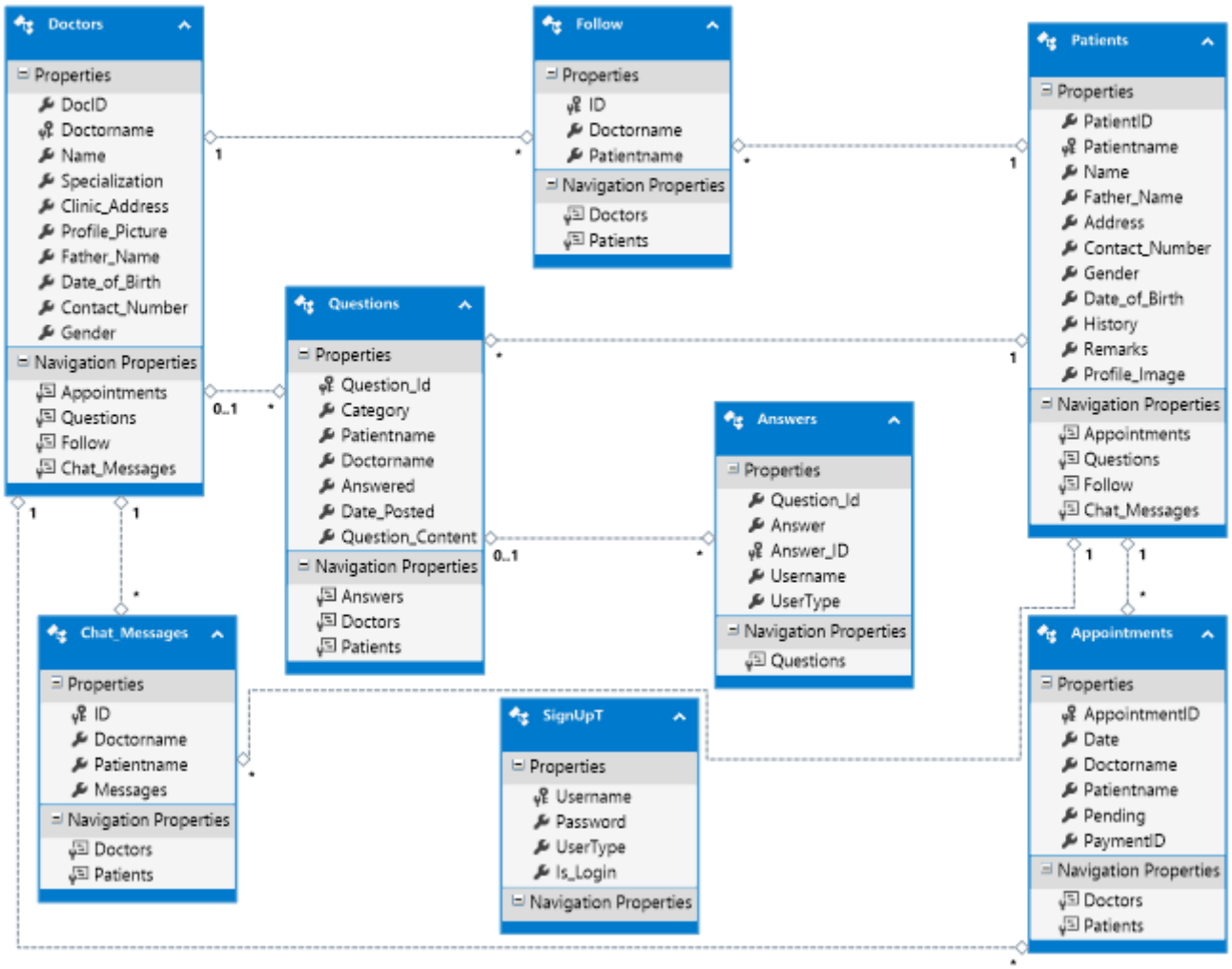


Figure 6: Database Diagram

4.2.2.3 Use Case Diagrams

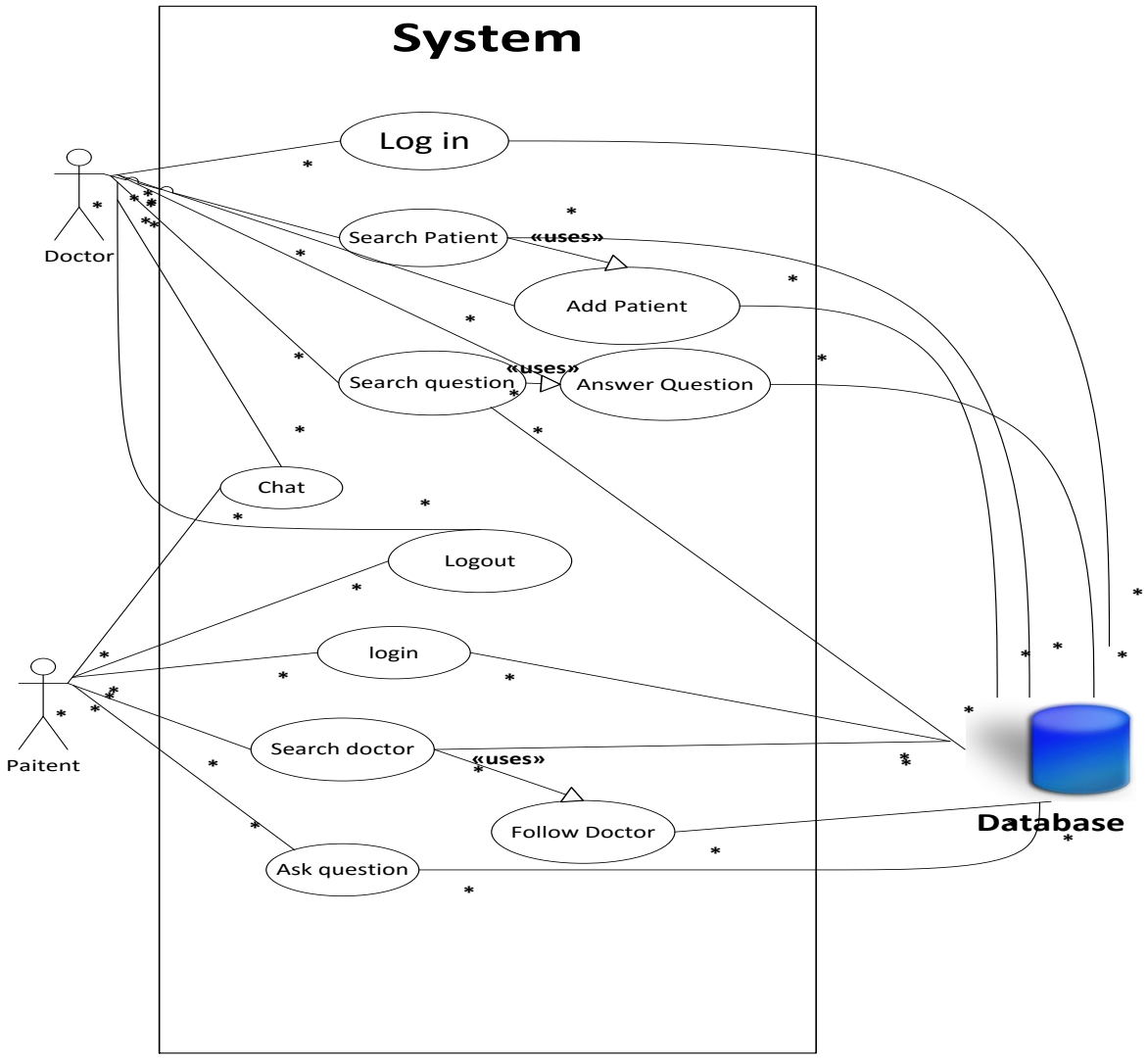


Figure 7: UML Use Case Diagram

Actors:

Primary Actors: Doctor, Patient

Secondary Actors: Database

Use Cases

A. Doctor

- Login
- Search Patient
- Add Patient
- Search question
- Answer question
- Chat
- Logout

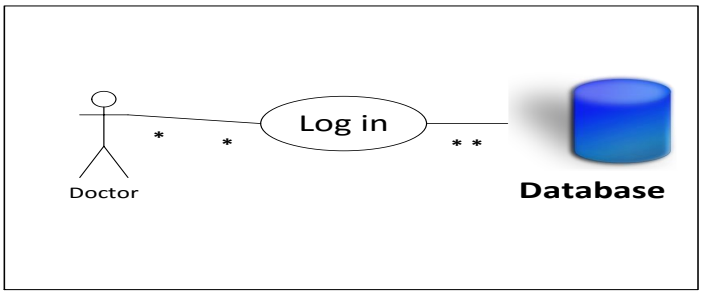
B. Patient

- Login
- Search Doctor
- Follow Doctor
- Ask Question
- Chat
- Logout

Use Case Description

Use Case 1:

Figure 8: Login

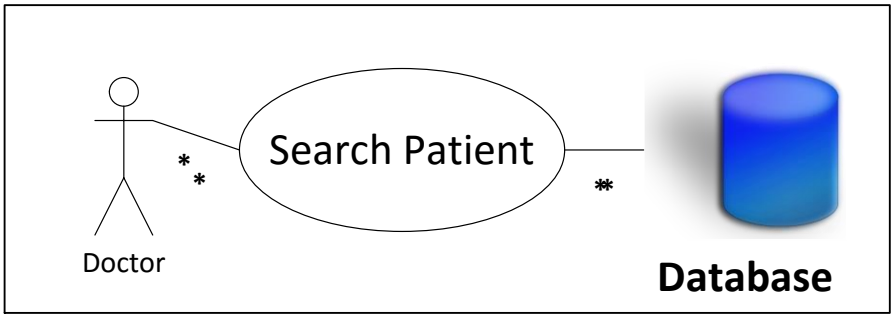


USE CASE NAME	Login
PRIMARY ACTOR	Doctor
SECONDARY ACTOR	System/Database
NORMAL COURSE	• Doctor enters his username in the required field to log in to the application. • Doctor than enters his password. • Doctor than presses the login button to enter the application.
ALTERNATE COURSE	• If the doctor provides incorrect username or password the application shows an error message and login will not be successful.
PRE CONDITION	Username and password of the doctor must be already registered at the time of coding.
POST CONDITION	The doctor successfully logins the system.
EXTENDS	NA
INCLUDE	NA
ASSUMPTIONS	The doctor provides correct information.

Table 2: Login

Use Case 2:

Figure 9: Search Patient

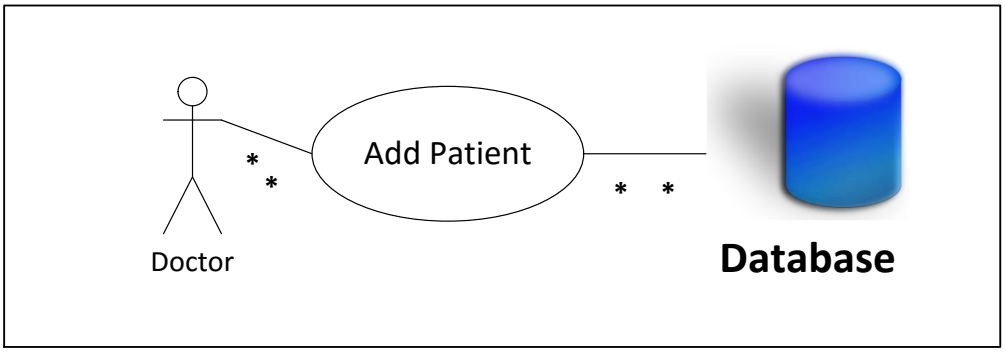


USE CASE NAME	Search Patient
PRIMARY ACTOR	Doctor
SECONDARY ACTOR	System/Database
NORMAL COURSE	Doctor searches the patient by entering his ID in the database, or scrolling through the list of available patients.
ALTERNATE COURSE	If the doctor has entered a wrong ID of the patient then the database will show an error message "The entry doesn't exist".
PRE CONDITION	The doctor must be logged into the system in order to search the patient
POST CONDITION	The doctor searches the patient easily
EXTENDS	NA
INCLUDE	NA
ASSUMPTIONS	The doctor provides with correct id of the patient.

Table 3: Search Patient

Use Case 3:

Figure 10: Add Patient

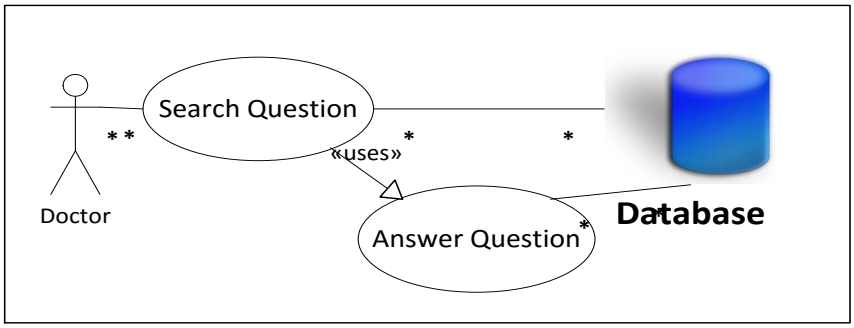


USE CASE NAME	Add Patient
PRIMARY ACTOR	Doctor
SECONDARY ACTOR	System/Database
NORMAL COURSE	• Doctor adds the patient by clicking the add button on the patient profile
ALTERNATE COURSE	N/A
PRE CONDITION	The doctor has successfully logged into the application and there is no duplication of data.
POST CONDITION	A notification will be triggered to the added patient
EXTENDS	NA
INCLUDE	NA
ASSUMPTIONS	The doctor fills all the required fields and provides correct information.

Table 4: Add Patient

Use Case 4:

Figure 11: Search Question

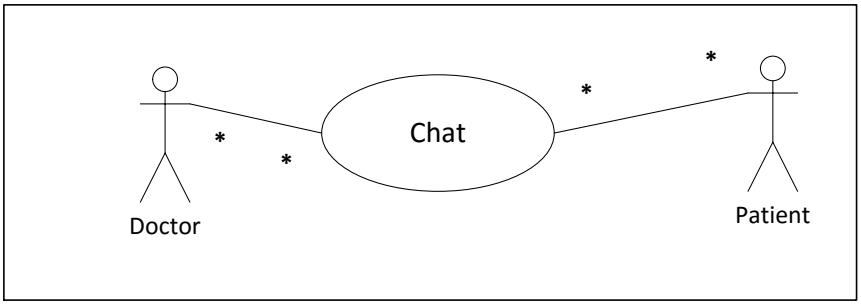


USE CASE NAME	Search Question
PRIMARY ACTOR	Doctor
SECONDARY ACTOR	System/Database
NORMAL COURSE	• Doctor enter the query and search relevant question • Question are displayed according to the priority • Doctor click on the question to answer them
ALTERNATE COURSE	No relevant question is found
PRE CONDITION	The doctor has successfully logged into the application and there is no duplication of data.
POST CONDITION	A notification will be triggered to the patient who has asked the question
EXTENDS	NA
INCLUDE	Answer Question
ASSUMPTIONS	NA

Table 5: Search Question

Use Case 5:

Figure 12:Chat

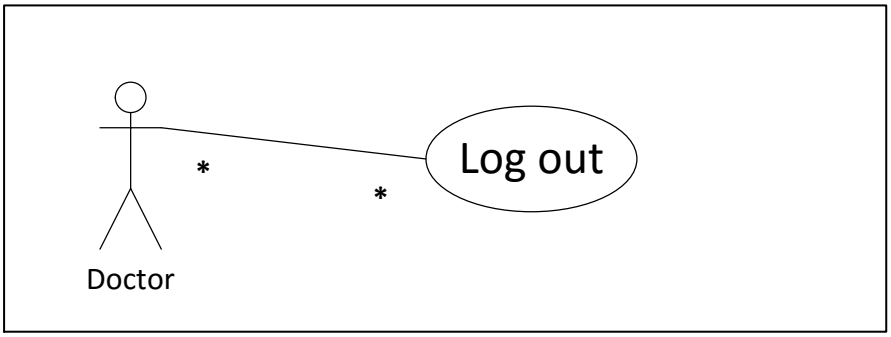


USE CASE NAME	Chat
PRIMARY ACTOR	Doctor
SECONDARY ACTOR	Patient
NORMAL COURSE	• Doctor select the patient • Doctor send message or call or video call according to appointment. • Doctor and patient talk
ALTERNATE COURSE	No relevant question is found
PRE CONDITION	The doctor has successfully logged into the application and there is no duplication of data.
POST CONDITION	Chat will be add to the history
EXTENDS	NA
INCLUDE	NA
ASSUMPTIONS	Both are online at the same time

Table 6: Chat

Use Case 6:

Figure 13: Logout

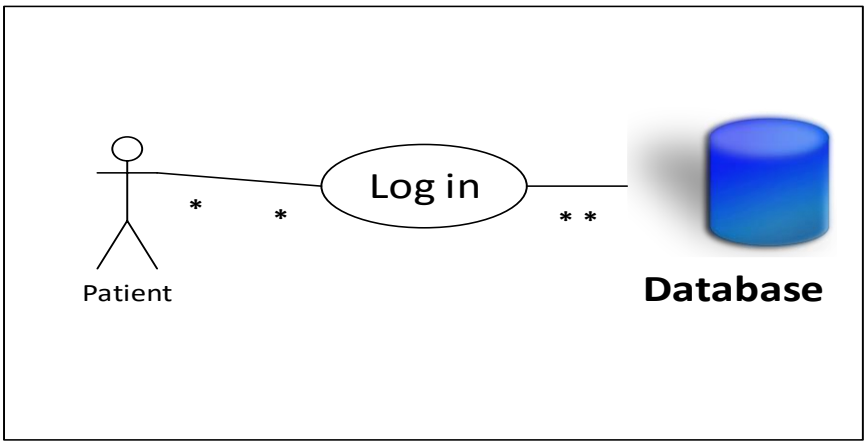


USE CASE NAME	Logout
ACTOR	Doctor
NORMAL COURSE	Doctor selects the logout option available at different screens and applications will successfully logout.
ALTERNATE COURSE	The application will not successfully logout and generates error message if the user is still performing any task.
PRE CONDITION	Doctor has finished the task he was performing before logging out of application.
POST CONDITION	Application successfully logs out.
EXTENDS	NA
INCLUDE	NA
ASSUMPTIONS	Doctor is done with all of the tasks and then logs out of the application.

Table 7: Logout

Use Case 7:

Figure 14: Login

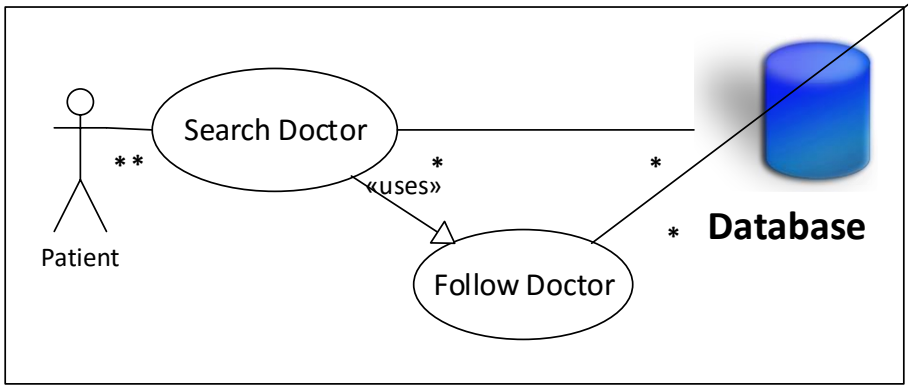


USE CASE NAME	Login
PRIMARY ACTOR	Patient
SECONDARY ACTOR	System/Database
NORMAL COURSE	• Patient enters his username in the required field to log in to the application. • Patient than enters his password. • Patient than presses the login button to enter the application.
ALTERNATE COURSE	• If the Patient provides incorrect username or password the application shows an error message and login will not be successful.
PRE CONDITION	Username and password of the doctor must be already registered at the time of coding.
POST CONDITION	The Patient successfully logins the system.
EXTENDS	NA
INCLUDE	NA
ASSUMPTIONS	The doctor provides correct information.

Table 8: Login

Use Case 8 :

Figure 15: Search Doctor

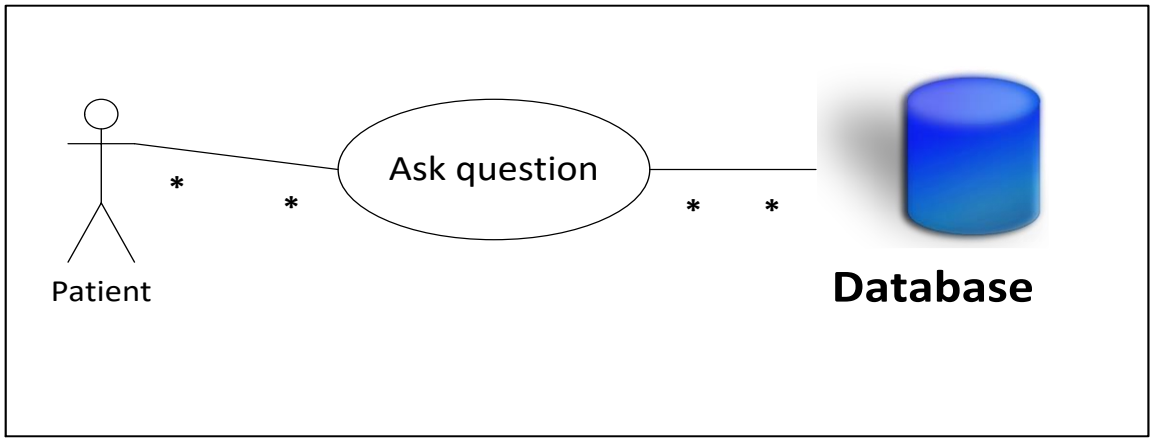


USE CASE NAME	Search Doctor
PRIMARY ACTOR	Patient
SECONDARY ACTOR	System/Database
NORMAL COURSE	Patient searches the patient by entering his ID in the database, or scrolling through the list of available patients.
ALTERNATE COURSE	If the Patient has entered a wrong ID of the Doctor then the database will show an error message "The entry doesn't exist".
PRE CONDITION	The Patient must be logged into the system in order to search the Doctor
POST CONDITION	The doctor searches the patient easily
EXTENDS	NA
INCLUDE	NA
ASSUMPTIONS	The doctor provides with correct id of the patient.

Table 9: Search Doctor

Use Case 9:

Figure 16: Ask Question

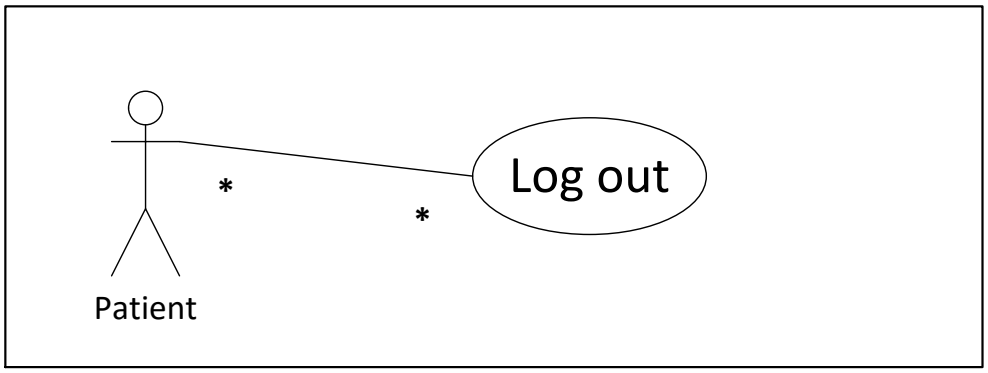


USE CASE NAME	Ask Question
PRIMARY ACTOR	Patient
SECONDARY ACTOR	System/Database
NORMAL COURSE	• Patient selects the ask question button. • He write his question, select he category • Confirm to post the question
ALTERNATE COURSE	• If the question is a repeated one , a notification is triggered and the patient confirm weather he wants to view the already asked question or ask another one.
PRE CONDITION	Username and password of the patient must be already registered at the time of coding.
POST CONDITION	The Question is added to the queue
EXTENDS	NA
INCLUDE	NA
ASSUMPTIONS	The patient is logged in

Table 10: Ask Question

Use Case 10:

Figure 17: Logout



USE CASE NAME	Logout
ACTOR	Patient
NORMAL COURSE	Patient selects the logout option available at different screens and applications will successfully logout.
ALTERNATE COURSE	The application will not successfully logout and generates error message if the user is still performing any task.
PRE CONDITION	Patient has finished the task he was performing before logging out of application.
POST CONDITION	Application successfully logs out.
EXTENDS	NA
INCLUDE	NA
ASSUMPTIONS	Patient is done with all of the tasks and then logs out of the application.

Table 11: Logout

4.2.2.3 Sequence diagram

Sequence Diagrams of key use cases are mentioned below:

4.2.2.3.1. Login Sequence Diagram

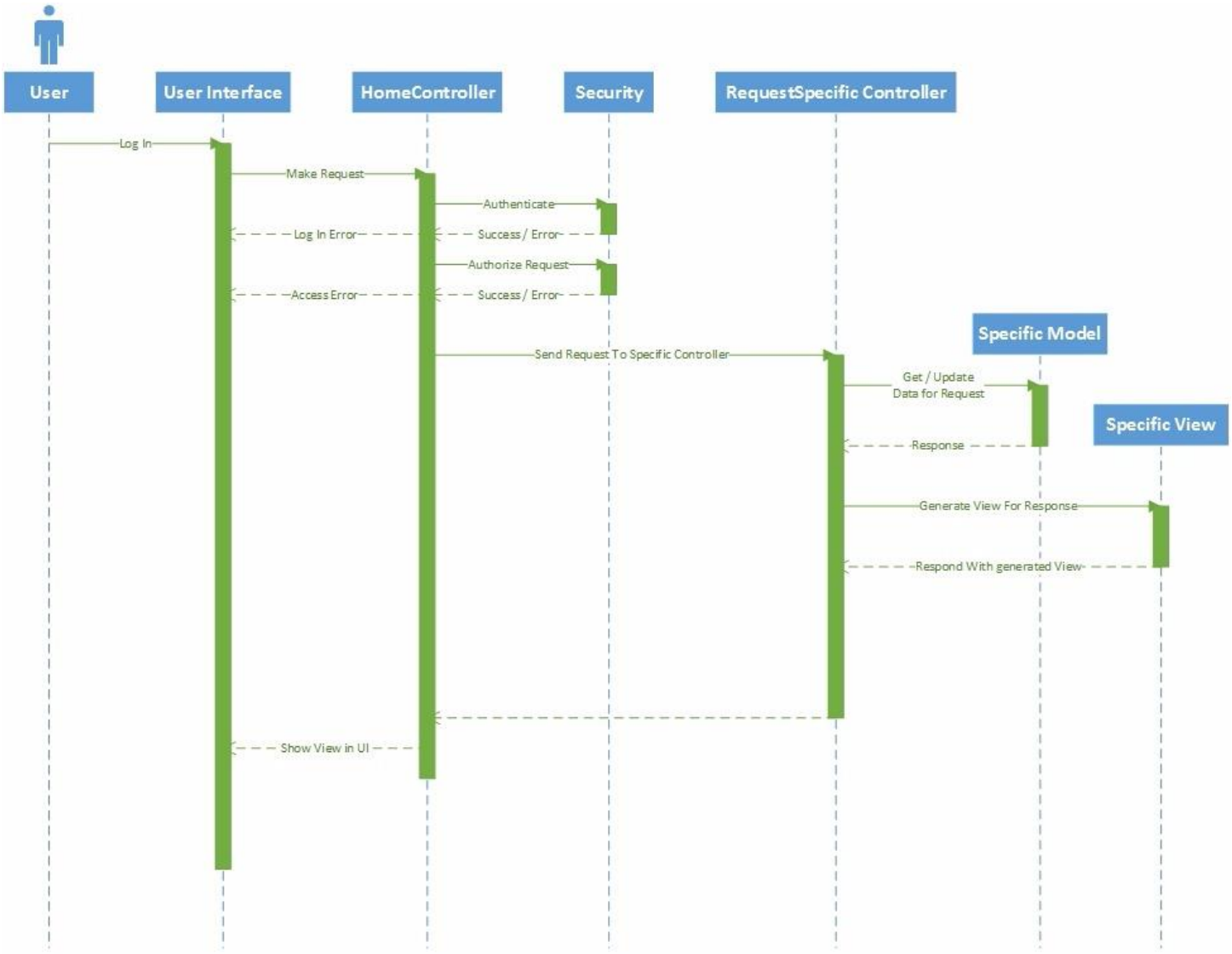


Figure 18: Login

4.2.2.3.2. Post a Question Sequence Diagram

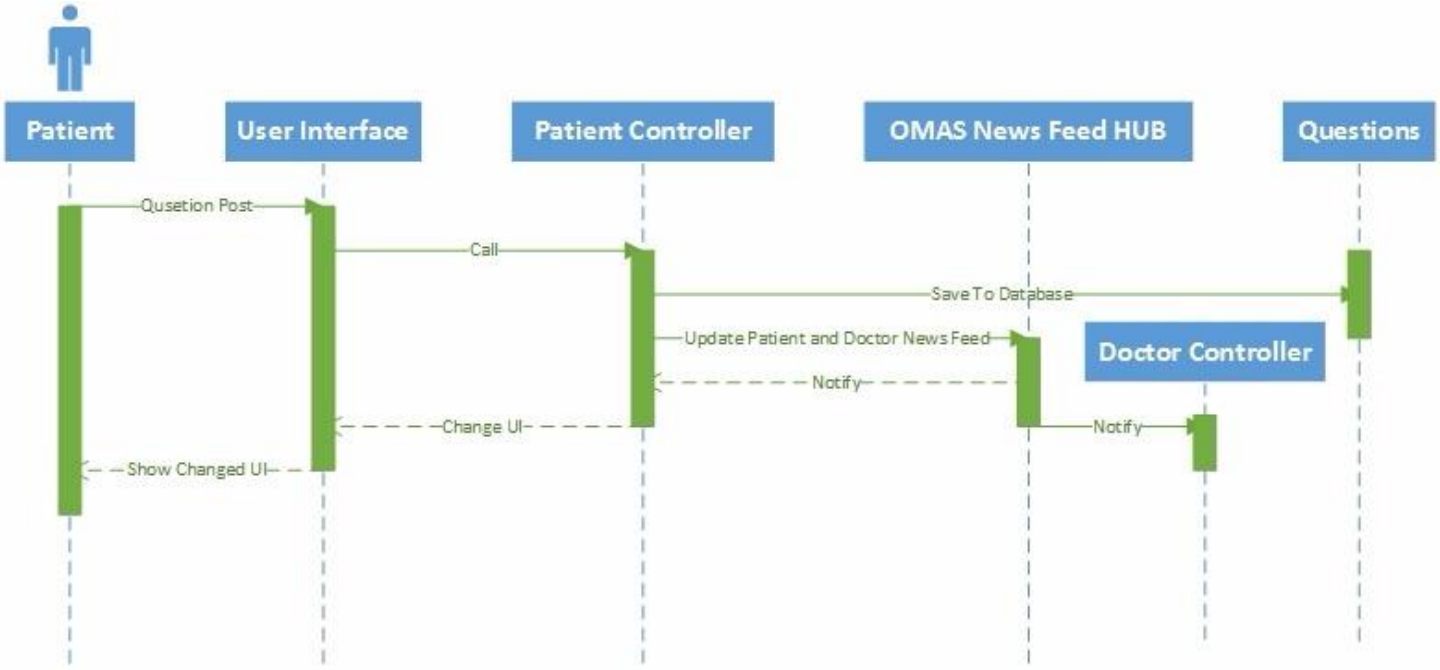


Figure 19: Post a Question

4.2.2.3.3.Answer a Question Sequence Diagram

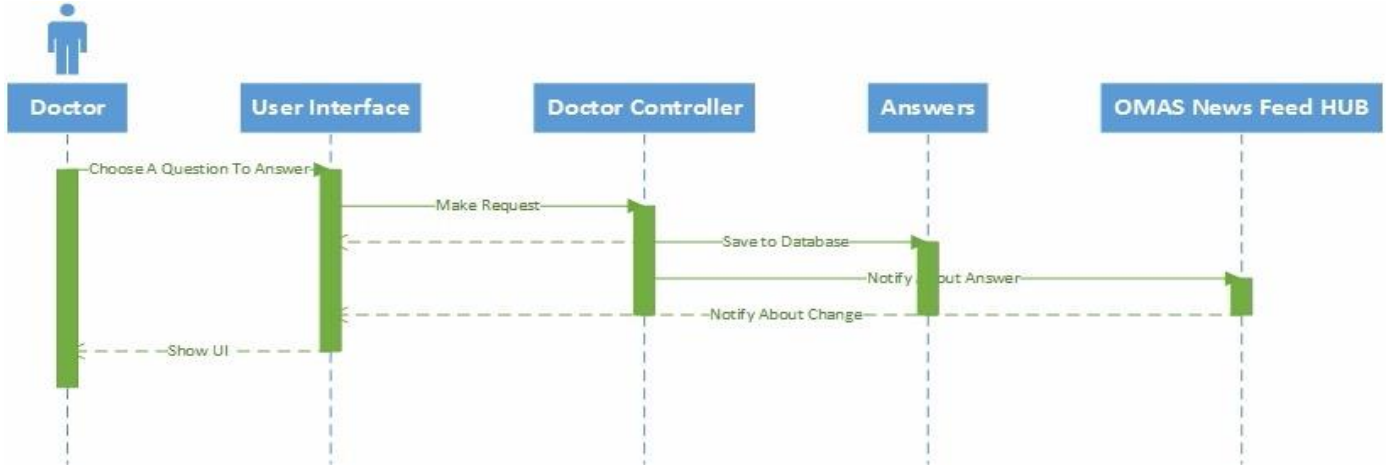


Figure 20: Ask a Question

4.2.2.3.4 Follow a Doctor Sequence Diagram

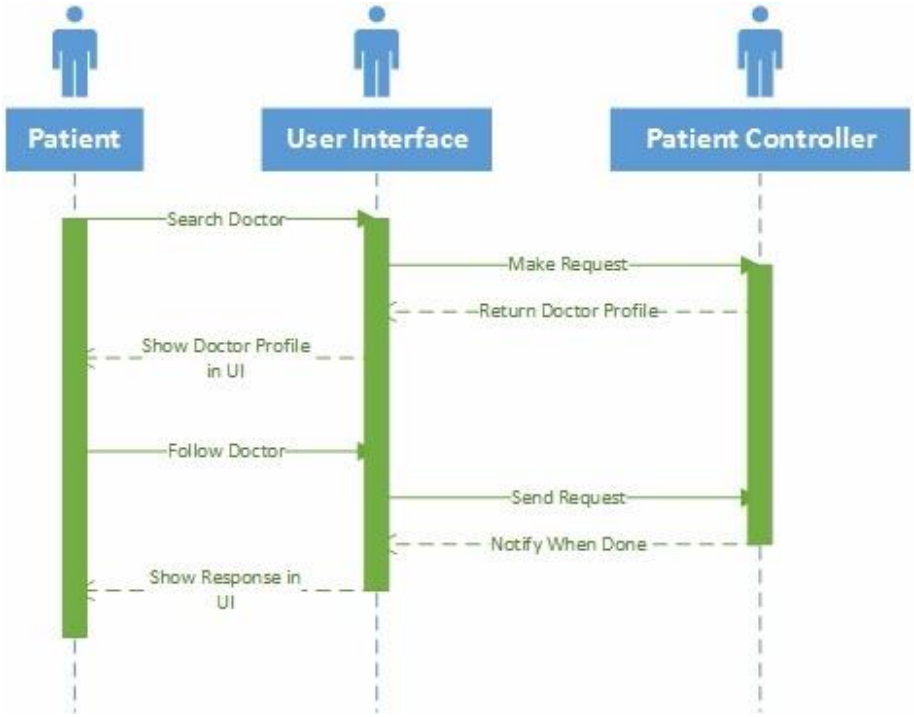


Figure 21: Follow a Doctor

4.2.2.3.5. Chat Sequence Diagram

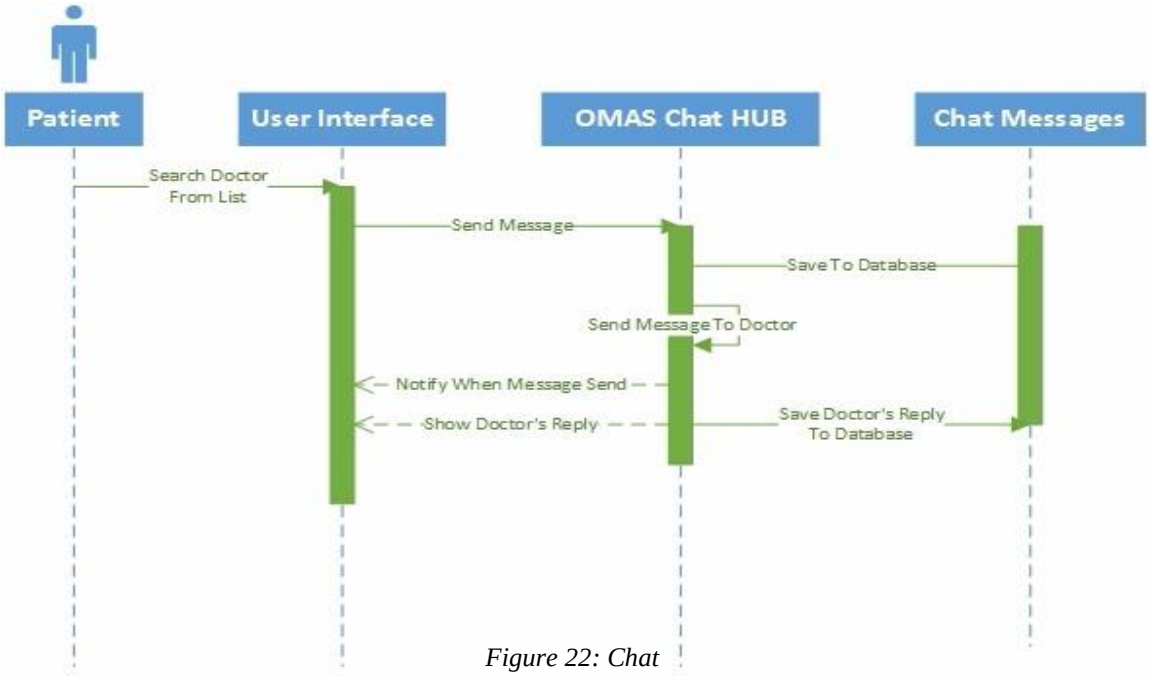


Figure 22: Chat

4.2.2.4 Activity Diagram

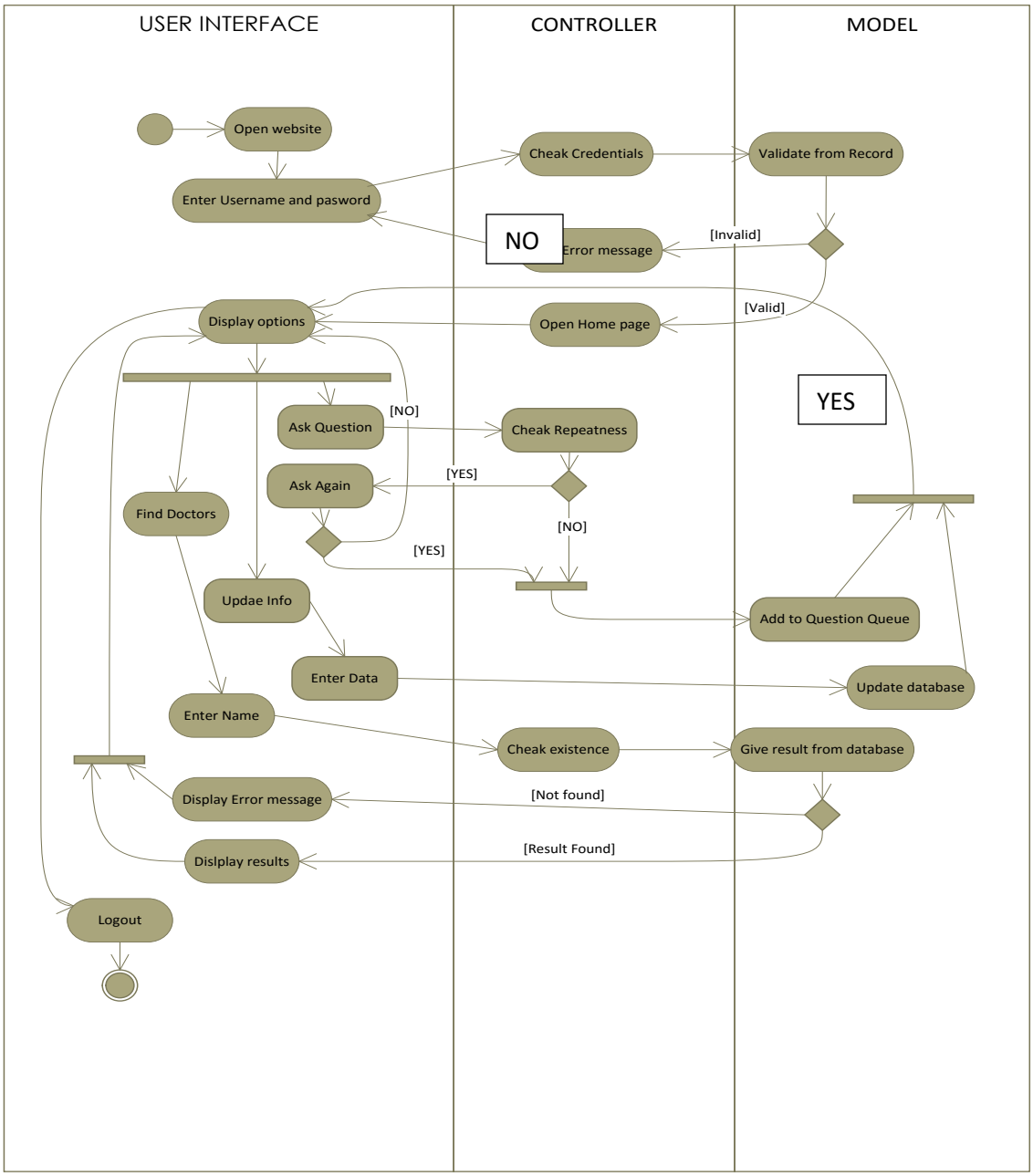


Figure 23: Activity

In activity diagram, the dynamic view of the system is shown. All the activities are shown concurrently and their respective start and end states are shown.

4.2.2.5 Implementation View (Class Diagram)

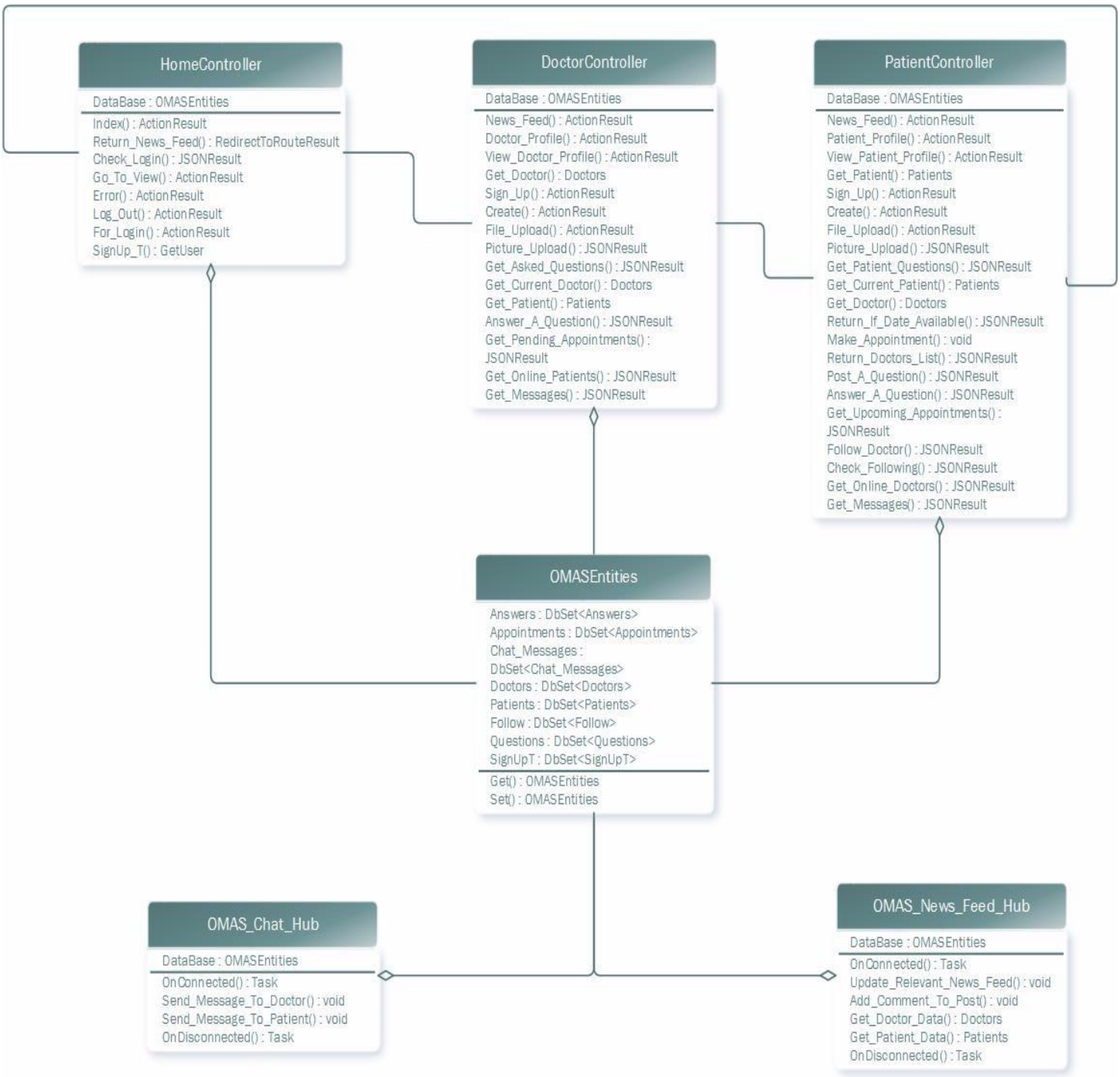


Figure 24: Class Diagram

Classes	Description
Home Controller	One of the 3 controller classes of the system. It manages all the communication between the view and model when the user is not identified as a doctor or patient. It manages functionalities like login, showing start up page etc,
Doctor Controller	The Controller class that manages the functionalities of a doctor view. For example answering a question, checking appointments, taking appointments etc
Patient Cntroller	The Controller class that manages the functionalities of a doctor view. For example posting a question, taking appointments, following doctor etc
OMAS_Chat_Hub	This class controls the chatting between a patient and a doctor.
OMAS_NewsFeed_Hub	This class manages the news feed of the patients and doctors.
OMAS_Entities	The Model of the Database as a class which facilitates using the whole database as a class.

Table 12:Class Diagram

Chapter 5: Testing and Results

This test plan aims to cover Online Medical Advice Service (OMAS). To make sure that all newly added features of OMAS portal are working correctly, this test plan has been created. This test plan shall ensure that all features testing of this system is performed accurately. The 1st version of this test plan will cover test scenarios, expected results and acceptance criteria of the product.

5.1. Introduction

This document sets forth a framework for software testing of OMAS. OMAS aims to provide numerous features. The test plan is designed to make sure that all features are working as required. Following are some main features of the OMAS:

- Home page
- Doctor profile
- Patient profile
- Reports/History
- Prescription
- Maps

This document is a guide to design test cases and documentation. We shall apply black box and white box both testing techniques to make sure we have tested 100% of the code. These techniques will enable us to ensure that validation has been performed properly for each module w.r.t requirement specification.

5.2. Test Items

This section will provide test items for unit testing. Black box testing will be performed on Subsystem interface. External interface for the following browsers will be tested:

- Firefox
- Opera
- Internet Explorer
- Google chrome
- Safari

5.3. Sub-systems for Test

- Manage Profile
- Display Appointments
- Manage Questions
- Manage Chat

5.4 Functionality Test:

- Sign up
- Sign in
- Upload picture
- Follow Doctor
- Un follow Doctor
- Search Doctors
- Error messages
- Post Question
- Post answer
- Post comment
- Make appointments
- vote

5.5. Performance tests

Response Time for:

- Login 3-4 second
- Sign up 8-15second (depending on the image size)
- Logout < 1 second
- Search 1-2 second
- Question 3-7 second

Test case #	Features	Description	Preconditions
1	Sign up	Add profile information such as DOB, GENDER, NAME, CNIC etc. [for faculty and students]	The user must press the sign up button

Table 13: SignUp

Test case #	Features	Input values/Valid Inputs	Expected output	Results
1	Sign up	Strings	Sign up will be successful	Signup is Successful

Table 14:Signup Results

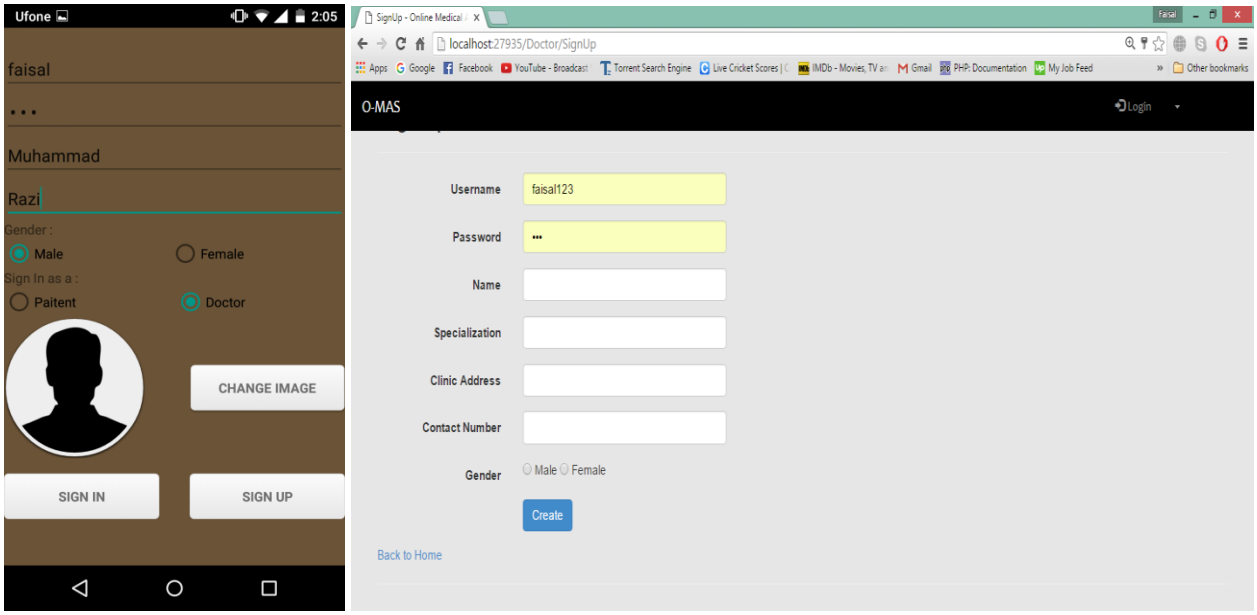


Figure 25: SignUp

Test case #	Features	Description	Preconditions
2	Upload picture	Profile pictures of Doctors and Patients	The user must have logged in

Test case #	Features	Input values/Valid Inputs	Expected output	Results
2	Upload picture	Image file [png, jpeg]	Image will be uploaded and linked to the profile	The picture is uploaded

Table 15: Upload Pic

Test case #	Features	Description	Preconditions
3	Follow Doctor	Check whether Patients can follow doctor	The Patient must have logged in

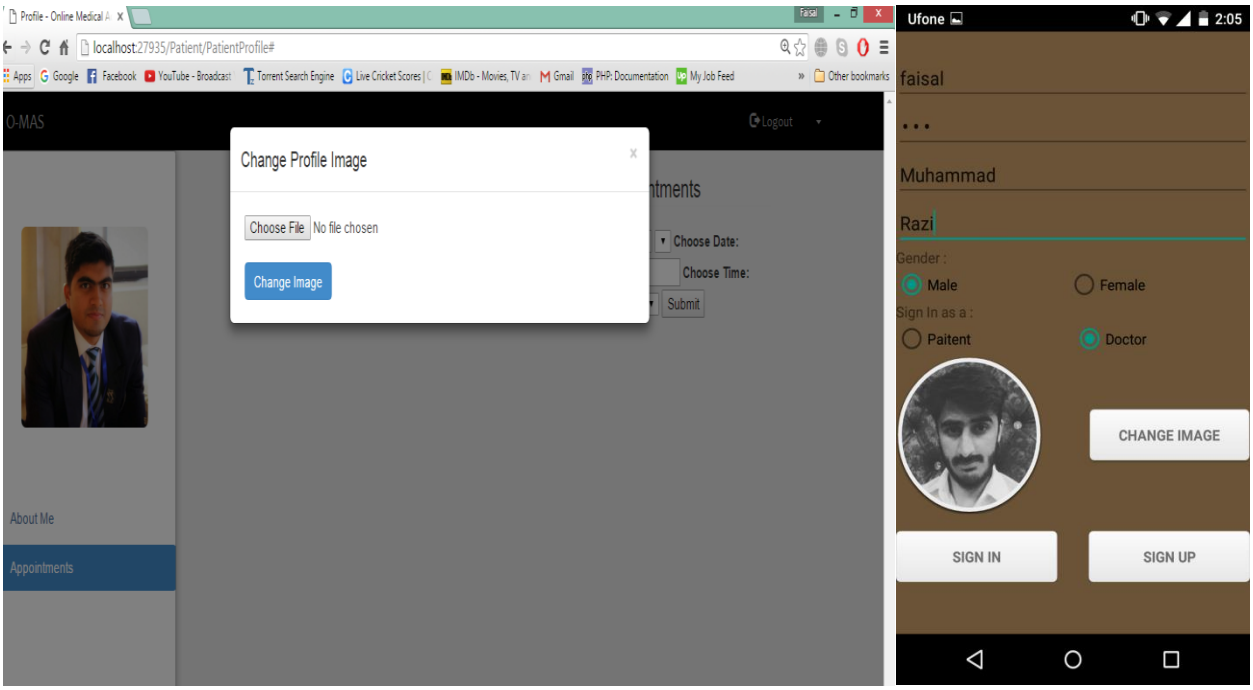


Figure 26: Upload Picture

Test case #	Features	Input values/Valid Inputs	Expected output	Results
3	Follow Doctor	NA	The doctor will be linked to the patient	The patient is successfully linked to the doctor

Table 16: Follow Doctor

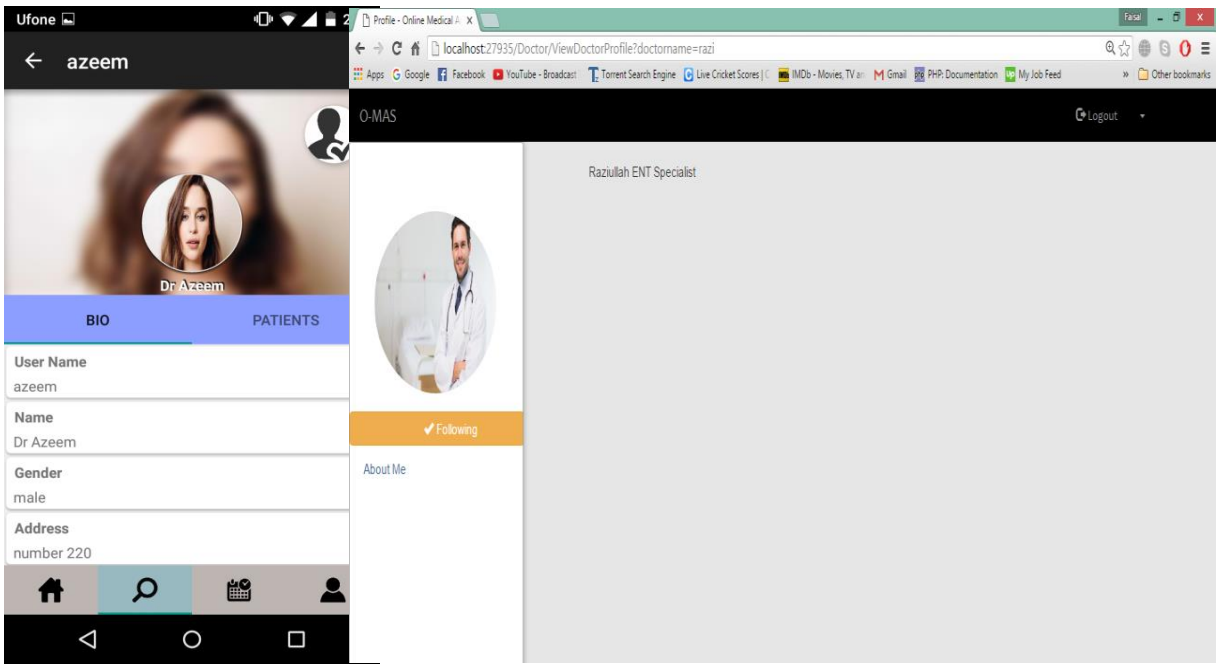


Figure 27:Follow

Test case #	Features	Description	Preconditions
4	Un Follow Doctor	Check whether Patients can follow doctor	The Patient must have logged in

Test case #	Features	Input values/Valid Inputs	Expected output	Results
4	Un Follow Doctor	NA	The link between doctor and patients will be removed	The link between doctor and patients is removed.

Table 17: Unfollow Doctor

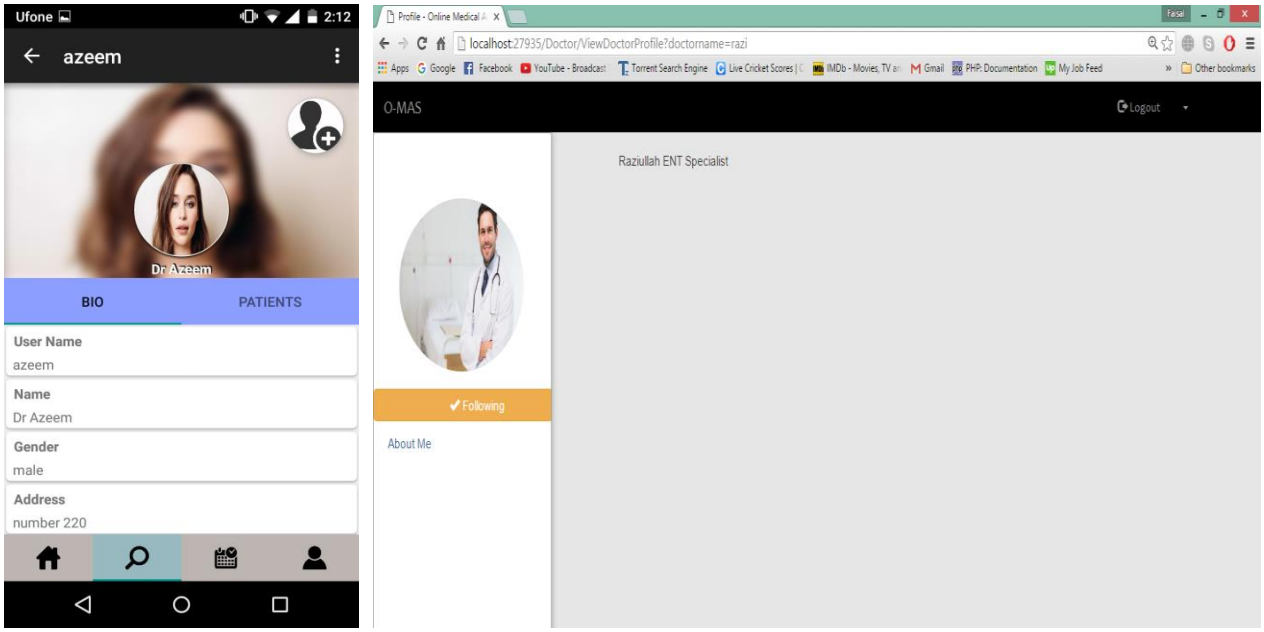
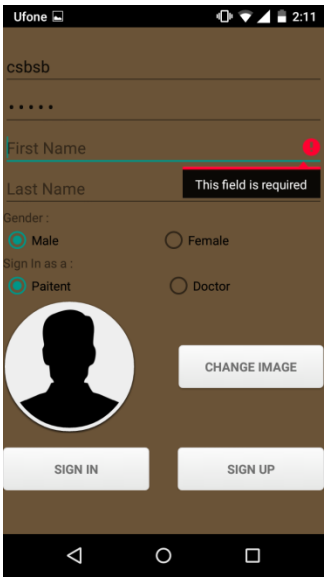


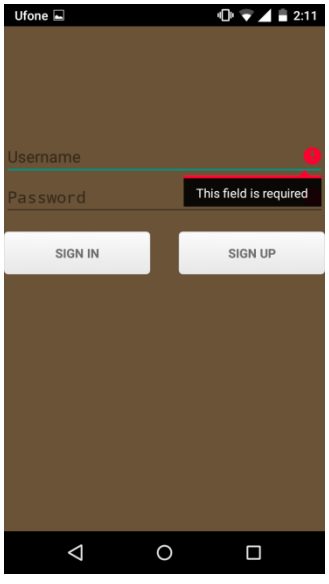
Figure 28: Unfollow

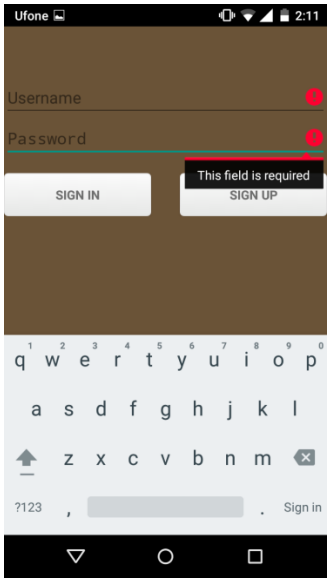
Test case #	Features	Description	Preconditions
5	Error Messages	Check all errors in input and displayed appropriate messages	Input or user navigation is required

Test case #	Features	Input values/Valid Inputs	Expected output	Results
5	Error Messages	Image file [png, jpeg]	The error messages will be prompt	The error messages are prompt

Table 18: Error Message



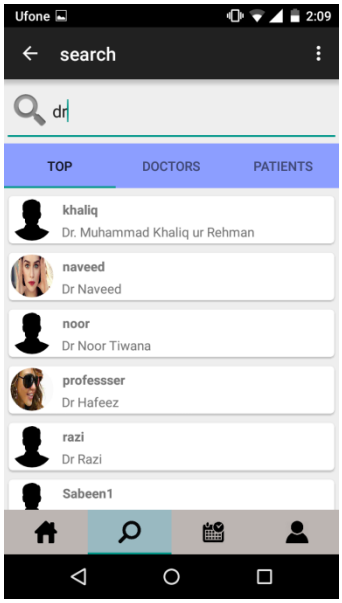




Test case #	Features	Description	Preconditions
6	Search profile	Check the searching functionality	The user must have logged in

Test case #	Features	Input values/Valid Inputs	Expected output	Results
6	Search profile	String	Result will be displayed according to input query	Result is displayed according to the query

Table 19: Search Profile



Test case #	Features	Description	Preconditions
7	Post Question	Check whether the question is posted to the server	The user must have logged in And must be a patient

Table 20: Post Question

Test case #	Features	Description	Preconditions
8	Answer Question	‘Check whether the answer are posted	The user must have logged in And must be a doctor

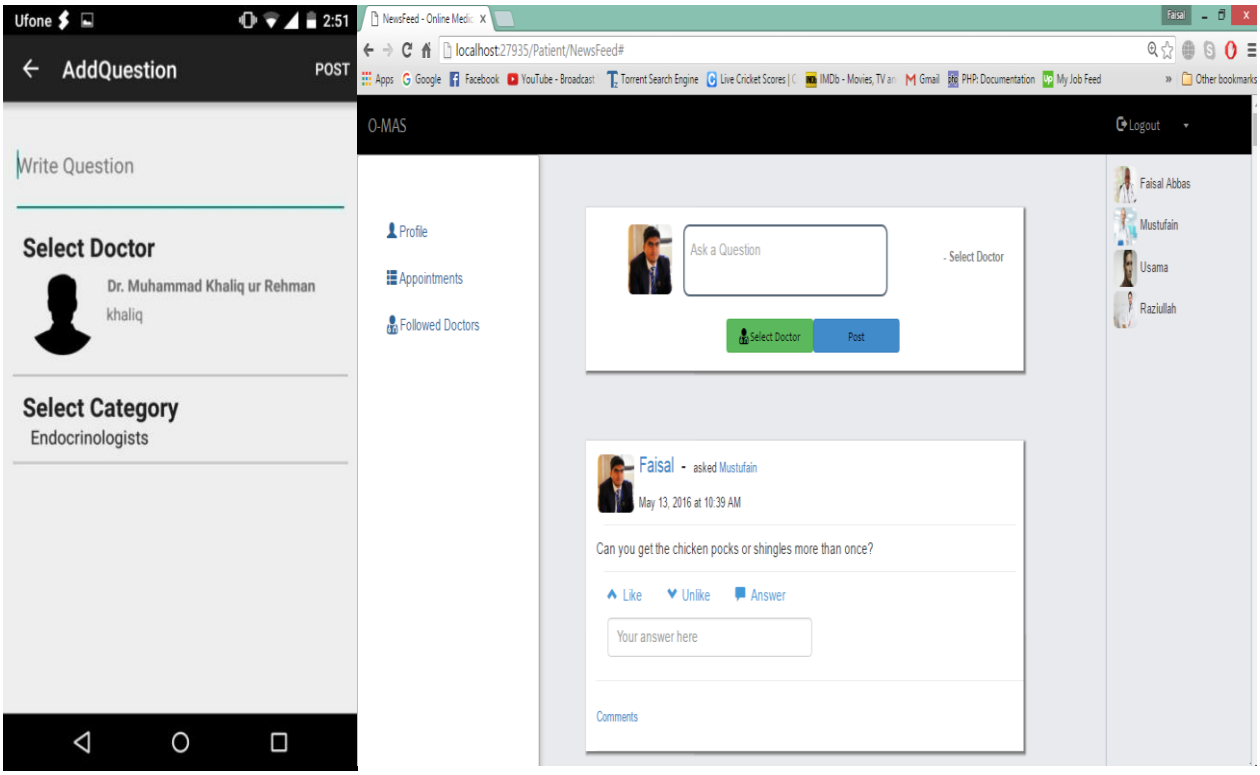


Figure 29: Ask Question

Test case #	Features	Input values/Valid Inputs	Expected output	Results
8	Answer Question	string	Answer will be posted	Answer are posted to the server

Table 22: Answer Question

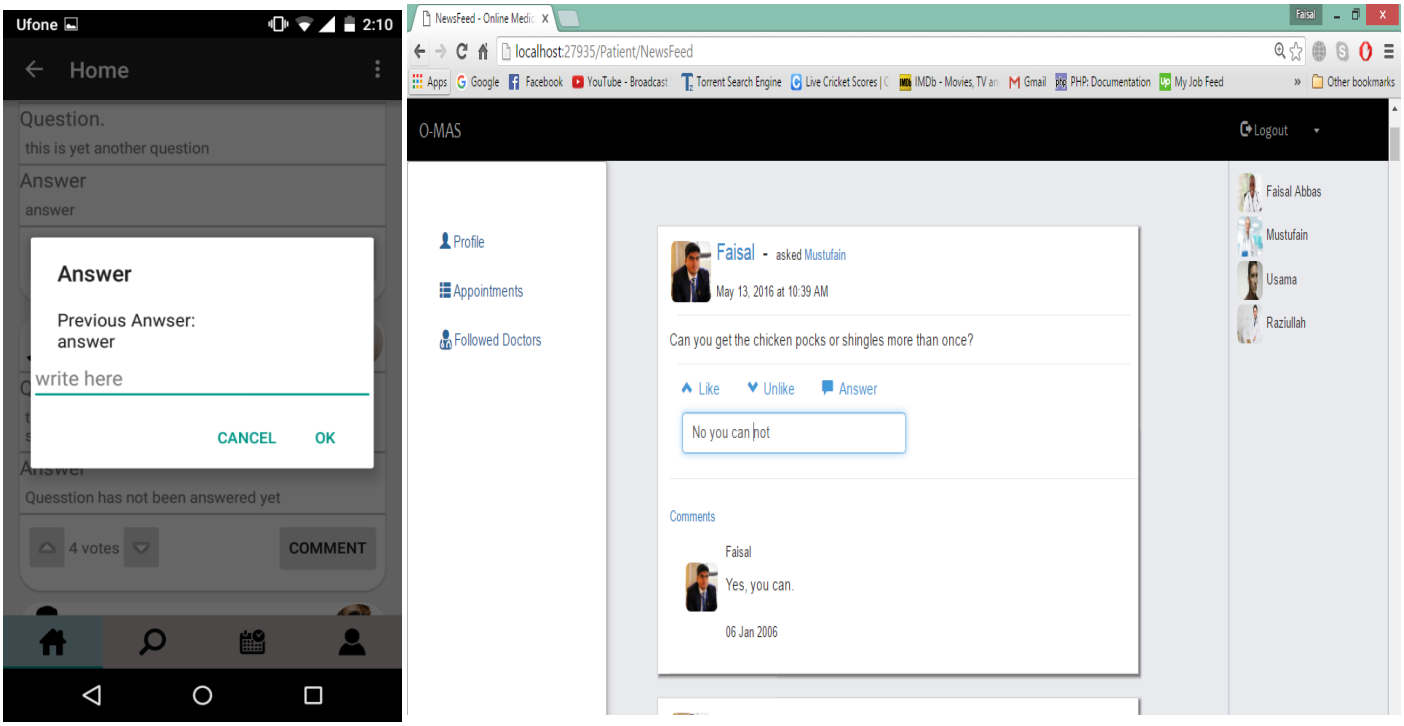
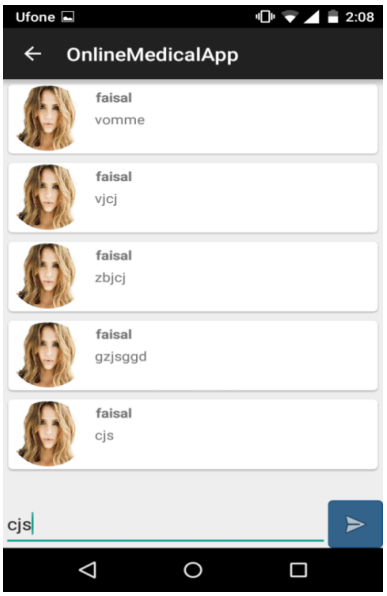


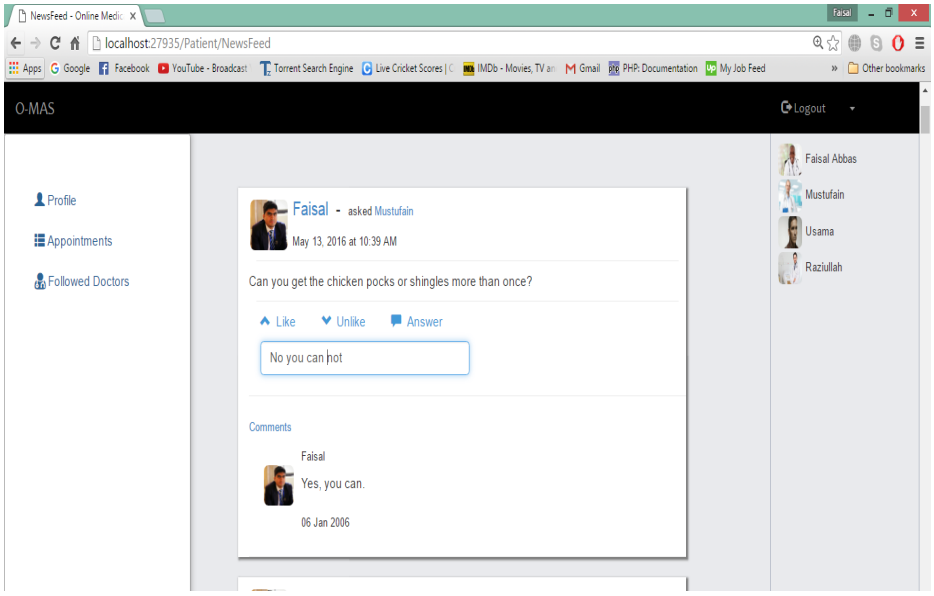
Figure 30: Answer Question

Test case #	Features	Description	Preconditions
9	Post Comments	Check whether the Comments are posted	The user must have logged in

Test case #	Features	Input values/Valid Inputs	Expected output	Results
9	Post Comment	string	Comments will be posted	Comments are posted to the server

Table 23: Post Comment

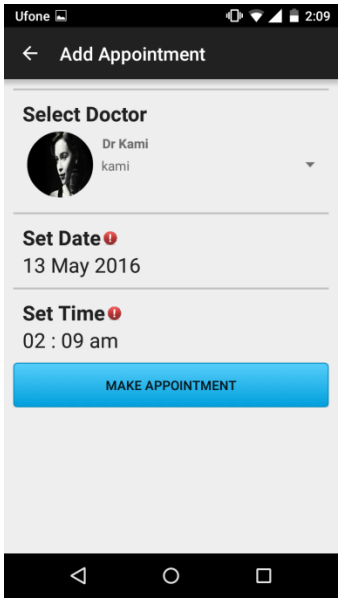


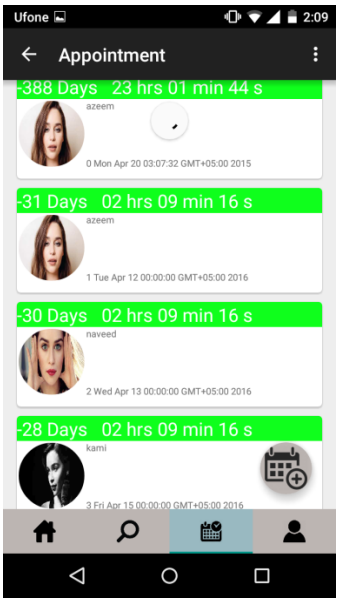


Test case #	Features	Description	Preconditions
10	Make Appointment	Check whether the Comments are posted	The user must have logged in And must be a Patient

Test case #	Features	Input values/Valid Inputs	Expected output	Results
10	Make Appointments	String, date, time	Appointments will be posted	Appointments are posted

Table 24: Make Appointments





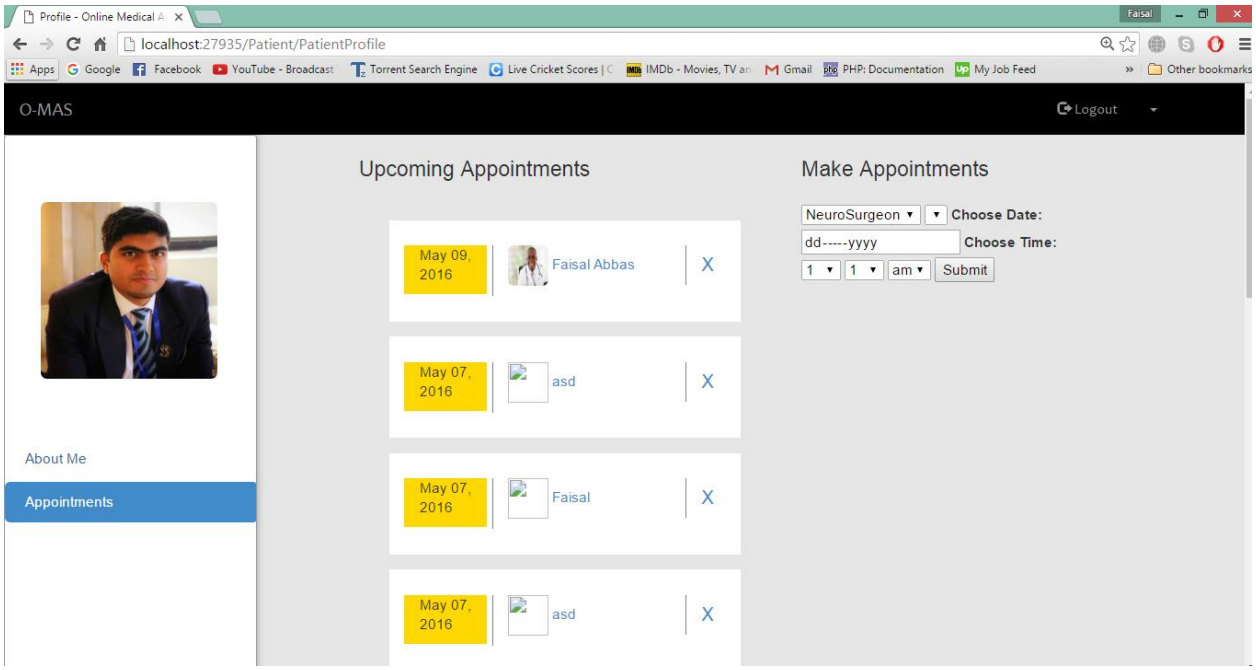
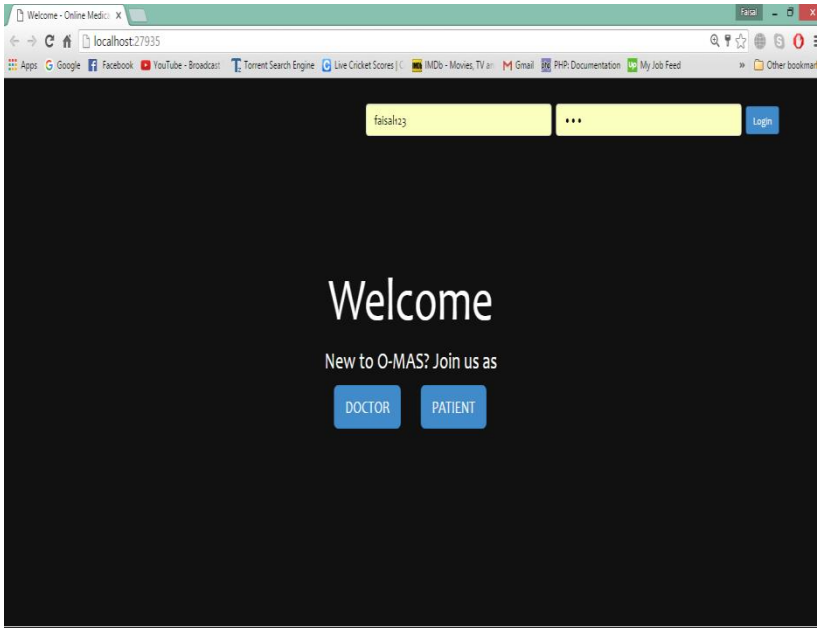


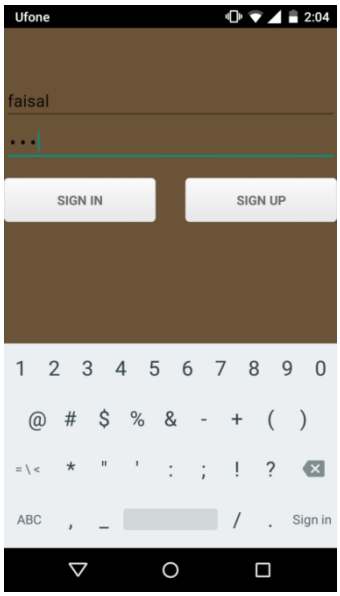
Figure 31: Appointments

Test case #	Features	Description	Preconditions
11	Sign in	Check]whether login is possible	The user must press the sign in button

Test case #	Features	Input values/Valid Inputs	Expected output	Results
11	Sign in	Strings	Sign in will be successful	Sign in is Successful

Table 25: Sign In





Test case #	Features	Description	Preconditions
12	Vote	Check weather the vote is posted	The user must press the up vote or down vote button

Test case #	Features	Input values/Valid Inputs	Expected output	Results
12	vote	N/A	Vote will be posted	Vote is posted

Table 26: Vote

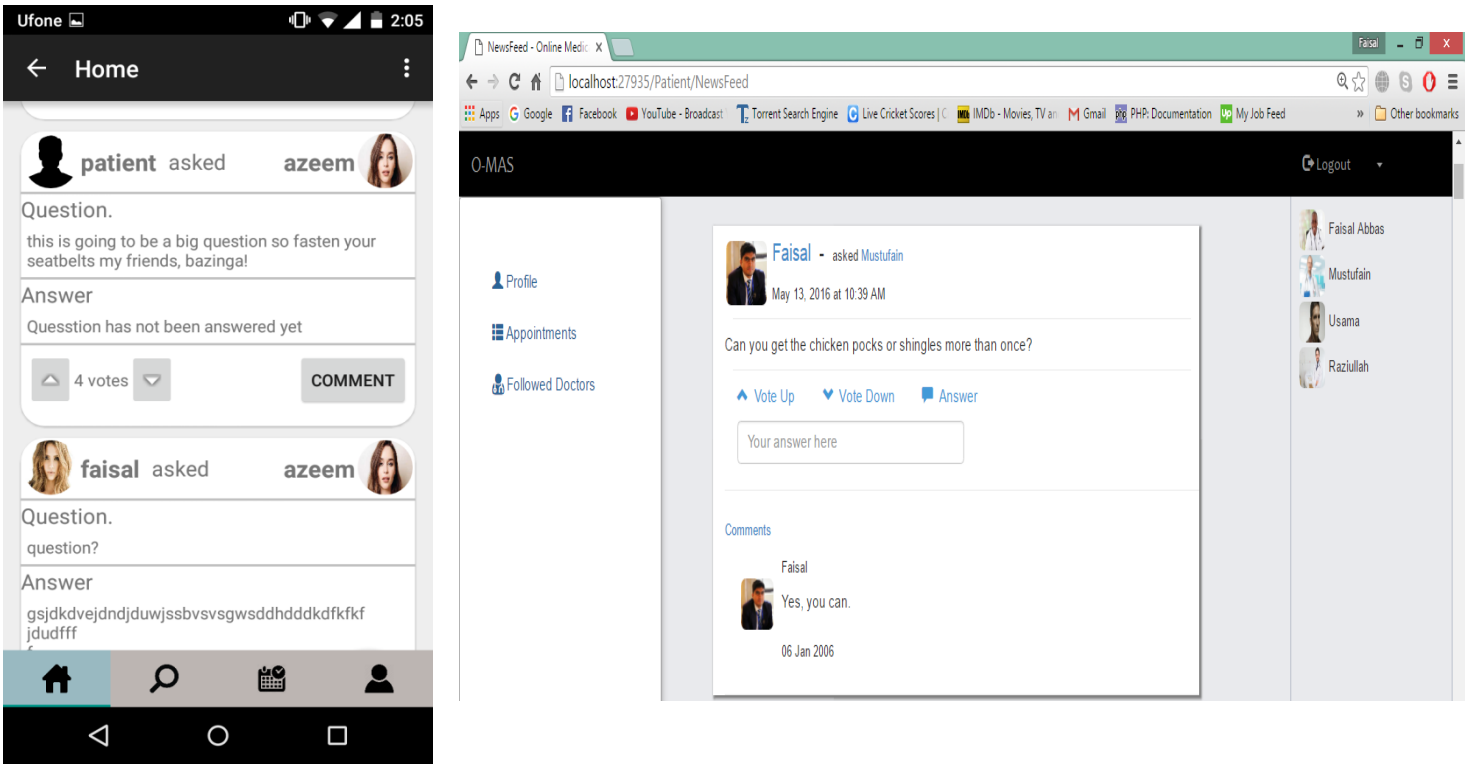


Figure 32: Vote