



EASTERN MEDITERRANEAN UNIVERSITY FAMAGUSTA NORTH -
CYPRUS

SOFTWARE ENGINEERING DEPARTMENT

CMSE473 - Software Process & Management

SYSTEM REQUIREMENTS SPECIFICATION DOCUMENT

Fall Semester 2016

Group 1

Contents

List of Figures	5
List of Tables.....	5
INTRODUCTION.....	6
1.Purpose.....	6
2.Scope	6
3.Glossary.....	6
4.Overview	7
General Description	7
1.Product Perspective	7
2.Product Functions.....	7
3.User Characteristics	8
4.General Constraints	8
5.Assumptions and Dependencies.....	8
6.Existing problems in this field	8
System specifications	9
1.Use case diagram.....	9
2.User requirements	10
1.Functional Requirements:	10
2.Non Functional Requirements.....	10
3.System Requirements	11
1.Functional requirements.....	11
2.Non- functional Requirements	13
4.Detailed Use Case Specification	15
5.Trace ability matrix	24
Design Documentation.....	25
1.Class Diagram	25
2.E R Diagram.....	26
3.Sequence Diagram	27
<u>i</u> .Sequence diagram on Login.....	27
ii.Sequence diagram on Registration.....	28

iii.Sequence diagram on search.....	29
iv.Sequence diagram on Book appointment.....	30
v.Sequence diagram on Cancel appointment	31
vi.Sequence diagram on Add available time.....	32
4.Data Flow Diagram	33
5.State Chart Diagrams	34
i.State Chart diagram for Registration	34
ii.State Chart Diagram for Login	34
iii.State Chart Diagram for Book appointment	35
iv.State Chart Diagram for Send enquiry.....	35
Implementation decisions.....	36
1.User module:	36
2.Admin module:.....	36
User module:	36
1.Login:	36
2.Registration:	36
3.Home page:	37
4.Doctor:.....	37
5.Department:.....	37
6.Appointment:.....	37
7.Contact:	37
8.Account Profile:	38
Admin module.....	38
1.Patient:.....	38
2.Doctor:.....	38
3.Department:	39
4.Hospital:	39
5.Appointment:.....	39
6.Enquiries:	40
7.Settings:.....	40
Acceptance Test Cases	41

1.Login Acceptance Test.....	41
2.Registration Acceptance Test.....	42
3.Home Page Acceptance Test.....	43
4.Book Appointment Acceptance Test	44
5.Contact form Acceptance Test.....	44
6.Account Profile Acceptance Test.....	45
7.Patient Acceptance Test.....	46
8.Doctor Acceptance Test	47
9.Department Acceptance Test	48
10.Hospital Acceptance Test.....	49
11.Appointment Acceptance Test	50
12.Make enquiries Acceptance Test	51
CONCLUSION	52
References	52

List of Figures

Figure 1. Use Case Diagram	9
Figure 2. Class Diagram.....	25
Figure 3. E R Diagram	26
Figure 4. Sequence diagram on Login	27
Figure 5. Sequence diagram on Registration	28
Figure 6. Sequence diagram on search.....	29
Figure 7. Sequence diagram on Book appointment	30
Figure 8. Sequence diagram on Cancel appointment.....	31
Figure 9. Sequence diagram on Add available time	32
Figure 10. Data Flow Diagram.....	33
Figure 11. State Chart diagram for Register	34
Figure 12. State Chart Diagram for Login	34
Figure 13. State Chart Diagram for Book appointment	35
Figure 14. State Chart Diagram for Send enquiry.....	35

List of Tables

Table 1. Glossary Table	6
Table 2. Trace ability matrix.....	24

INTRODUCTION

1. Purpose

The purpose of this document is to describe the requirements for the Online Doctor Appointment Reservation System (ODARS). The intended audience includes all stakeholders in the potential system. These include, but are not necessarily limited to, the following: Administrative Staff, doctors, patients and developers.

Developers should consult this document and its revisions as the only source of requirements for the project. They should not consider any requirements statements, written or verbal as valid until they appear in this document or its revision.

2. Scope

The proposed software product is the Online Doctor Appointment Reservation System (ODARS). In this project we are going to design and build a fully functional web based online doctor appointment system.

The aim of this project is to create an online doctor appointment system. Customers can be able to book doctor's appointment anywhere in the world, this can be done via web browser. Doctors can also register to this system, view appointments and set available time for appointment thereby making it more convenient for them. Administrator also have access to the website and able to change information of the website and have access to database. Payment is made after appointment.

3. Glossary

Table 1. Glossary Table

Term	Definition
Appointment	An arrangement to meet someone at a particular time and place.
Web-based application	An application that runs on the Internet.
Database	Collection of all the information monitored by this system.
Constraints	Limitations to the system.
Field	A cell within a form.
Business Domain	A set of classes that represents objects in the business model being implemented.
Web browser	A software application for retrieving and traversing information resources on world wide web.
Self-contained	Having all that is needed by itself.
Administrative staff	Person responsible for the management of operations.
Graphic user interface	A visual way of interacting with a computer using items

	such as windows, menus etc.
Software Requirements Specification	A document that completely describes all of the functions of a proposed system and the constraints under which it must operate. For example, this document.
Stakeholder	Any person with an interest in the project who is not a developer.
User	Patient or doctor.

4. Overview

This Software Requirements Specification (SRS) is the requirements work product that formally specifies Online Doctor Appointment Reservation System (ODARS). It includes the results of both business analysis and systems analysis efforts. Various techniques were used to elicit the requirements and we have identified your needs, analyzed and refined them. The objective of this document therefore is to formally describe the system's high level requirements including functional requirements, non-functional requirements and constraints. The detail structure of this document is organized as follows:

First we will provide an overview of the business domain that the proposed Online Doctor Appointment Reservation System (ODARS) will support. These include a general description of the product, user characteristics, general constraints, and any assumptions for this system. Second we will mention the User and System Requirements. After that, we will take a look at the design documentation and implementation decisions of the system and some test cases. At the end, my report will conclude the progress I made and discuss the problems faced during the work.

General Description

1. Product Perspective

This Doctor Appointment Reservation System is a self-contained system that allows patients to book appointment and doctors to manage appointments. Various stakeholders are involved in this system.

2. Product Functions

- a. Provide an application that enables patients to book an appointment with any available doctor.
- b. Doctors will be able to view their appointments and manage them properly.
- c. Coordinate various calendars and finding available time slots for appointment.
- d. Reserve equipment and rooms for appointment.
- e. Alert patients in case there is an earlier available time slot.

3. User Characteristics

The system will be used in the hospital. The doctors and patients will be the main users. Given the condition that not all the users are computer-literate. Some users may have to be trained on using the system. The system is also designed to be user-friendly. It uses a Graphical User Interface (GUI).

Patient: These are the people who want to make the appointment.

Doctor: These are the specialist whom appointments are being booked for.

Database Administrator: They are responsible for maintaining and overseeing the database of the system.

4. General Constraints

- The system must be delivered by the proposed deadline.
- The system must be user-friendly.

5. Assumptions and Dependencies

- It is assumed that IBM compatible computers will be available before the system is installed and tested.
- It is assumed that the Hospital will have enough trained staff to take care of the system.

6. Existing problems in this field

The major problem in this field is lengthy waiting time, patients would come to the hospital to book an appointment and would have to wait a really long time before they can see the doctor. But with this system each patient has an allocated time to see the doctor.

System specifications

1. Use case diagram

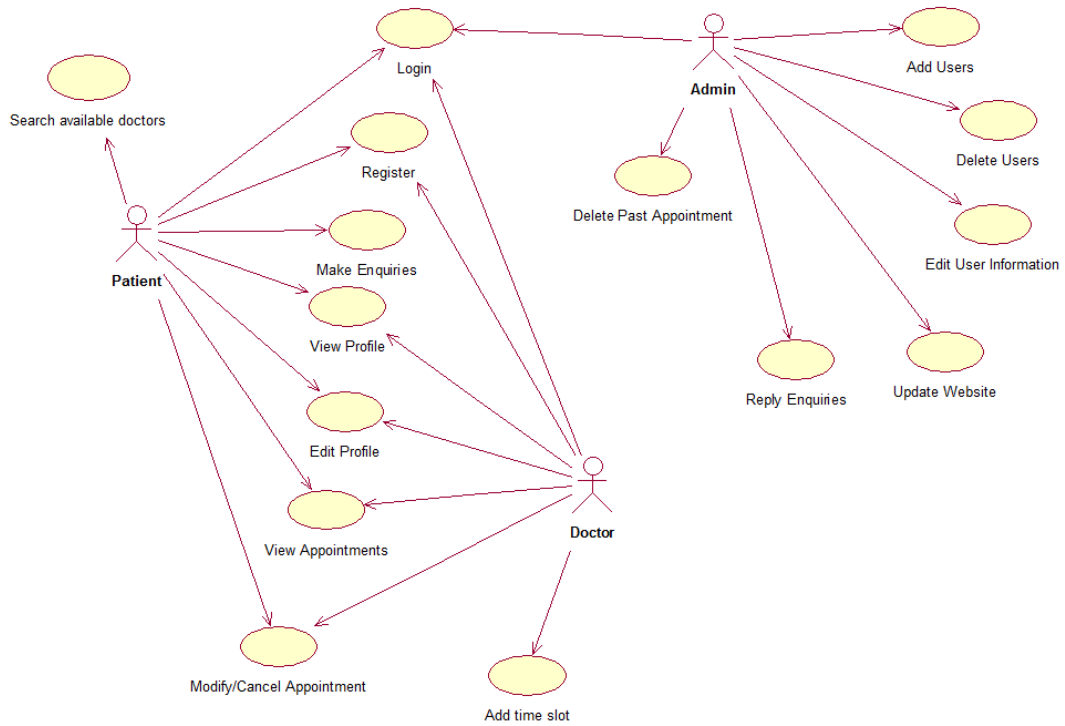


Figure 1. Use Case Diagram

2. User requirements

1. Functional Requirements:

- REQ1. The system should enable patients and doctors to log in.
- REQ2. The system should enable patients and doctors to register.
- REQ3. The system should enable patients and doctors to log out.
- REQ4. The system should allow Patients to book appointment.
- REQ5. The system should allow Patients to make enquiries.
- REQ6. The system should allow Patients to search for available doctors.
- REQ7. The system should allow Patients to modify or cancel their appointment.
- REQ8. The system should allow Patients and Doctors to view and modify their profile.
- REQ9. The system should allow Doctors to set their available time.
- REQ10. The system should enable administrator to log in.
- REQ11. The system should allow administrator to manage users
- REQ12. The system should enable administrator to reply enquiries.
- REQ13. The system should allow administrator to delete past appointments.
- REQ14. The system should allow administrator to manage & access data base.

2. Non Functional Requirements

i. Reliability

- The system should be available when requested for service by users.
- The system should have a very low failure rate.

ii. Performance

- The system must have a good response time.
- The system should be able to achieve a lot in a specified amount of time.
- The system must run error free while operating with a huge set of data.

iii. Security

- All external communications between the system's data server and clients must be encrypted.
- The access permissions for system data may only be changed by the system's data administrator.

- All system data must be backed up every 24 hours and the backup copies stored in a secure location which is not in the same building as the system.
- iv. Usability**
 - The system should include well-structured user manuals.
 - The system should have Informative error messages.
 - Efficient help facilities.
 - The system should have a well-formed graphical user interfaces.
 - The system should be user friendly.
- v. Safety**
 - The system should maintain a good back up.
- vi. Supportability**
 - The system should be able to be transferred from one environment to another.
 - The system should be easy to maintain
 - The system should be able to deal with additional international conventions such as languages, or number formats, styles.
 - The system should be able to be used on multiple platforms.

3. System Requirements

1. Functional requirements

- 1.1. The system should enable patients and doctors to log in.
 - They shall enter their username and password.
 - The information given shall be valid.
 - Access shall be granted/denied.
- 1.2. The system should enable patients and doctors to register.
 - In the case of patient collect user information (Names, Date of birth, address, telephone, email, password etc.).
 - In the case of doctor collect (Names, email, password, Date of birth, gender, department, address, telephone etc.).
 - Check if information is valid:
 - Password is not empty.
 - Password and confirm password is same.
 - Email hasn't been used before.
 - If information is valid save and add user to database.
- 1.3. The system should enable patients and doctors to log out.
 - Log user out when user clicks on log out button.
- 1.4. The system should allow Patients to book appointment.
 - The system shall check if the patient is logged in or not.

- The patient shall select the department and hospital of interest.
 - The system shall display the available time of the particular doctor to be booked.
 - The system shall generate a unique booking ref for each appointment.
 - The system shall send a confirmation email when appointment is made.
- 1.5. The system should allow patients to make enquiries.
- The system shall require the customer to give their email so a response can be sent.
 - The system shall require a comment to be entered, describing whatever issues the customer will like to know.
- 1.6. The system should allow patients to search for available doctors of a particular field.
- The Patient should be able to enter the first name or last name of doctor to be searched for.
 - The system displays the all doctors that fits patient's criteria.
 - The system shall display the doctor's available time.
- 1.7. The system should allow patients to modify or cancel their bookings.
- The system shall allow reservations to be modified without having to reenter all the patient's information.
 - The patient just has to provide their booking reference.
 - The system shall make the necessary updates after changes have been made.
- 1.8. The system should allow Patients and Doctors to view and modify their profile.
- They shall enter the new information.
 - This information then replaces the old information in database.
- 1.9. The system should allow Doctors to set their available time.
- The doctor will enter the time he'll be available.
 - This information is saved in the database.
- 1.10. The system should enable administrator to log in.
- The user shall enter their username and password.
 - The information given shall be valid.
 - Access shall be granted/denied.
- 1.11. The system should allow administrator to manage Users
- The system enable administrator to access database and add new customers.
 - The system enable administrator to delete any user due to some rules from database.
 - The system enable administrator to change patients or doctors information.
 - The system enable administrator to change patients or doctors information like last name, email, password, department etc.

- 1.12. The system should enable administrator to reply enquiries.
 - The administrator should be able to read the enquiries.
 - After the administrator writes the reply it should be sent to user.
- 1.13. The system should allow administrator to delete past appointments from system.
 - After the date of an appointment passes the administrator should delete the appointment from database.
- 1.14. The system should allow administrator to manage & access data base.
 - The system enable administrator to access database and manage database information.

2. Non- functional Requirements

i. Reliability

- The system should be available when requested for service by users: The system should work 24/7, it should always be up and running so that whenever the user wants to use it, it's available.
- The system should have a very low failure rate: The failure rate should be kept as minimal as possible, preferably less than 0.01.

ii. Performance

- The system must have a good response time.
- The load time for the user interface should take less than two seconds.
- The log in information should be verified within five seconds.
- Queries shall return results within five seconds.
- The system should be able to achieve a lot in a specified amount of time.
- The system should be able to withstand a heavy workload.
- It should be able to respond to multiple numbers of people at the same time.
- The system must run error free while operating with a huge set of data.
- The system should be precise and accurate when dealing with data.
- The system's error rate should be minimal.

iii. Security

- All external communications between the system's data server and clients must be encrypted:
- To ensure that the system is secure access to the various subsystems will be protected by a user log in screen and requires a user name and password.
- The access permissions for system data may only be changed by the system's data administrator: The system's administrator should be the only one with the authority to enable access to the system data.
- All system data must be backed up every 24 hours and the backup copies stored in a secure location which is not in the same building as the system: This is done to avoid loss of information in case of system crash. The system data should be stored in storage device e.g. hard drive, CD, Flash drive or it could be stored in files.

iv. Usability

- The system should include well-structured user manuals.
- The system should have a well- structured easy to understand manual to guide its users.
- The system should have Informative error messages.
- It should explain what the user did wrong.
- It should show where exactly the error can be found.
- It should explain how to recover from the error.
- The error message should be simple to understand.
- The system should have a well-formed graphical user interfaces.
- The system should be user friendly:
- The system must be easy to learn for both novices and users with experience from similar systems.
- The system must be efficient for the frequent user.
- The system must be easy to remember for the casual user.
- The user must understand what the system does.
- The user must feel satisfied with the system.

v. Safety

- The system should maintain a good backup: Maintaining backups ensures that the system's database is secured, which means that in case of an emergency or accident the system can be easily restored.

vi. Supportability

- The system should be able to be transferred from one environment to another.
- The system should still work perfectly when it is transferred from one operating environment to another.
- It should run on Microsoft windows, Linux, UNIX, and Mac OS.
- The system should be easy to maintain.
- In other for the system to be easy to maintain it should be done with an object oriented language which is easy to maintain.
- Maintenance of the system should be cost efficient.
- Maintenance of the system should be less frequent.
- The system should easily adapt to changes made.
- The system should be able to deal with additional international conventions such as languages, time zone, styles.
- The time zone should correspond to that of the user.
- The system should be able to be used on multiple platforms.
- The system should function properly on various platforms like hardware, browser, and virtual machine etc.

4. Detailed Use Case Specification

Use Case UC-1:	Login	
Related Requirements:	REQ1, REQ3, REQ4, REQ7, REQ8, REQ9, REQ10, REQ11, REQ12, REQ13, REQ14,	
Initiating Actor:	Patient, Doctor, Database administrator	
Actor's Goal:	To log into account	
Participating Actors:	—	
Preconditions:	The system displays the menu for doctor or patient to enter username and password.	
Post conditions:	The system should redirect user to homepage.	
Flow of Events for Main Success Scenario:		
→	1.	Doctor/patient opens website and clicks Log in.
←	2.	System shows user a menu to enter username and password.
→	3.	User enters the information and click on submit button.
←	4.	System displays “login successful”
Flow of Events for Extensions (Alternate Scenarios) :		
User enter invalid username or password		
←	1.	System shows a warning message that username or password entered was wrong and asked for entering information again.

Use Case UC-2:	Register	
Related Requirements:	REQ2, REQ4, REQ7, REQ8, REQ9	
Initiating Actor:	Patient, Doctor	
Actor's Goal:	To Create an account that can book appointment for patient and for doctor to able confirm and manage appointment.	
Participating Actors:	—	
Preconditions:	The system displays the menu for doctor or patient to enter personal information.	
Post conditions:	The system should redirect user to homepage.	
Flow of Events for Main Success Scenario:		
→	1.	Doctor/student open website and clicks register to create an account
←	2.	System shows user a page to get some basic information about user such as name, surname, user ID, date of birth, username, password and etc.
→	3.	User enters the information and click OK to account to be made
←	4.	System displays “registration successful”
Flow of Events for Extensions (Alternate Scenarios) :		
User enter invalid username or password		
←	1.	System shows a warning message that username or password entered was wrong and asked for entering information again.

Use Case UC-3	Appointment Reservation	
Related Requirements:	REQ1, REQ2, REQ4, REQ7, REQ13	
Initiating Actor:	Patient	
Actor's Goal:	To make online doctor appointment	
Participating Actors:	—	
Preconditions:	User must create an account and logged in	
Post conditions:	The system should display doctor's schedule	
Flow of Events for Main Success Scenario:		
→	1.	Patient requests a reservation by clicking on book appointment button
←	2.	System shows patient a menu to choose department and location.
→	3.	Patient enters information.
←	4.	System shows a list of doctors that suit the criteria.
→	5	Patient selects doctor.
←	6	System display's doctor's schedule.
→	7	Patient selects preferred time and day.
←	8	System displays "appointment booked".
Flow of Events for Extensions (Alternate Scenarios) :		
User wants to Cancel the reservation		
→	1.	User clicks on "Cancel Reservation" button
←	2.	System shows a message that cancelation is done

Use Case UC-4:	View and modify profile.	
Related Requirements:	REQ1, REQ2, REQ8	
Initiating Actor:	Patient, Doctor	
Actor's Goal:	To change personal information.	
Participating Actors:	—	
Preconditions:	The user must log in to the system	
Post conditions:	The system should display a verification message to ensure information updated	
Flow of Events for Main Success Scenario:		
→	1.	User clicks on account page.
←	2.	System displays user's personal account page.
→	3.	User selects option edit info
←	4	The system displays form to user.
→	4.	User enters new information saves the made changes by clicking on the Save button
←	5.	System displays "account updated".

Use Case UC-5	Set available time	
Related Requirements:	REQ1, REQ2, REQ9	
Initiating Actor:	Doctor	
Actor's Goal:	To set available time slot in schedule	
Participating Actors:	—	
Preconditions:	User must create an account and logged in	
Post conditions:	The system should display doctor's new schedule	
Flow of Events for Main Success Scenario:		
→	1.	Doctor clicks on account button.
←	2.	System shows account page.
→	3.	Doctor selects a free time slot on displayed schedule.
←	4.	Updates schedule displays new available time slot in schedule.

Use Case UC-6	Cancel appointments	
Related Requirements:	REQ1, REQ2, REQ4, REQ7	
Initiating Actor:	Patient, Doctor	
Actor's Goal:	To cancel previously booked appointment	
Participating Actors:	—	
Preconditions:	The user must log in to the system	
Post conditions:	The system must display a message “appointment canceled”	
Flow of Events for Main Success Scenario:		
→	1.	User clicks on appointment slot on calendar.
←	2.	System displays confirmation box “are you sure you want to cancel appointment”.
→	3.	User selects YES.
←	4.	System deletes appointment display a message “appointment cancelled”.
Flow of Events for Extensions (Alternate Scenarios) :		
User doesn't want to cancel anymore		
→	1.	User selects NO on confirmation box.
←	2.	System displays initial appointment schedule.

Use Case UC-7	Search doctor	
Related Requirements:	REQ6	
Initiating Actor:	Patient/Doctor	
Actor's Goal:	To search for a particular doctor	
Participating Actors:	—	
Preconditions:	User opens website	
Post conditions:	The system should display doctor's information	
Flow of Events for Main Success Scenario:		
→	1.	User clicks on search button.
←	2.	System displays search form.
→	3.	User enters doctors name and clicks search.
←	4.	System displays doctor's information.
Flow of Events for Extensions (Alternate Scenarios) :		
Doctor not in system.		
←	1.	System displays "doctor doesn't exist"

Use Case UC-8:	Manage users	
Related Requirements:	REQ1, REQ10, REQ11	
Initiating Actor:	Database administrator	
Actor's Goal:	To add user to system	
Participating Actors:	—	
Preconditions:	The user must log in to the system	
Post conditions:	The system must show a message to verify that update has been completed	
Flow of Events for Main Success Scenario:		
→	1.	User clicks on doctor/patient page.
←	2.	System displays list of doctor/patient.
→	3.	User selects option add new doctor/patient
←	4.	System displays form.
→	4.	User enters information and clicks on add button.
←	5.	System displays a message to show that user has been added.

Use Case UC-9:	Manage & Access Data Base	
Related Requirements:	REQ10, REQ14	
Initiating Actor:	Database Administrator	
Actor's Goal:	To managing data, add, edit, update or delete data	
Participating Actors:	—	
Preconditions:	The user must log in to the system	
Post conditions:	The system should show a message that modified data is stored into the data base	
Flow of Events for Main Success Scenario:		
→	1.	User clicks on database button
←	2.	System displays all the data in database
→	3.	User can add, remove or update specific part of database
	4.	User saves the changes by clicking on save button
←	5.	System displays a message to inform user that changes has been made successfully

5. Trace ability matrix

Table 2. Trace ability matrix

REQ'T	PW	UC1	UC2	UC3	UC4	UC5	UC6	UC7	UC8	UC9
REQ1	5	X		X	X	X	X		X	
REQ2	2		X	X	X	X	X			
REQ3	4	X								
REQ4	2	X	X	X			X			
REQ5	2									
REQ6	4							X		
REQ7	4	X	X	X			X			
REQ8	3	X	X		X					
REQ9	3	X	X			X				
REQ10	3	X							X	X
REQ11	4	X							X	
REQ12	5	X								
REQ13	3	X		X						
REQ14	2	X								X
MAX PW		5	4	5	5	5	5	4	5	3
TOTAL PW		38	14	16	10	10	13	4	12	5

Design Documentation

1. Class Diagram

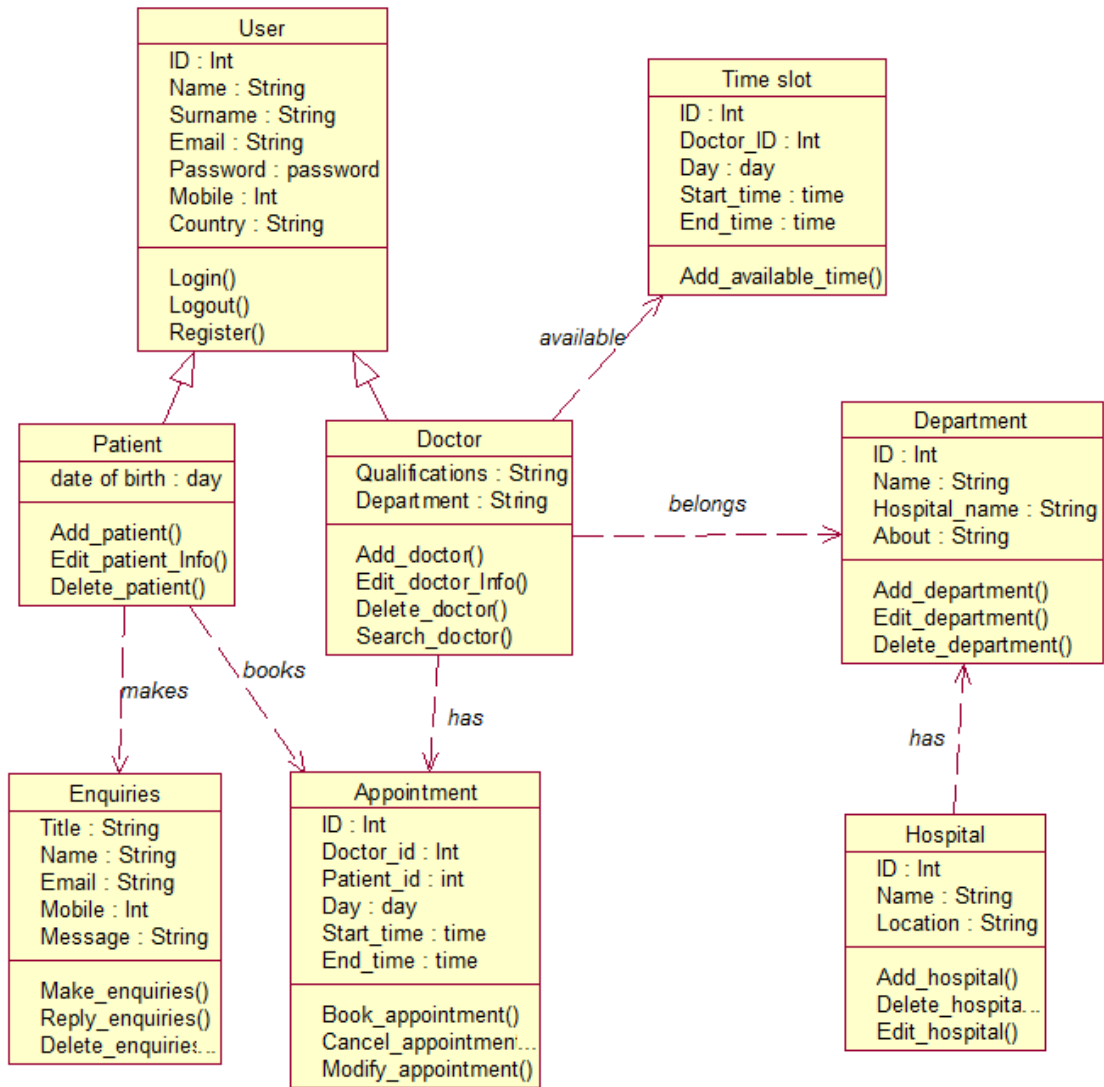


Figure 2. Class Diagram

2. E R Diagram

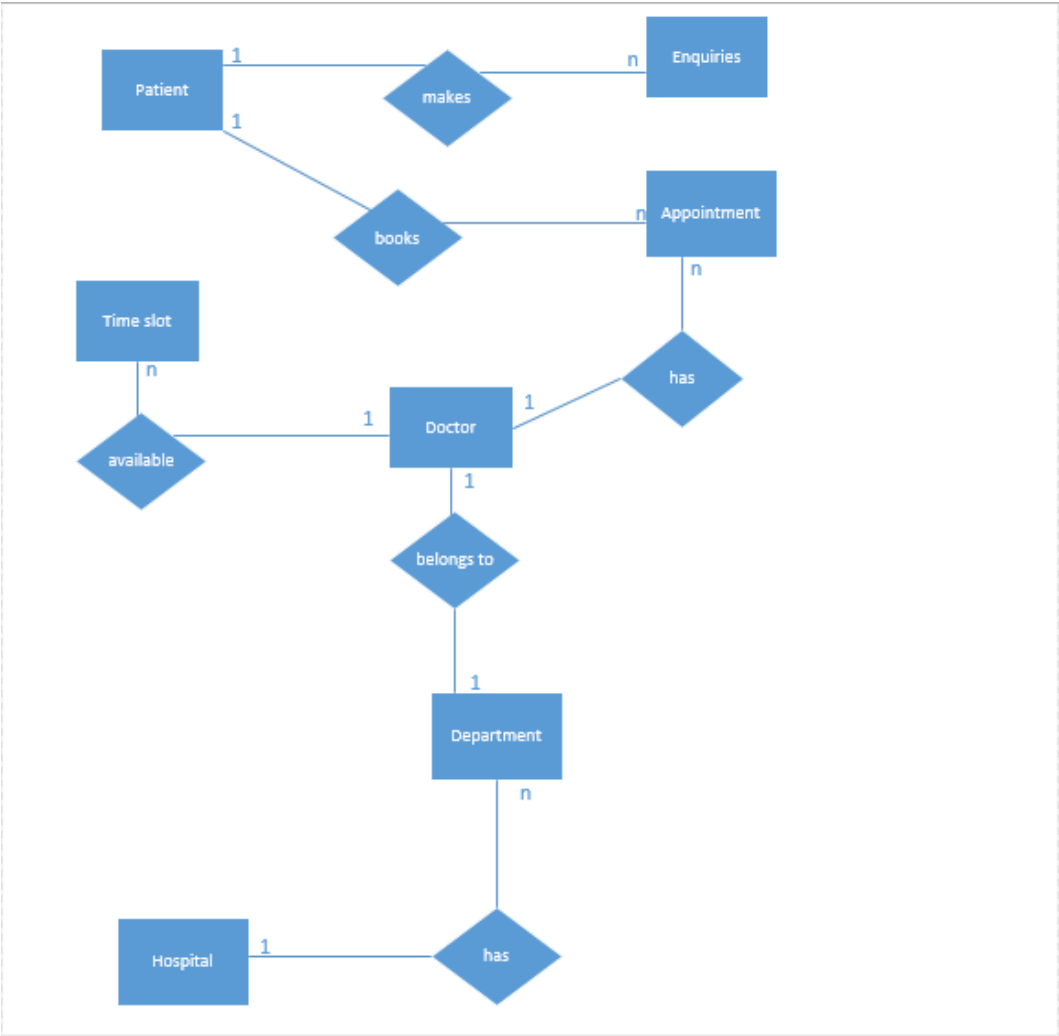


Figure 3. E R Diagram

3. Sequence Diagram

i. Sequence diagram on Login

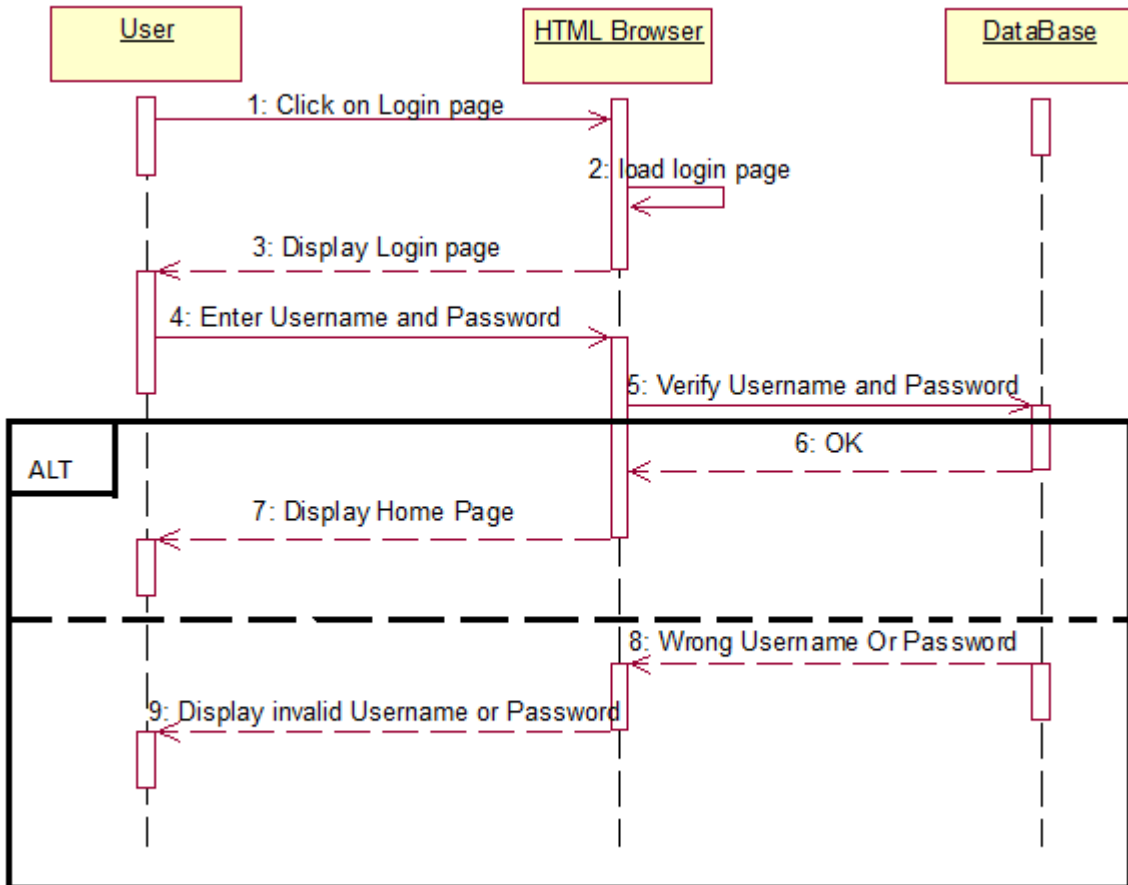


Figure 4. Sequence diagram on Login

ii. Sequence diagram on Registration

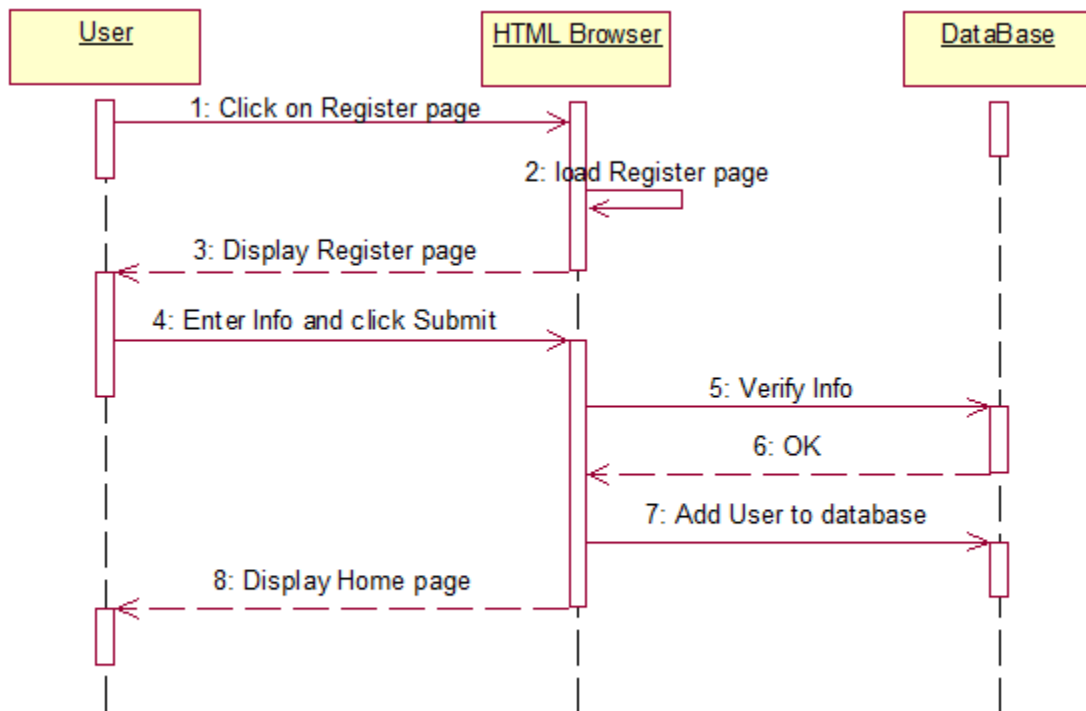


Figure 5. Sequence diagram on Registration

iii. Sequence diagram on search

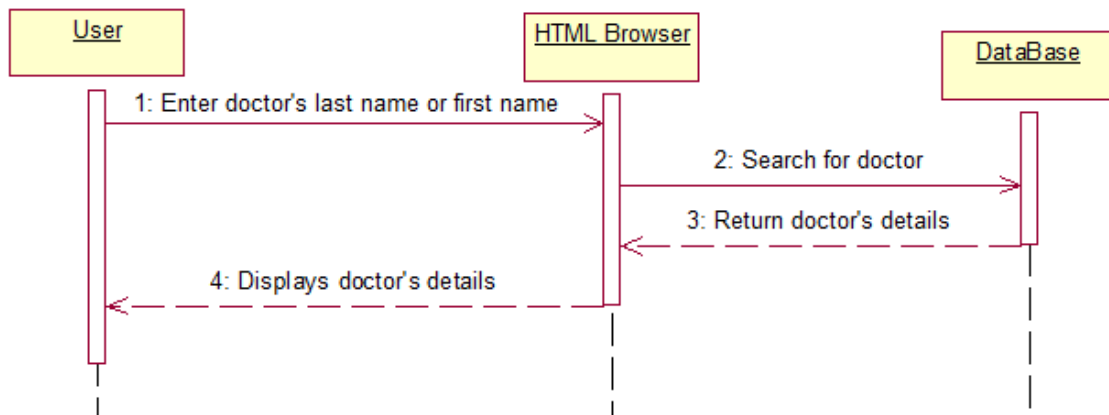


Figure 6. Sequence diagram on search

iv. Sequence diagram on Book appointment

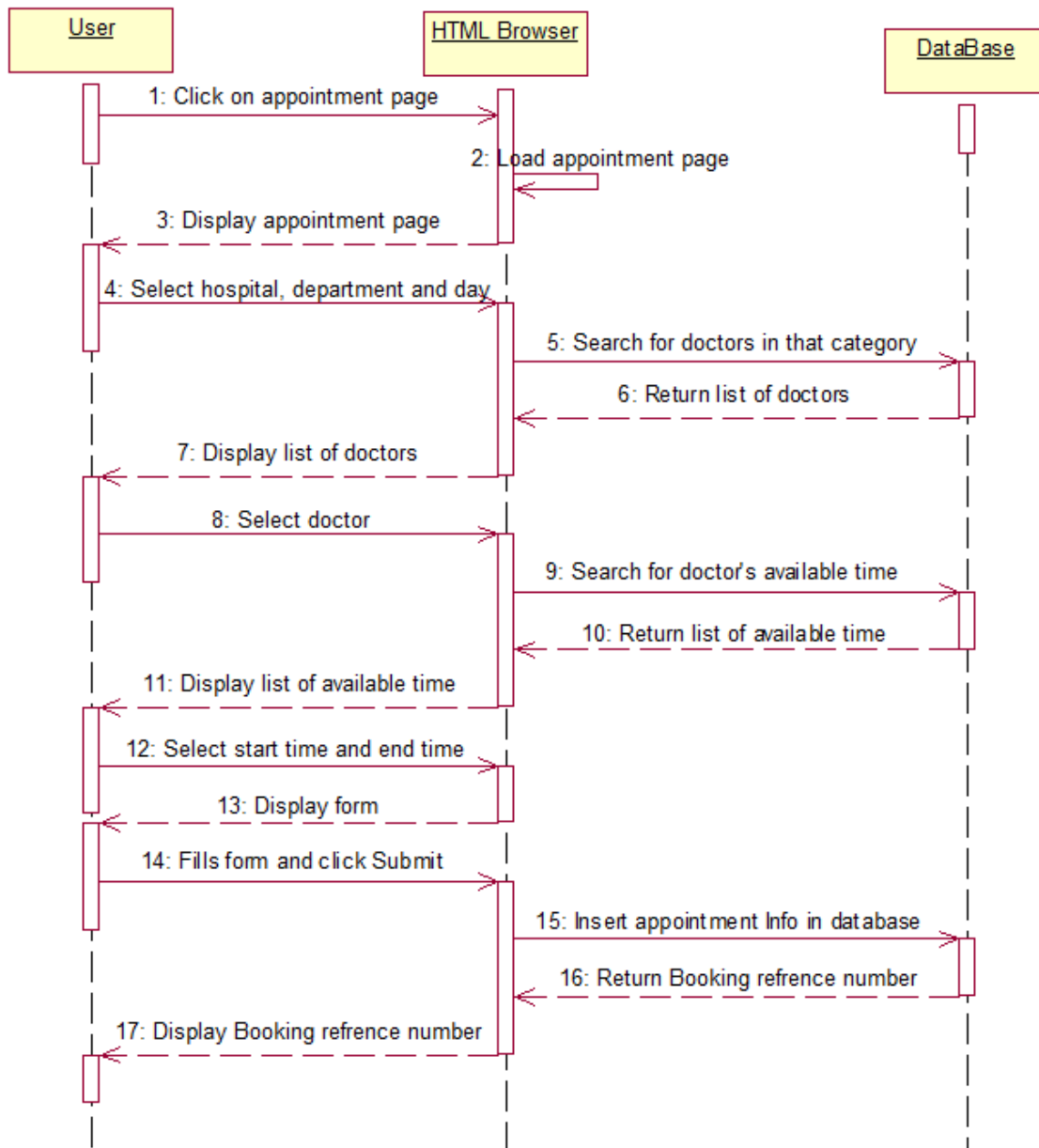


Figure 7. Sequence diagram on Book appointment

v. Sequence diagram on Cancel appointment

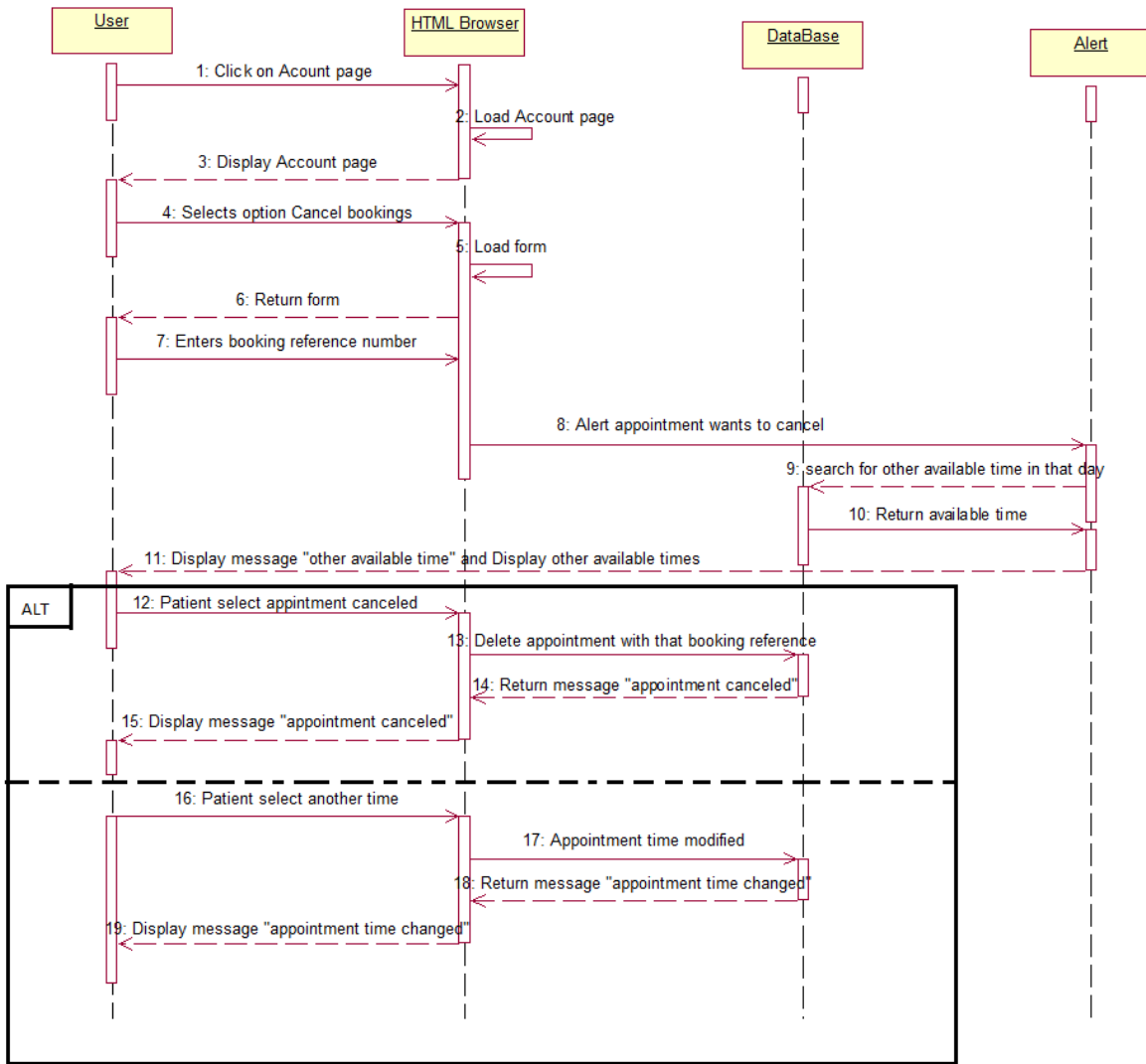


Figure 8. Sequence diagram on Cancel appointment

vi. Sequence diagram on Add available time

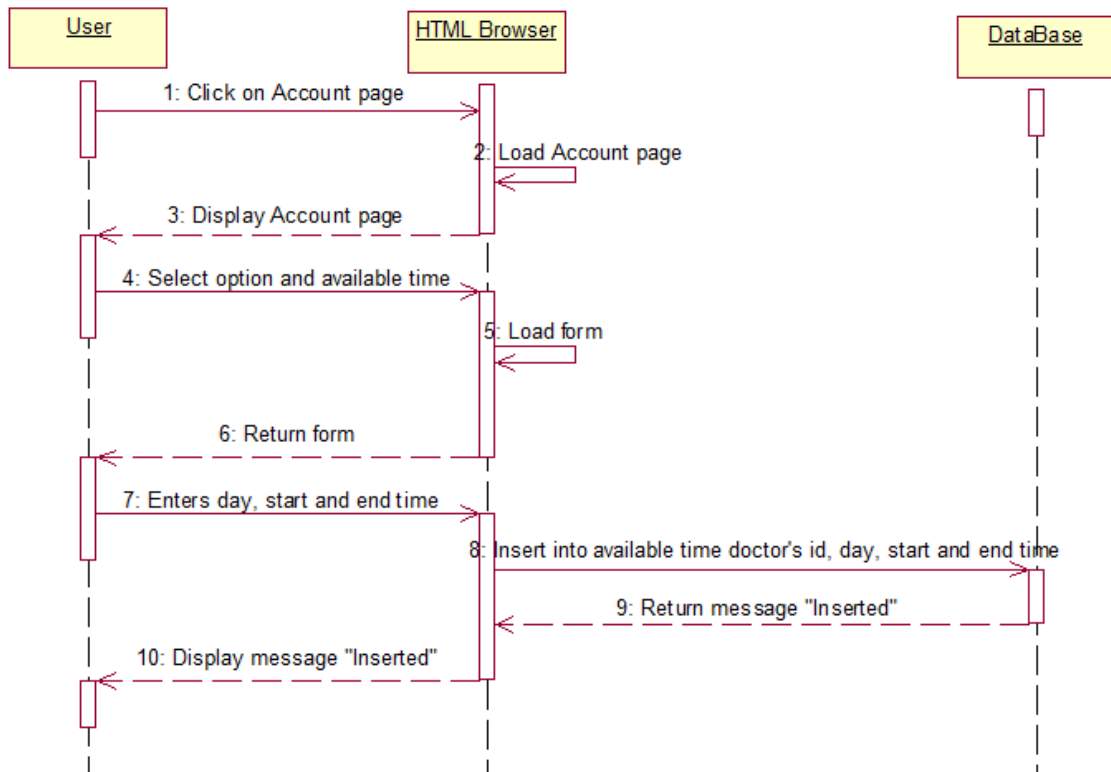


Figure 9. Sequence diagram on Add available time

4. Data Flow Diagram

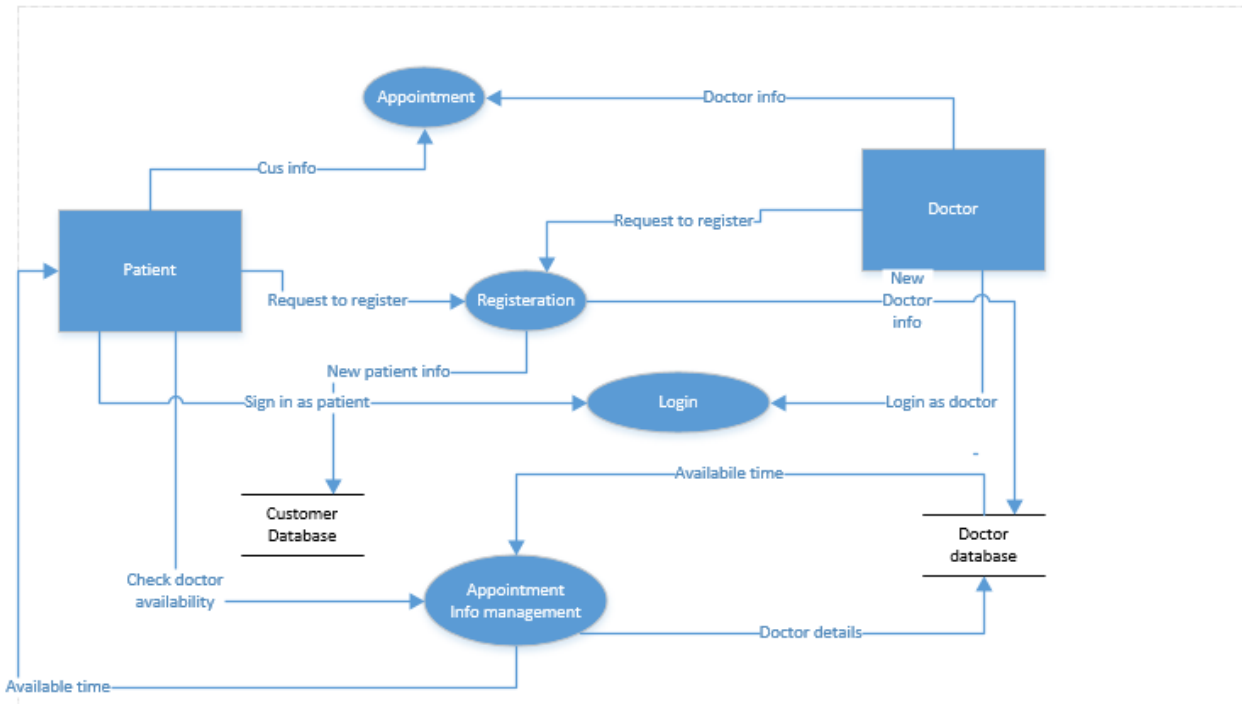


Figure 10. Data Flow Diagram

5. State Chart Diagrams

i. State Chart diagram for Registration

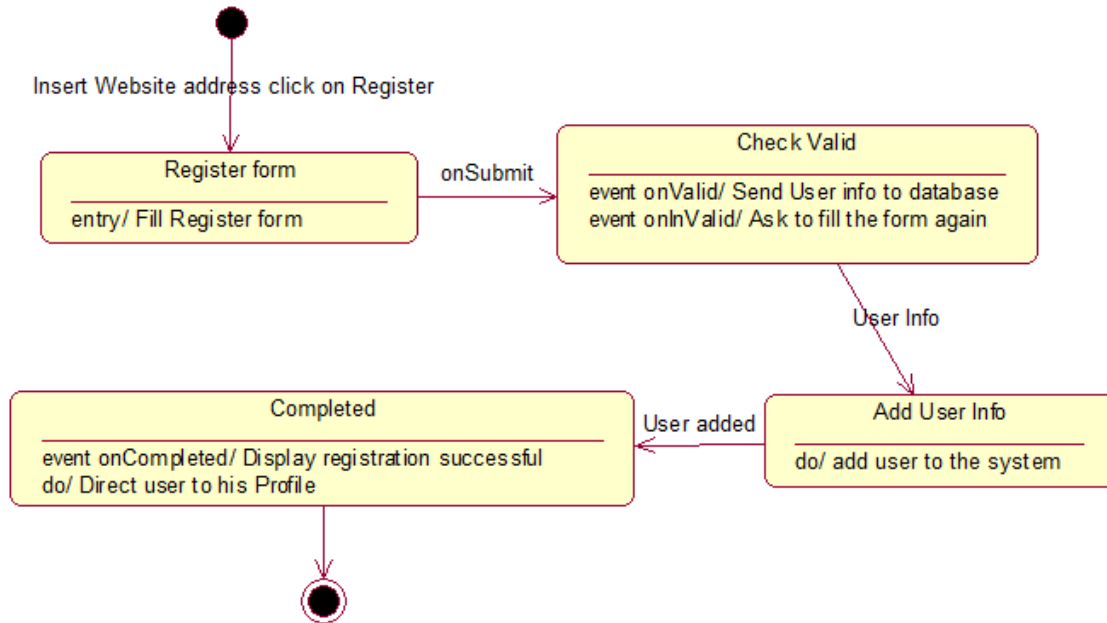


Figure 11. State Chart diagram for Register

ii. State Chart Diagram for Login

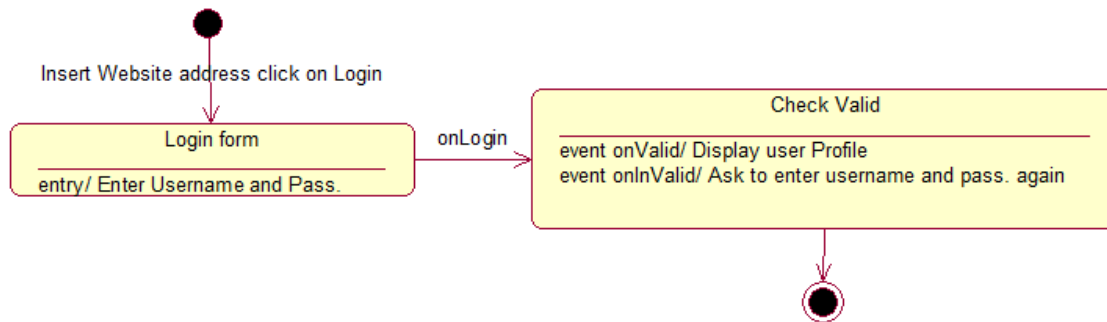


Figure 12. State Chart Diagram for Login

iii. State Chart Diagram for Book appointment

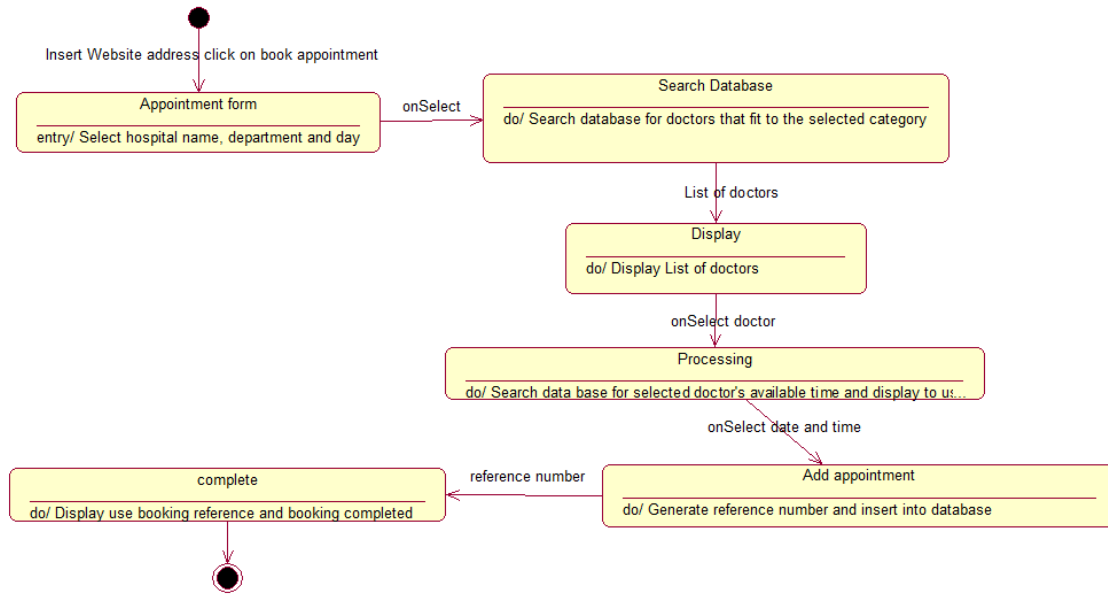


Figure 13. State Chart Diagram for Book appointment

iv. State Chart Diagram for Send enquiry

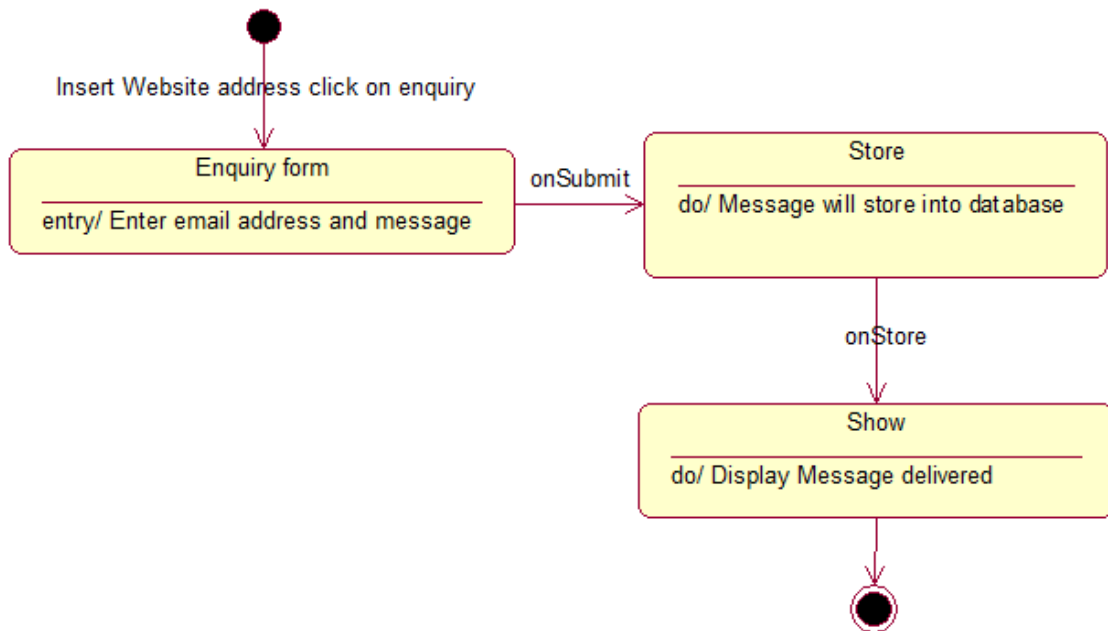


Figure 14. State Chart Diagram for Send enquiry

Implementation decisions

The system was implemented based on its functional requirements. Based on the functional requirements the system should have the following modules:

- 1. User module:** this is the module in which Users activities would occur it entails the following interfaces:
 1. Login
 2. Registration
 3. Home page
 4. Doctor
 5. Department
 6. Appointment
 7. Contact
 8. Account Profile
- 2. Admin module:** this is the controller module in which administrator would use to monitor and control users and database information, it entails the following interfaces:
 1. Patient
 2. Doctor
 3. Department
 4. Hospital
 5. Appointment
 6. Enquiries
 7. Settings

User module:

1. Login:

In order for the user (patient & doctor) to login, a username and password needs to be entered and this was implemented in a form that contained fields for name and password, when the form is submitted, the server checks the database if the email is available and the validity of the password, if the password and username is valid the user is redirected to the homepage else an error message is shown informing the user about the error. The login page also had a link that directs users that are not already registered to the registration page.

2. Registration:

In the case of registration the user has to provide personal information like name, email, password, verify password, date of birth, address, gender, telephone all these information are to be inputted into their respective fields in a form, when the form is submitted the server first checks, if the password field is empty then if the password and verify password are the same,

the system checks if the email is already registered if all these are correct, then a new user is created by inserting all these information in the database else an error message is displayed.

3. Home page:

This is the default page of the website i.e. the first page you see when you enter the website or after you login or register, this page is the most important it implements some functionalities specified in the requirements like:

- **Search:** A search box is provided at the top of the page, with this you can search for doctors by first name or last name, when the search is submitted the user is redirected to another page and a list of cars fitting that description is displayed.

The homepage contains a slider displaying some of the services offered by the business, it also has a section briefly discussing information about the business, departments in the business and founders of the business.

4. Doctor:

The doctor page displays the all the doctors available in the database, the doctor's information are displayed, also there is a book appointment button on each doctor row, when the button is clicked on the user is directed to another page where the patient can book that particular doctor appointments can only be booked if the user is logged in else the user is directed to the log in page after which the user is redirected back to the destined/original page.

5. Department:

The department page displays all the necessary information about a particular department available in the database, there is also a list of hospitals that have that particular department.

6. Appointment:

The reservation page displays a form with the fields Hospital, Department, date when this form is filled and submitted by the users, the system uses this information to search through the database and displays doctors fitting the users specification, when the user selects a doctor, he is directed to a page to book the doctor, appointments can only be booked if the user is logged in else the user is directed to the log in page after which the user is redirected back to the previous action page. After the user fills all necessary information and books the appointment a confirmation email is sent.

7. Contact:

The contact page consists of a map displaying the company's location, also the address and contact details of the company and lastly a contact form in which users can fill and send in the case of making enquiries. When this form is submitted it is stored in the database awaiting reply.

8. Account Profile:

The account profile interface implements the following functionalities:

- **Edit profile:** the user can view and edit their profile after which the new information is saved and updated in the database.
- **View appointments:** all the bookings made by the user is displayed in a tabular form, appointment information is displayed.
- **Edit appointments:** The user can change the dates of appointments; this new information is updated and stored in the database.
- **Cancel appointment:** a field for booking reference is provided, with this booking reference the user's booking is canceled thereby deleting it from the database.
- **Logout:** when this option is clicked the user is logged out and redirected to a more general homepage.

Admin module

1. Patient:

In this interface the following functionalities are implemented for the administrator:

- **View patient:** A list of all the patients in the database is displayed and when a patient item on the list is clicked, the patient details is displayed in a form
- **Add patient:** the administrator can add patients to the database, a form is provided and the patient's details are entered, when the form is submitted the new patient is created and inserted into the database.
- **Edit patient information:** The administrator can also edit patient information, whatever information entered is saved and updated in the database.
- **Delete patient:** A delete option is placed beside each patients in the list of patients and when it is clicked that particular patient is deleted.

2. Doctor:

In this interface the following functionalities are implemented for the administrator:

- **View doctor:** A list of all the doctors in the database is displayed and when a doctor item on the list is clicked, the doctor details is displayed in a form
- **Add doctor:** the administrator can add doctors to the database, a form is provided and the doctor's details are entered, when the form is submitted the new doctor is created and inserted into the database.
- **Edit doctor information:** The administrator can also edit doctor information, whatever information entered is saved and updated in the database.

- **Delete doctor:** A delete option is placed beside each doctor in the list of doctors and when it is clicked that particular doctor is deleted.

3. Department:

In this interface the following functionalities are implemented for the administrator:

- **View department:** A list of all the departments in the database is displayed and when a department item on the list is clicked, the department details is displayed in a form
- **Add department:** the administrator can add departments to the database, a form is provided and the department's details are entered, when the form is submitted the new department is created and inserted into the database.
- **Edit department information:** The administrator can also edit department information, whatever information entered is saved and updated in the database.
- **Delete departments:** A delete option is placed beside each departments in the list of departments and when it is clicked that particular department is deleted.

4. Hospital:

In this interface the following functionalities are implemented for the administrator:

- **View hospital:** A list of all the hospitals in the database is displayed and when a hospital item on the list is clicked, the hospital details is displayed in a form
- **Add hospital:** the administrator can add hospitals to the database, a form is provided and the hospital's details are entered, when the form is submitted the new hospital is created and inserted into the database.
- **Edit hospital information:** The administrator can also edit hospital information, whatever information entered is saved and updated in the database.
- **Delete hospital:** A delete option is placed beside each hospitals in the list of hospitals and when it is clicked that particular hospital is deleted.

5. Appointment:

In this interface the following functionalities are implemented for the administrator:

- **View appointment:** A list of all the appointments in the database is displayed in the order of dates and when an appointment item on the list is clicked, the appointment details is displayed in a form
- **Add appointment:** the administrator can add appointments to the database, a form is provided and the appointment's details are entered, when the form is submitted the new appointment is created and inserted into the database.
- **Edit appointment information:** The administrator can also edit appointment information, whatever information entered is saved and updated in the database.

- **Delete appointment:** A delete option is placed beside each appointment in the list of hospitals and when it is clicked that particular hospital is deleted.

6. Enquiries:

This interface enables the administrator to view and reply enquiries submitted by the patients. All the enquiries submitted by patients are displayed in a list with a reply option provided for each enquiry, when it is clicked a form is displayed where the administrator can write a reply to patient, when this form is submitted the content is sent as an email to the patient.

7. Settings:

This interface enables the administrator to change the general settings of the website e.g. Site title.

Acceptance Test Cases

1. Login Acceptance Test

Test-Case ID:	TC-01
Test-Case Name:	Login (Doctor & Patient)
Pass/Fail Criteria :	The test passes if the user enters a correct username or password that is contained in the database.
Input Data:	Numeric and alphabet key code
Test Procedure:	Expected Outcomes :
<u>Step1</u> Type an incorrect username or password	System beeps to indicate failure; records unsuccessful attempt in the database; an error message is shown informing the user about the error.
<u>Step 2</u> Type a correct username and password	System indicate success; records successful access in the database; user is redirected to the homepage
Comments:	The login page have a link that directs users that are not already registered to the registration page.

2. Registration Acceptance Test

Test-Case ID:	TC-02
Test-Case Name:	Registration
Pass/Fail Criteria :	The test passes if the user provide personal information like name, email, password , verify password ,date of birth, address, gender, telephone all these information are to be inputted into their respective fields in a form
Input Data:	Numeric and alphabet key code
Test Procedure:	Expected Outcomes :
<u>Step 1</u> Empty fields in the form	System beeps to indicate failure; records unsuccessful attempt in the database; an error message is displayed.
<u>Step 2</u> Fill in all the fields	System indicate success; records successful access in the database If the password and verify password are the same, the system checks the email is already registered correct, then a new user is created .
<u>Step 3</u> Type different passwords in the fields of password and verify password	System failure; records unsuccessful attempt in the database; an error message is displayed
<u>Step 4</u> Type same passwords in the fields of password and verify password	System indicate success; records successful access in the database the password and verify password are the same, then the system checks the email is already registered correct, then a new user is created .

3. Home Page Acceptance Test

Test-Case ID:	TC-03
Test-Case Name:	Home Page
Pass/Fail Criteria :	The test passes if website is opens.This is the default page of the website. i.e. the first page you see when you enter the website or after you login or register.
Input Data:	Enter name of the site with internet connection.
Test Procedure:	Expected Outcomes :
<u>Step 1</u> Type a doctor name or last name in a search box	System indicate success; records successful access in the database user can search for doctors by first name or last name, when the search is submitted the user is redirected to another page and a list of cars fitting that description is displayed
<u>Step 2</u> Type something other than a doctor name or last name in a search box	System indicate success; System can not find any records in database and an error message is displayed

4. Book Appointment Acceptance Test

Test-Case ID:	TC-04
Test-Case Name:	Book Appointment
Pass/Fail Criteria :	The test passes if appointment is booked.
Input Data:	Numeric and alphabetic key code, date
Test Procedure:	Expected Outcomes :
<u>Step 1</u> Selects a time that is already booked.	System displays “unavailable time”.
<u>Step 2</u> Selects an available time	System records appointment; Displays “appointment booked!”
<u>Step 3</u> Book appointment without login	The user is directed to the log in page after which the user is redirected back to the destined/original page.

5. Contact form Acceptance Test

Test-Case ID:	TC-05
Test-Case Name:	Contact Form
Pass/Fail Criteria :	The test passes if the user fill in all the fields and submit
Input Data:	Numeric and alphabetic key code, date
Test Procedure:	Expected Outcomes :
<u>Step 1</u> Fill in all the fields and submit	When this form is submitted it is stored in the database awaiting reply .

6. Account Profile Acceptance Test

Test-Case ID:	TC-06
Test-Case Name:	Account Profile
Pass/Fail Criteria :	The test passes if user click on account profile.
Input Data:	The test passes if the user click on and enters account profile page.
Test Procedure:	Expected Outcomes :
<u>Step 1</u> Edit Profile	User can view and edit information of profile ; new information is saved and updated in the database.
<u>Step 2</u> View appointments	Appointment information is displayed.
<u>Step 3</u> Edit appointments	The user can change the dates of appointments; this new information is updated and stored in the database.
<u>Step 4</u> Cancel appointment	A field for booking reference is provided, with this booking reference the user's booking is canceled.
<u>Step 5</u> Logout	When this option is clicked the user is logged out and redirected to a more general homepage.

7. Patient Acceptance Test

Test-Case ID:	TC-7
Test-Case Name:	Patient
Pass/Fail Criteria :	The test passes if user (admin) click on patient. In this interface the following functionalities are implemented for the administrator.
Input Data:	user click on and enters patient page.
Test Procedure:	Expected Outcomes :
<u>Step 1</u> View patient	A list of all the patients in the database is displayed and when a patient item on the list is clicked, the patient details is displayed in a form
<u>Step 2</u> Add patient	The administrator can add patients to the database, a form is provided and the patient's details are entered, when the form is submitted the new patient is created and inserted into the database.
<u>Step 3</u> Edit patient information	The administrator can also edit patient information, whatever information entered is saved and updated in the database.
<u>Step 4</u> Delete patient	A delete option is placed beside each patients in the list of patients and when it is clicked that particular patient is deleted.

8. Doctor Acceptance Test

Test-Case ID:	TC-8
Test-Case Name:	Doctor
Pass/Fail Criteria :	The test passes if user (admin) click on doctor part. In this interface the following functionalities are implemented for the administrator.
Input Data:	user click on doctor part.
Test Procedure:	Expected Outcomes :
<u>Step 1</u> View doctor	A list of all the doctors in the database is displayed and when a doctor item on the list is clicked, the doctor details is displayed in a form
<u>Step 2</u> Add doctor	the administrator can add doctors to the database, a form is provided and the doctor's details are entered, when the form is submitted the new doctor is created and inserted into the database.
<u>Step 3</u> Edit doctor information	The administrator can also The administrator can also edit doctor information, whatever information entered is saved and updated in the database.
<u>Step 4</u> Delete doctor	A delete option is placed beside each doctor in the list of doctors and when it is clicked that particular doctor is deleted.

9. Department Acceptance Test

Test-Case ID:	TC-9
Test-Case Name:	Department
Pass/Fail Criteria :	The test passes if user (admin) click on department part. In this interface the following functionalities are implemented for the administrator.
Input Data:	user click on department part.
Test Procedure:	Expected Outcomes :
<u>Step 1</u> View department	A list of all the departments in the database is displayed and when a department item on the list is clicked, the department details is displayed in a form
<u>Step 2</u> Add department	The administrator can add departments to the database, a form is provided and the department's details are entered, when the form is submitted the new department is created and inserted into the database.
<u>Step 3</u> Edit department information	The administrator can also edit department information, whatever information entered is saved and updated in the database.
<u>Step 4</u> Delete department	A delete option is placed beside each departments in the list of departments and when it is clicked that particular department is deleted.

10.Hospital Acceptance Test

Test-Case ID:	TC-10
Test-Case Name:	Hospital
Pass/Fail Criteria :	The test passes if user (admin) click on hospital part. In this interface the following functionalities are implemented for the administrator.
Input Data:	user click on hospital part.
Test Procedure:	Expected Outcomes :
<u>Step 1</u> View hospital	A list of all the hospitals in the database is displayed and when a hospital item on the list is clicked, the hospital details is displayed in a form
<u>Step 2</u> Add hospital	The administrator can add hospitals to the database, a form is provided and the hospital's details are entered, when the form is submitted the new hospital is created and inserted into the database.
<u>Step 3</u> Edit hospital information	The administrator can also edit hospital information, whatever information entered is saved and updated in the database.
<u>Step 4</u> Delete hospital	A delete option is placed beside each hospitals in the list of hospitals and when it is clicked that particular hospital is deleted.

11. Appointment Acceptance Test

Test-Case ID:	TC-11
Test-Case Name:	Appointment
Pass/Fail Criteria :	The test passes if user (admin) click on appointment part. In this interface the following functionalities are implemented for the administrator.
Input Data:	user click on appointment part.
Test Procedure:	Expected Outcomes :
<u>Step 1</u> View Appointment	A list of all the appointments in the database is displayed and when a appointment item on the list is clicked, the appointment details is displayed in a form
<u>Step 2</u> Add Appointment	The administrator can add appointments to the database, a form is provided and the appointment's details are entered, when the form is submitted the new appointment is created and inserted into the database.
<u>Step 3</u> Edit Appointment information	The administrator can also edit appointment information, whatever information entered is saved and updated in the database.
<u>Step 4</u> Delete Appointment	A delete option is placed beside each appointment in the list of h appointment s and when it is clicked that particular appointment is deleted.

12. Make enquiries Acceptance Test

Test-Case ID:	TC-12
Test-Case Name:	Enquiries
Pass/Fail Criteria :	The test passes if it enables the administrator to view and reply enquiries submitted by the patients.
Input Data:	Numeric and alphabetic key code, date
Test Procedure:	Expected Outcomes :
<u>Step 1</u> Reply Enquiries	All the enquiries submitted by patients are displayed in a list with a reply option provided for each enquiry, when it is clicked a form is displayed where the administrator can write a reply to patient, when this form is submitted the content is sent as an email to the patient.

CONCLUSION

In conclusion we implemented all the functionalities stated in the requirement, all were tested and it showed that they are all working properly with the exception of the mail functionality which works perfectly fine on the local server but not on the live server, this is due to the fact that we are using a server that blocks the port for mail.

Also we implemented the system requirement quality like responsiveness so on different devices the system adapts to suit it, and also it works on these browsers like safari, google chrome properly.

References

IEEE. IEEE STD 830-1998 IEEE Recommended Practice for Software Requirements Specifications. IEEE Computer Society, 1998.