Online Clinic Management Platform

Final project Report Submitted to

The Department of Software Engineering

Faculty of Engineering and Information Technology

Al-Azhar University - Gaza

In Partial Fulfillment of the Requirements for the Degree of Bachelor of Science in Software Engineering

By:

Yasser Abdel-Rahmaan Abu Shanab (20191216)

Spring semester Oct 2024

Contents

Table of Figures 3

ACKNOWLEDGEMENTS 4

ABSTRACT 5

Introduction 6

1. Purpose 6

2. Scope 6

3. Overview 7

4- Glossary 7

General Description 8

1- Product Perspective 8

2- Product functions 8

3- User Characteristics 8

4- General Constraints 9

5- Existing problems in this field 9

Methodology 10

1- Waterfall Model 10

2 - Justification of Methodology 10

3- Description of Methodology 11

Requirements 12

1- User requirements 12

1. Functional Requirements: 12

2. Non-Functional Requirements 12

2- System Requirements 13

1. Functional requirements 13

2. Non- functional Requirements 15

3- Detailed Use Case Specification 18

Design Documentation 35

1- Use-Cases Diagram 35

2- Communication Diagrams 36

1- Communication Diagram on login 36

2- Communication Diagram on register 37

3- Communication Diagram on view and edit profile 38

4- Communication Diagram on view and cancel appointment 39

5- Communication Diagram on search doctors 40

6- **Communication Diagram on make enquiry** 40
7- **Communication Diagram on book appointment** 41
8- **Communication Diagram on updating available time** 41
9- **Communication Diagram on view and edit users** 42
10- **Communication Diagram on add users** 43
11- **Communication Diagram on view and delete enquiries** 44
3- Class Diagram 45
4- Entity Relation Diagram 46
5- Sequence Diagrams 47
i. **Sequence diagram on Login** 47
ii. **Sequence diagram on Registration** 48
iii. **Sequence diagram on search** 49
iv. **Sequence diagram on Book appointment** 50
v. **Sequence diagram on Cancel appointment** 51
vi. **Sequence diagram on Add available time** 52

Table of Figures

Figure 1 waterfall model 10
Figure2 Use-Cases Diagram 35
Figure 3 Communication Diagram on login 36
Figure 4 Communication Diagram on register 37
Figure 5 Communication Diagram on view and edit profile 38
Figure 6 Communication Diagram on view and cancel appointment 39
Figure 7 Communication Diagram on search doctors 40
Figure8 Communication Diagram on make enquiry 40
Figure 9 Communication Diagram on book appointment 41
Figure 10 Communication Diagram on updating available time 41
Figure 11 Communication Diagram on view and edit users 42
Figure 12 Communication Diagram on add users 43
Figure 13 Communication Diagram on view and delete enquiries 44
Figure 14 Class Diagram 45
Figure 15 E R Diagram 46
Figure 16 Sequence diagram on Login 47
Figure 17 Sequence diagram on Registration 48
Figure 18 Sequence diagram on search 49
Figure 19 Sequence diagram on Book appointment 50
Figure 20 Sequence diagram on Cancel appointment 51
Figure21 Sequence diagram on Add available time 52

ACKNOWLEDGEMENTS

First of all, I thank Allah for giving me the ability to reach this level in my life, Insha'Allah I`ll be able to help the people in my society and I`ll be able to finish this part of project.

Thanks also for the instructor Dr. Abdelbaset Al-Masri who will supervise my project.

Hopefully this never-ending nightmare will end soon, then we`ll rebuild Gaza again Insha'Allah. May Allah have mercy on the martyrs, heal the wounded and inspire us patience and solace.

ABSTRACT

Online Clinic Management is one of the most important modern medical tools, as doctor's information and appointments details are transferred via technology means from the clinic to the patients. In light of the current circumstances of the ongoing war here in Gaza and the massive destruction the occupation caused, having an appointment with a doctor is almost impossible due to the destruction, one of the few solutions we have to offer for our people is an online booking system. (OCM) is an interactive platform affiliated to a specific clinic that supports patinas, it is a web application that will be developed in PHP and powerful MYSQL database backend. To implement (OCM) application, users do not need expensive hardware and software, they just need an internet connection and Smartphone or desktops. (OCM) is very organized, user-friendly and it is easy access at any time, the design of Interface will encourage patinas stay in contact with the doctors. It is providing various services to patinas and doctors and It is easy to use. it will be not hard to use and it will provide helpful health education topics to help visitors and improve their knowledge.

Introduction

The purpose of this document is to describe the requirements for the Online Clinic Management System (OCM). 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.

1. Purpose

Gaza is a region suffering from a lack of healthcare infrastructure, making access to medical services difficult, especially for those living in remote areas. The Online Clinic Platform project offers an innovative solution to this problem, allowing easier and more efficient access to healthcare services. It aims to improve access to healthcare services and medical care via the internet, regardless of geographic location, and increase the efficiency of clinics by providing a website for managing appointments.

The platform provides a fast, user-friendly services improving patient experience. The platform features online appointment booking, making it easier for patients to schedule appointments with specialists in participating clinic. The clinic management system offers clinics a comprehensive tool for managing appointments, while patients can access reliable health information through the platform.

2. Scope

The proposed software product Online Clinic Management System (OCM). In this project we are going to design and build a fully functional web based Online appointment booking System. Customers can be able to book doctor's appointment anywhere in the world, this can be done via web browser. Doctors can also login 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. Overview

This Software Requirements Specification (SRS) is the requirements work product that formally specifies Online Clinic Management System (OCMS). It includes the results of 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 Clinic Management System (OCMS). 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.

4- Glossary

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.
Web browser	A software application for retrieving and traversing information resources on world wide web
Administrative staff	Person responsible for the management of operations.
Graphic user interface	A visual way of interacting with a computer using items

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

. doctor and the patinate can log into the platform using the username/password.

. patinate can book an appointment.

. patinate can pre-written topics for healthcare.

. patinate can cancel a booked appointment if needed.

. doctors can manage their booking slots.

. doctor can view the patinate information.

. doctor and patinate users can change their information.

. The admin can add, edit, and delete patinates and doctors, appointment scheduled for patinate and doctor.

3- User Characteristics

The system will be used in a Clinic. The doctors and patients will be the main users.

Given the condition that not all the users are computer-literate. The system is also designed to be user-friendly.

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

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

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.
- The Clinic must have enough trained staff to take care of the system.

5- Existing problems in this field

The main issue with the patient appointment scheduling system in outpatient clinics is:
Lack of a Unified Electronic System: clinics in Gaza relies on a manual system for scheduling appointments, leading to:

- Wasted time and effort for patients and staff.
- Potential for human errors.
- Difficulty in tracking and following up on appointments.
- Inaccurate information about appointment availability.
- The current system's inability to meet the hospital's needs: The manual system cannot handle the large volume of appointment requests or provide detailed patient information.

Methodology

1- Waterfall Model

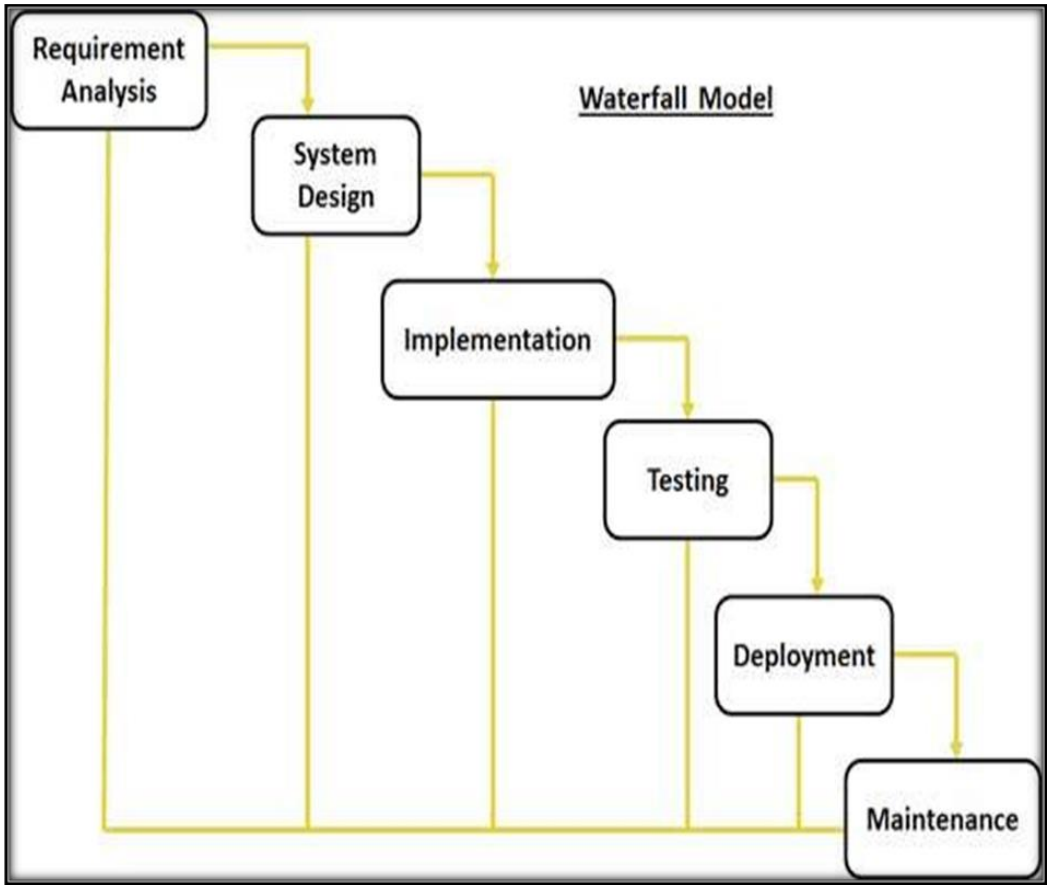


Figure 1 waterfall model

2 - Justification of Methodology

Every software developed is different and requires a suitable SDLC approach to be followed based on the internal and external factors. Some situations where the use of Waterfall model is most appropriate are:

- Requirements are very well documented, clear and fixed.
- Product definition is stable.
- Technology is understood and is not dynamic.
- The project is short.
- Simple and easy to understand and use
- Easy to manage due to the rigidity of the model, each phase has specific deliverables and a review process.

- Phases are processed and completed one at a time.
- Easy to arrange tasks.

3- Description of Methodology

The sequential phases in Waterfall model are:

Requirement Gathering and analysis: All possible requirements of the system to be developed are captured in this phase and documented in a requirement specification doc.

System Design: The requirement specifications from first phase are studied in this phase and system design is prepared. System Design helps in specifying hardware and system requirements and also helps in defining overall system architecture.

Implementation: With inputs from system design, the system is first developed in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality which is referred to as Unit Testing.

Integration and Testing: All the units developed in the implementation phase are integrated into a system after testing of each unit. Post integration the entire system is tested for any faults and failures.

Deployment of system: Once the functional and nonfunctional testing is done, the product is deployed in the customer environment or released into the market.

Maintenance: There are some issues which come up in the client environment. To fix those issues patches are released. Also, to enhance the product some better versions are released. Maintenance is done to deliver these changes in the customer environment.

All these phases are cascaded to each other in which progress is seen as flowing steadily downwards (like a waterfall) through the phases. The next phase is started only after the defined set of goals are achieved for previous phase and it is signed off, so the name "Waterfall Model". In this model phases do not overlap.

Requirements

1- 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 enable patients and doctors to switch between dark and light mode.
- REQ5. The system should allow Patients to book appointment.
- REQ6. The system should allow Patients to make enquiries.
- REQ7. The system should allow Patients to search for available doctors.
- REQ8. The system should allow Patients to modify or cancel their appointment.
- REQ9. The system should allow Patients and Doctors to view and modify their profile.
- REQ10. The system should allow Doctors to set their available time.
- REQ11. The system should enable administrator to log in.
- REQ12. The system should allow administrator to manage users
- REQ13. The system should enable administrator to view enquiries.
- REQ14. The system should allow administrator to delete past
- appointments.
- REQ15. 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.

2- 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 enable patients and doctors to switch between dark and light mode.

- switch between light mode when user clicks on theme button.

1.5. 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.

1.6. 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.
- The system should allow patients to search for available doctors of a particular field.

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.

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.

1.8. 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.

1.9. 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.

1.10. 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.

1.11. 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.

1.12. 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.

1.13. 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.

3- Detailed Use Case Specification

Element	Description
Use Case UC-1:	Login
Initiating Actor:	Patient, Doctor, Database administrator
Actor's Goal:	To log into account
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) :	← 1. System shows a warning message that username or password entered was wrong and asked for entering information again

Element	Description
Use Case UC-2:	Register
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.
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) :	← 1. System shows a warning message that username or password entered was wrong and asked for entering information again.

Element	Description
Use Case UC-3:	Book Appointment
Initiating Actor:	Patient
Actor's Goal:	To make online doctor appointment
Preconditions:	User must create an account and logged in
Post conditions:	The system should book at appointment.
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) :	→ 1. User clicks on "Cancel Reservation" button ← 2. System view home page.

Element	Description
Use Case UC-4:	View profile.
Initiating Actor:	Patient, Doctor
Actor's Goal:	To view personal information.
Preconditions:	The user must log in to the system
Post conditions:	The system should display users information.
Flow of Events for Main Success Scenario:	→ 1. User clicks on account page. ← 2. System displays user's personal account page.
Flow of Events for Extensions (Alternate Scenarios) :	

Element	Description
Use Case UC-5:	Set available time
Initiating Actor:	Doctor
Actor's Goal:	To set available time slot in schedule
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:	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.
Flow of Events for Extensions (Alternate Scenarios) :	

Element	Description
Use Case UC-6:	View Appointments
Initiating Actor:	Patient, Doctor
Actor's Goal:	To view scheduled appointments.
Preconditions:	The user must log in to the system
Post conditions:	The system should display user's appointments.
Flow of Events for Main Success Scenario:	→ 1. User clicks on appointments button. ← 2. System shows appointments page.
Flow of Events for Extensions (Alternate Scenarios) :	

Element	Description
Use Case UC-7:	Cancel appointments
Initiating Actor:	Patient, Doctor
Actor's Goal:	To cancel previously booked appointment
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) :	→ 1. User selects NO on confirmation box. ← 2. System displays initial appointment schedule.

Element	Description		
Use Case UC-8:	Search doctor		
Initiating Actor:	Patient		
Actor's Goal:	To search for a particular doctor		
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) :	←	1.	System displays "doctor doesn't exist"

Element	Description
Use Case UC-9:	view users
Initiating Actor:	Database administrator
Actor's Goal:	To view users in the system
Preconditions:	The user must log in to the system
Post conditions:	The system must show a list of users
Flow of Events for Main Success Scenario:	→ 1. User clicks on doctor/patient page. ← 2. System displays list of doctor/patient.
Flow of Events for Extensions (Alternate Scenarios) :	

Element	Description
Use Case UC-10:	Manage & Access Data Base
Initiating Actor:	Database Administrator
Actor's Goal:	To managing data, add, edit, update or delete data
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
Flow of Events for Extensions (Alternate Scenarios) :	

Element	Description
Use Case UC-11:	Make Enquiries.
Initiating Actor:	Patient
Actor's Goal:	To make enquiries.
Preconditions:	The user must log in to the system
Post conditions:	The system should display a verification message to ensure enquiry made.
Flow of Events for Main Success Scenario:	→ 1. User clicks on enquiry page. ← 2. System displays enquiry page. → 3. User fills feedback/complain field. → 4 User fills email field. ← 5. System displays "enquiry sent".
Flow of Events for Extensions (Alternate Scenarios) :	→ 1. User enters invalid email. ← 2. System displays " invalid email".

Element	Description
Use Case UC-12:	View Enquiries.
Initiating Actor:	Admin
Actor's Goal:	To view enquiries.
Preconditions:	The user must log in to the system
Post conditions:	The system should display enquiries
Flow of Events for Main Success Scenario:	→ 1. User clicks on view enquiries page. ← 2. System displays enquiries page.
Flow of Events for Extensions (Alternate Scenarios) :	

Element	Description
Use Case UC-13:	Delete users
Initiating Actor:	Database administrator
Actor's Goal:	To delete user from the system
Preconditions:	The user must log in to the system
Post conditions:	The system must show a message to verify that user has been deleted
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 delete doctor/patient ← 4. System verification alert. → 4. User enters delete. ← 5. System displays a message to show that user has been deleted.
Flow of Events for Extensions (Alternate Scenarios) :	← 1. System verification alert. → 2. User enters cancel. ← 3. System displays list of doctor/patient

Element	Description
Use Case UC-14:	Edit user's information
Initiating Actor:	Database administrator
Actor's Goal:	To edit user to information
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 edit doctor/patient ← 4. System displays form. → 4. User enters information and clicks on save button. ← 5. System displays a message to show that user has been updated.
Flow of Events for Extensions (Alternate Scenarios) :	

Element	Description
Use Case UC-15:	add users
Initiating Actor:	Database administrator
Actor's Goal:	To add user to system
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.
Flow of Events for Extensions (Alternate Scenarios) :	

Element	Description
Use Case UC-16:	modify profile.
Initiating Actor:	Patient, Doctor
Actor's Goal:	To change personal information.
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".
Flow of Events for Extensions (Alternate Scenarios) :	

Element	Description
Use Case UC-17:	Delete Enquiries.
Initiating Actor:	Admin
Actor's Goal:	To delete enquiries.
Preconditions:	The user must log in to the system
Post conditions:	The system should delete enquiries
Flow of Events for Main Success Scenario:	→ 1. User clicks on delete enquiries page. ← 2. System displays enquiries deleted message.
Flow of Events for Extensions (Alternate Scenarios) :	

Design Documentation

1- Use-Cases Diagram

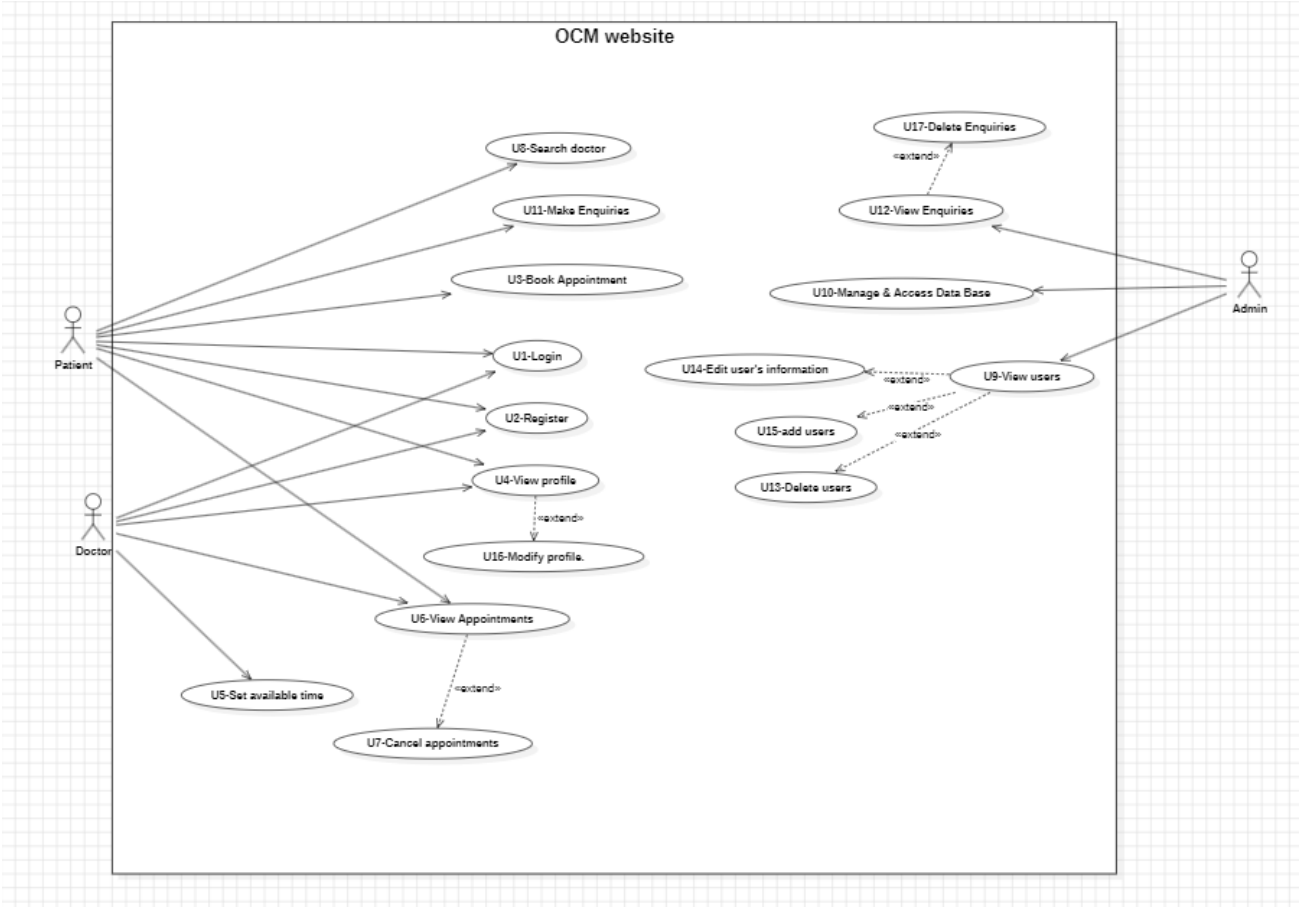


Figure 2 Use-Cases Diagram

2- Communication Diagrams

1- Communication Diagram on login

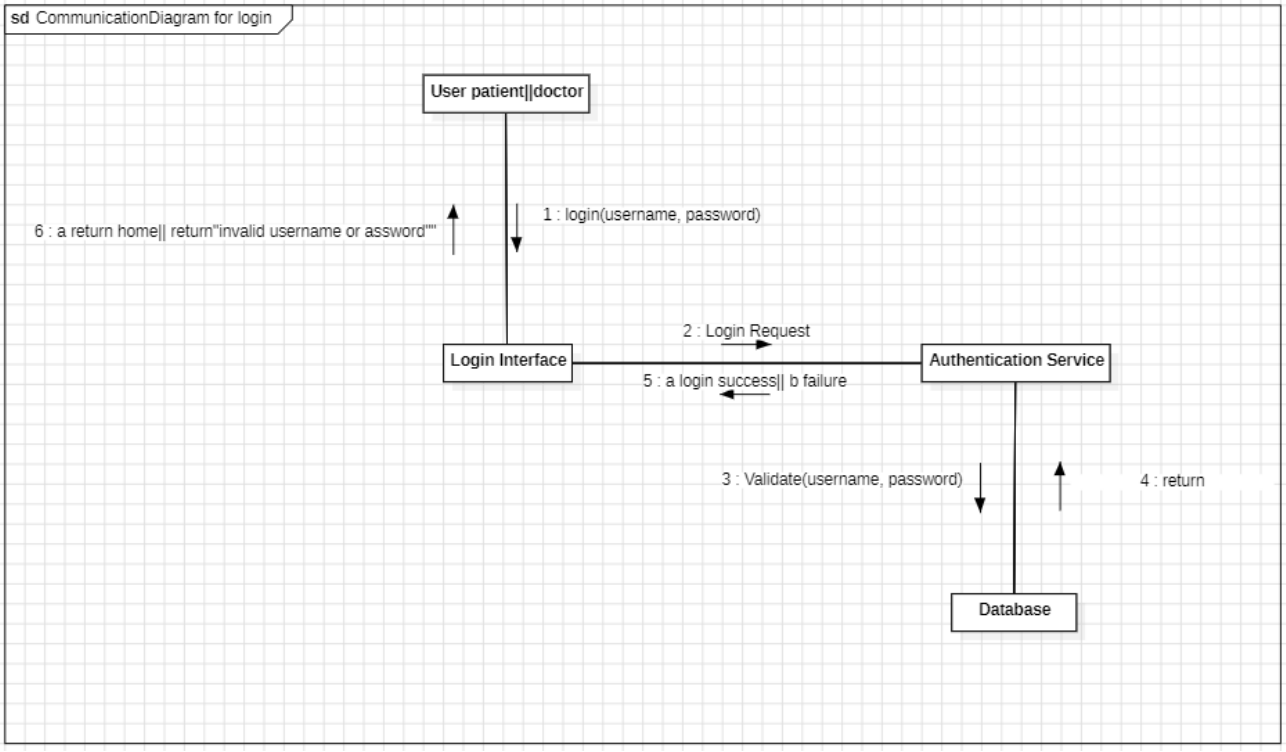


Figure 3 Communication Diagram on login

2- Communication Diagram on register

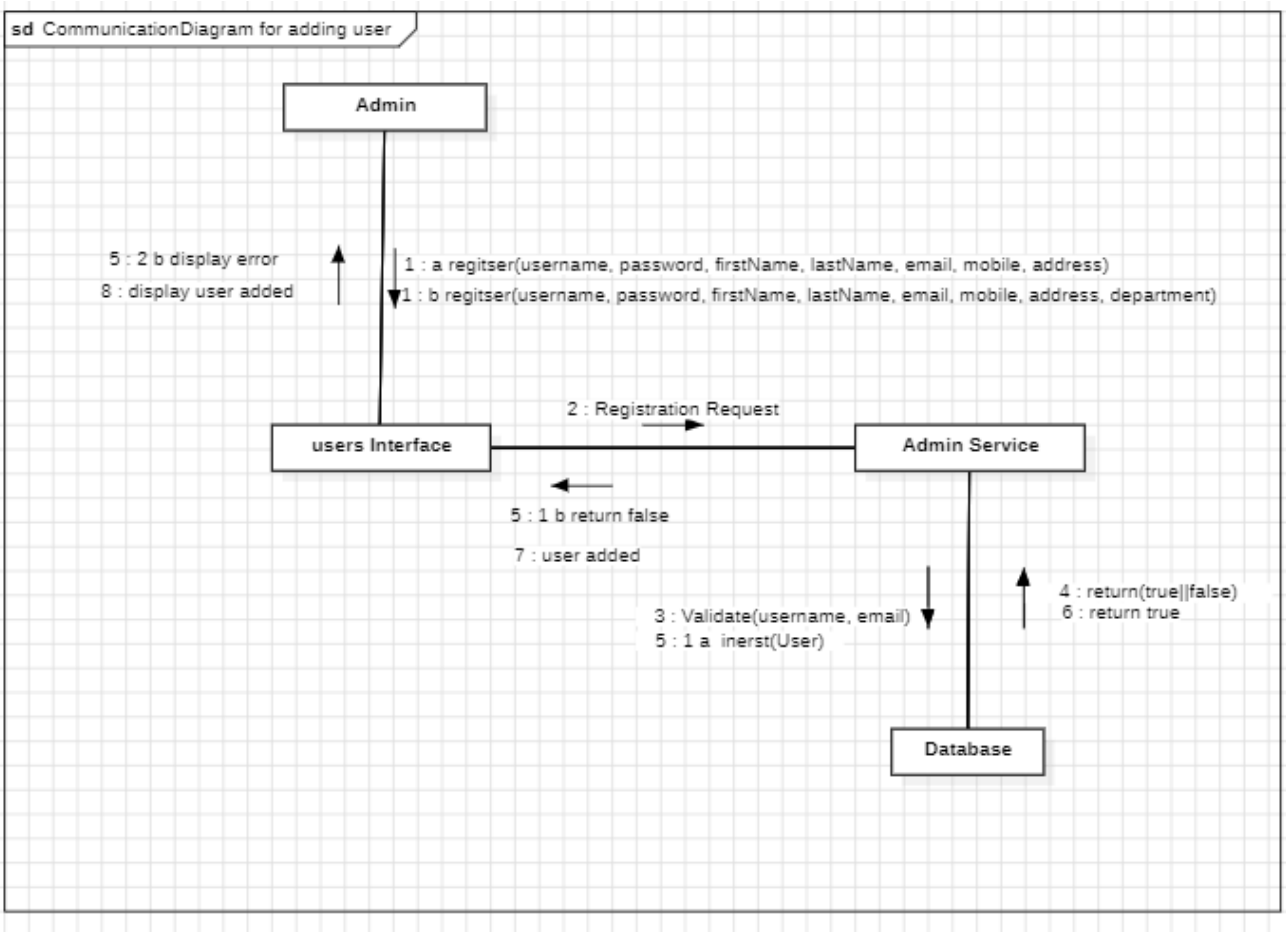


Figure 4 Communication Diagram on register

3- Communication Diagram on view and edit profile

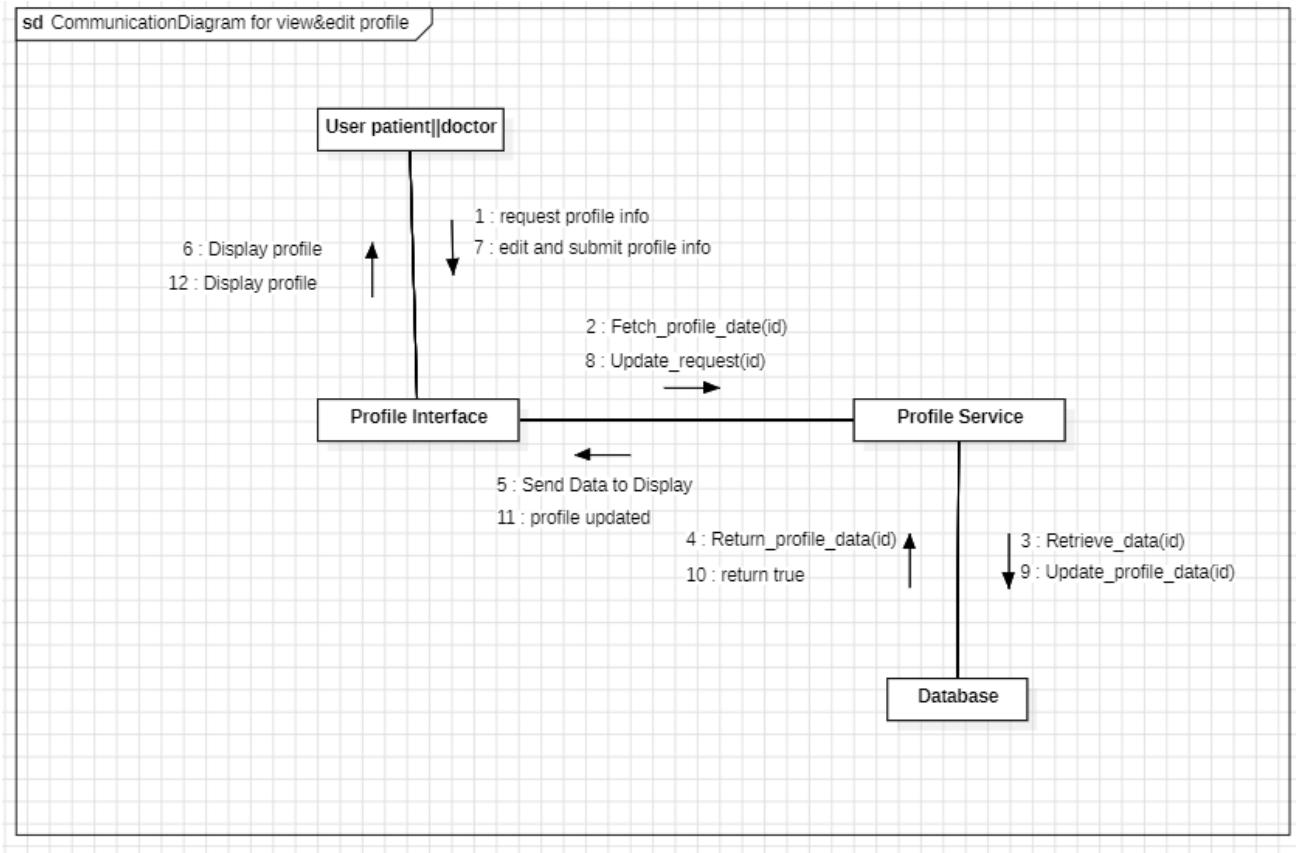


Figure 5 Communication Diagram on view and edit profile

4- Communication Diagram on view and cancel appointment

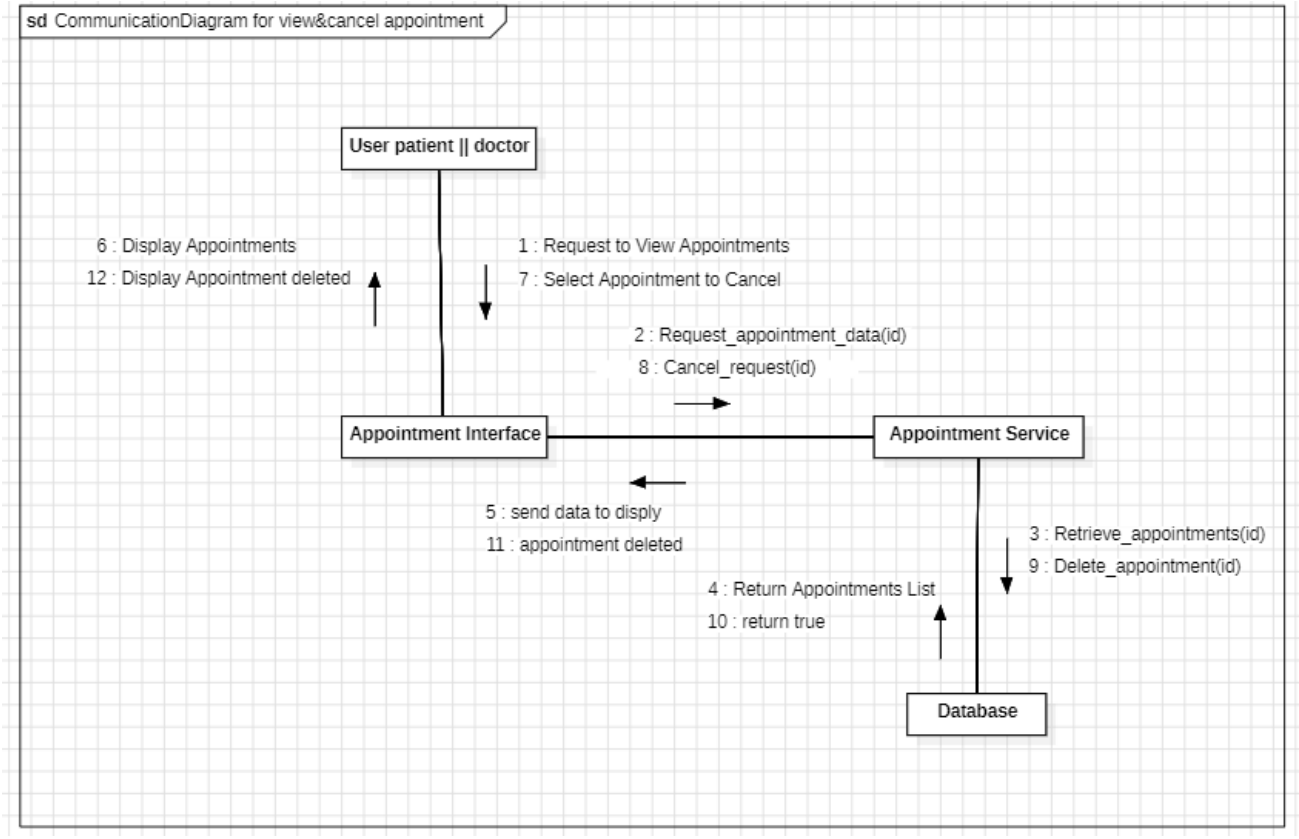


Figure 6 Communication Diagram on view and cancel appointment

5- Communication Diagram on search doctors

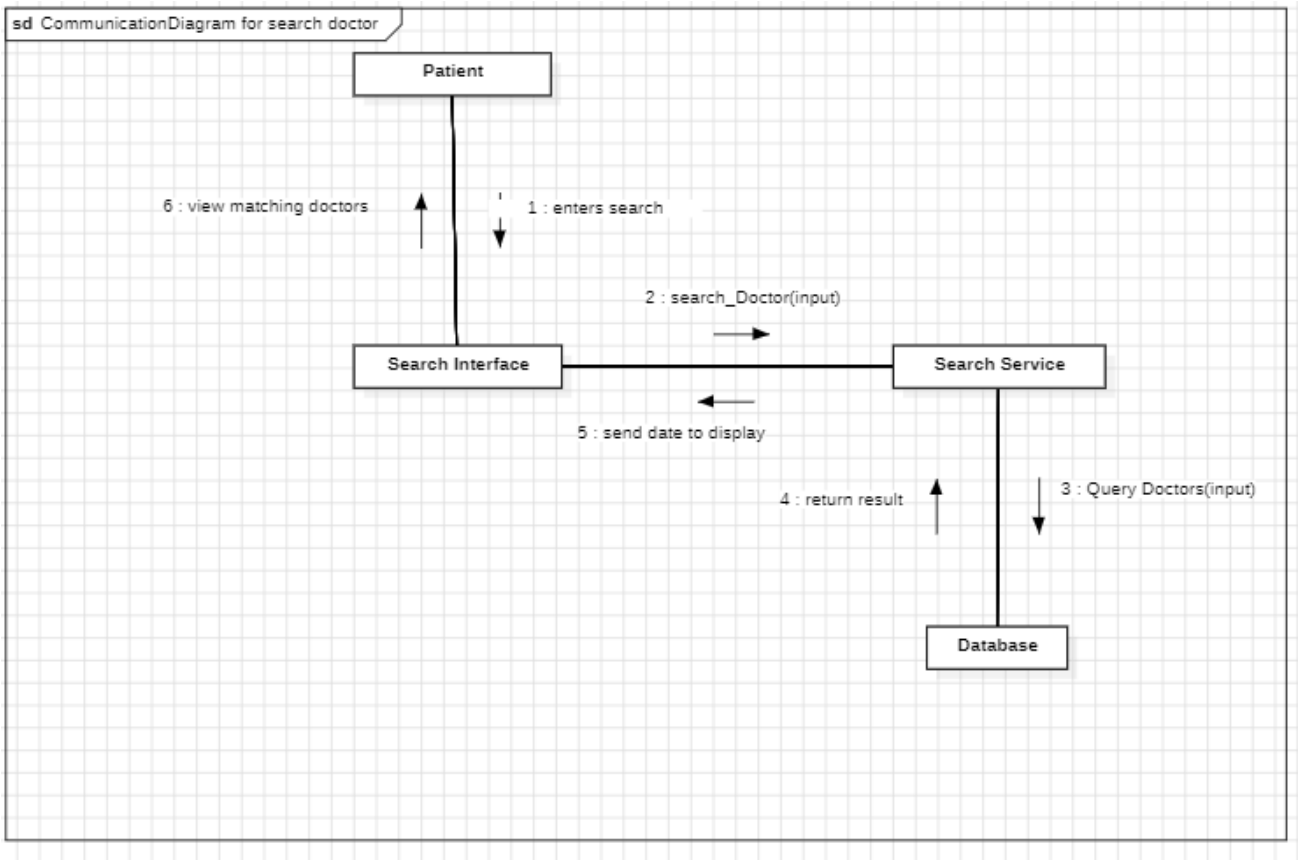
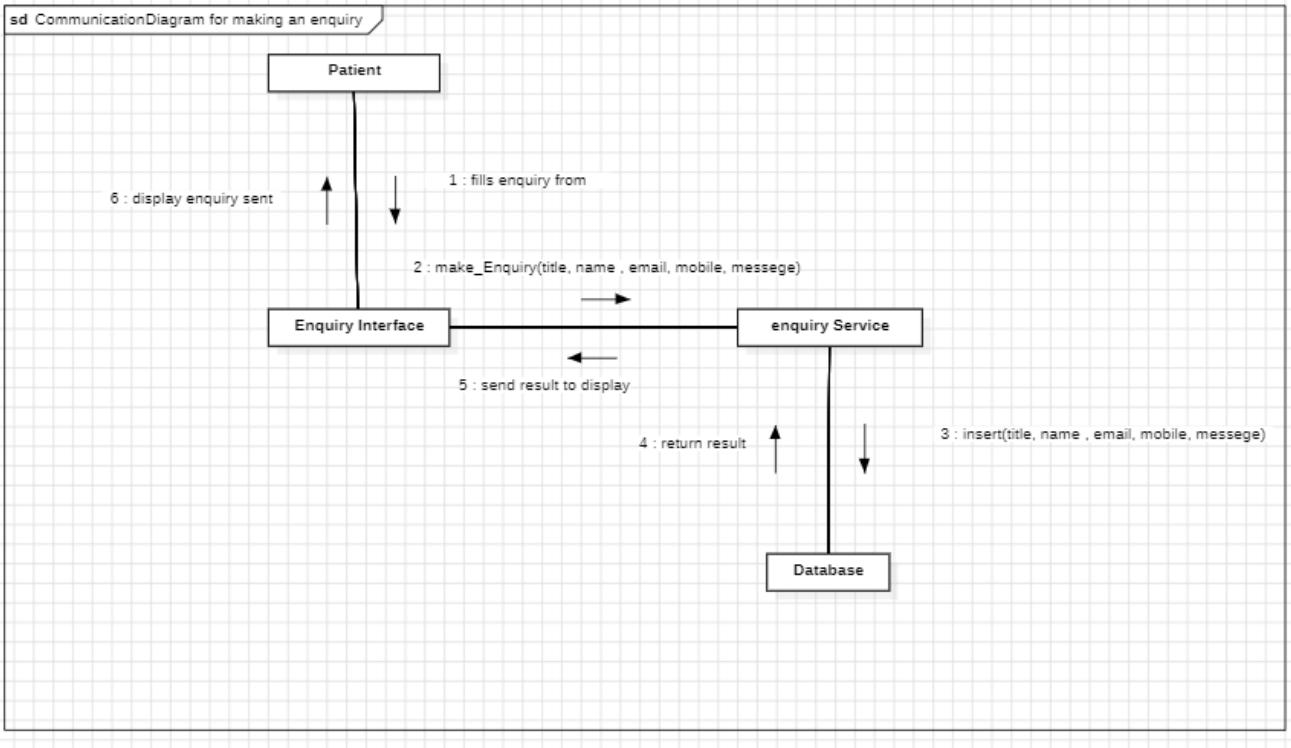


Figure 7 Communication Diagram on search doctors

6- Communication Diagram on make enquiry



8 Communication Diagram on make enquiry Figure

7- Communication Diagram on book appointment

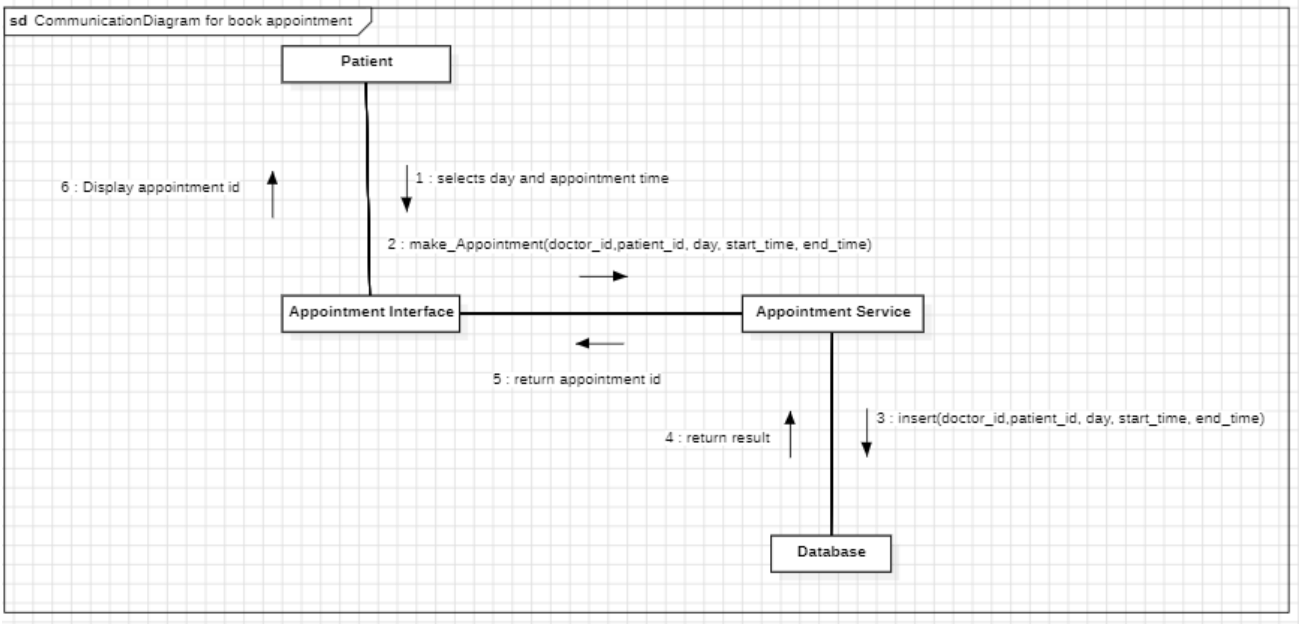


Figure 9 Communication Diagram on book appointment

8- Communication Diagram on updating available time

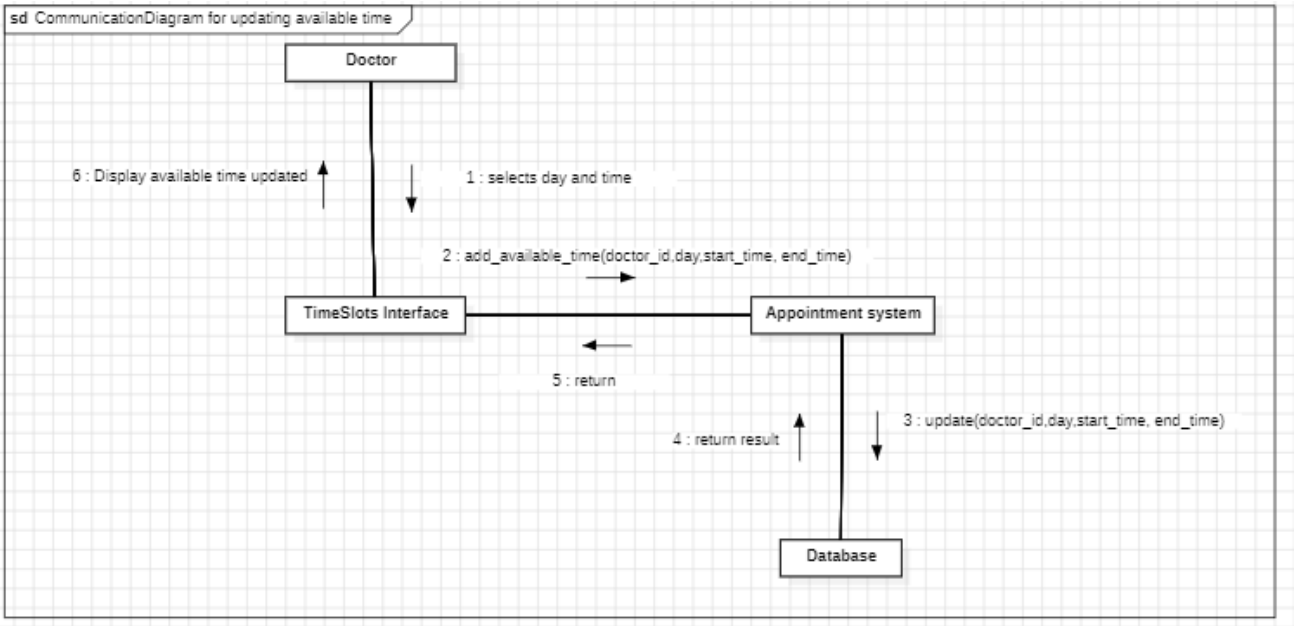


Figure 10 Communication Diagram on updating available time

9- Communication Diagram on view and edit users

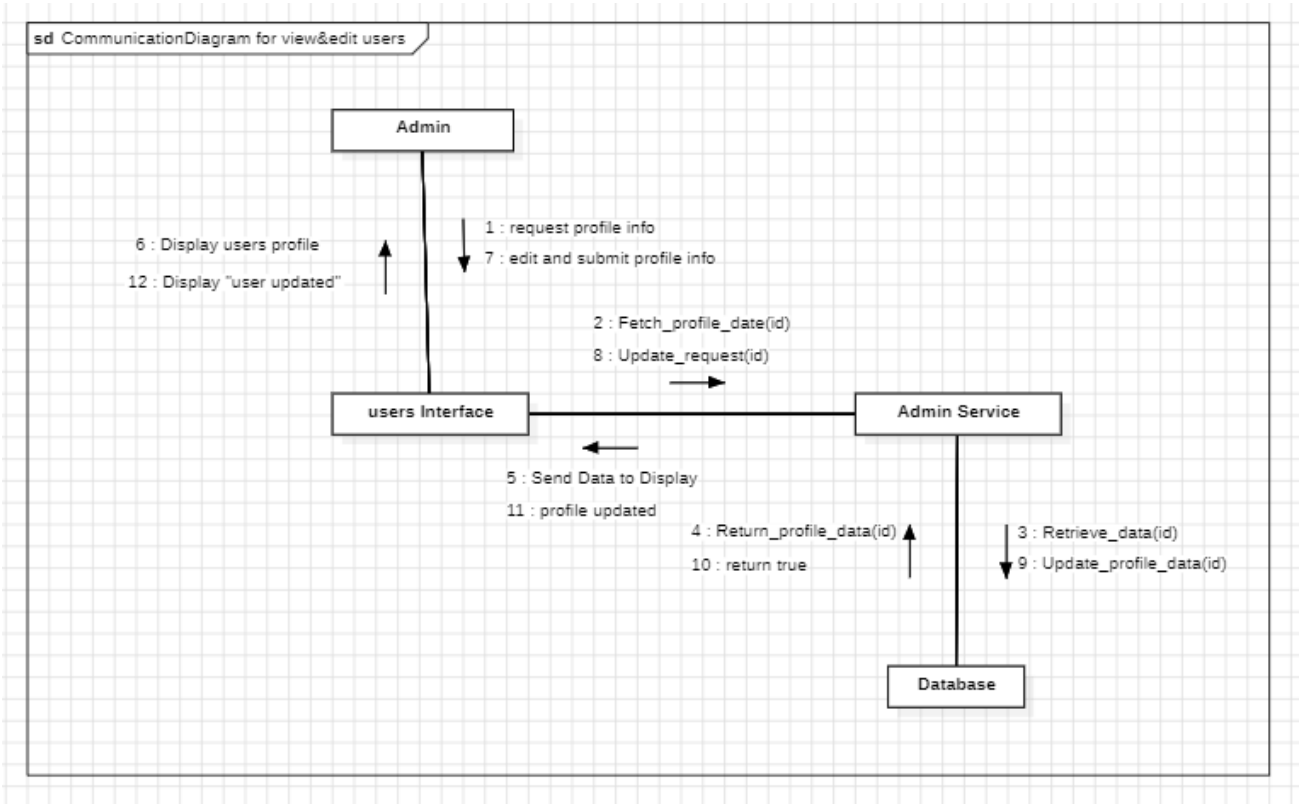


Figure 11 Communication Diagram on view and edit users

10- Communication Diagram on add users

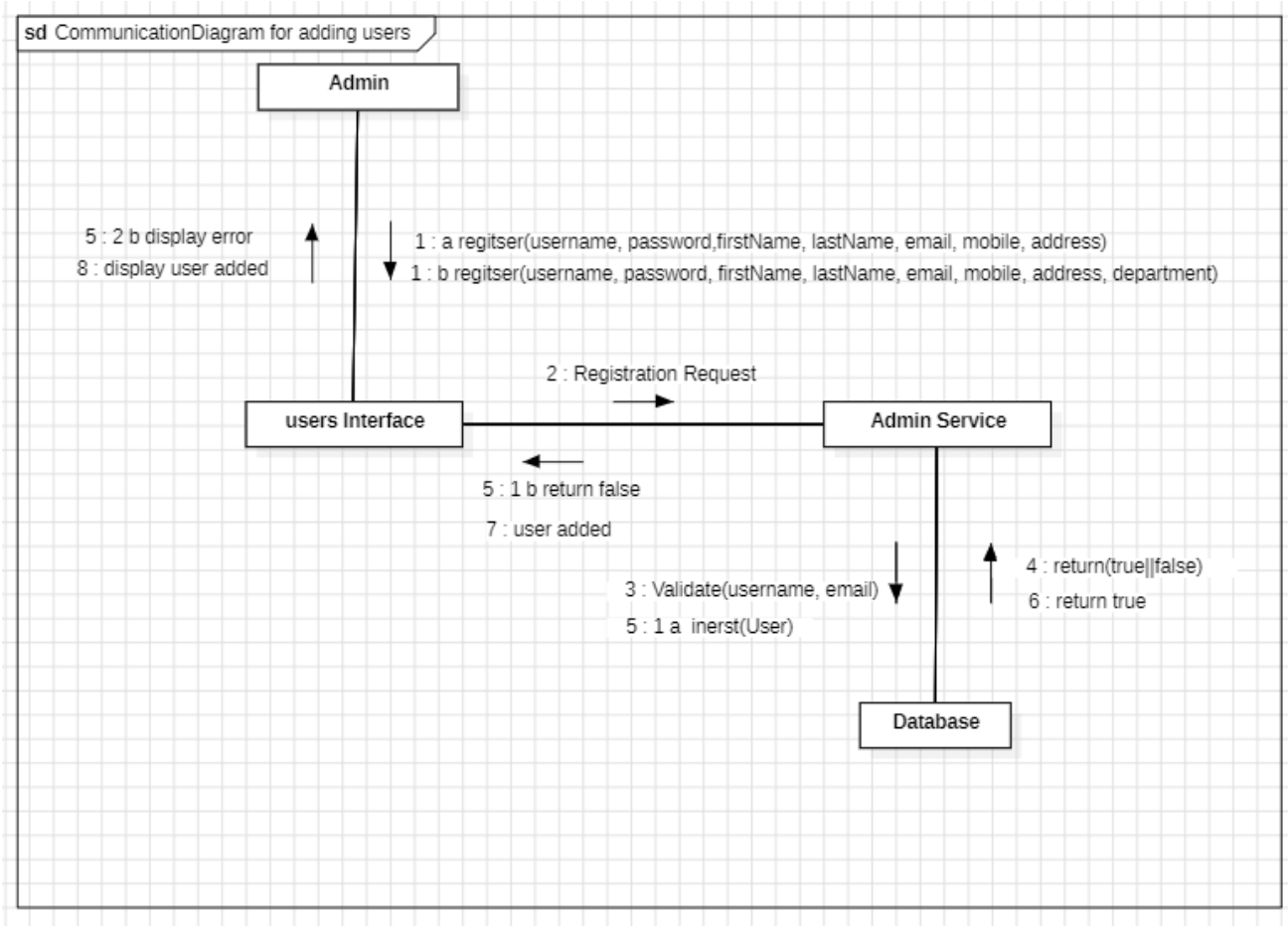


Figure 12 Communication Diagram on add users

11- Communication Diagram on view and delete enquiries

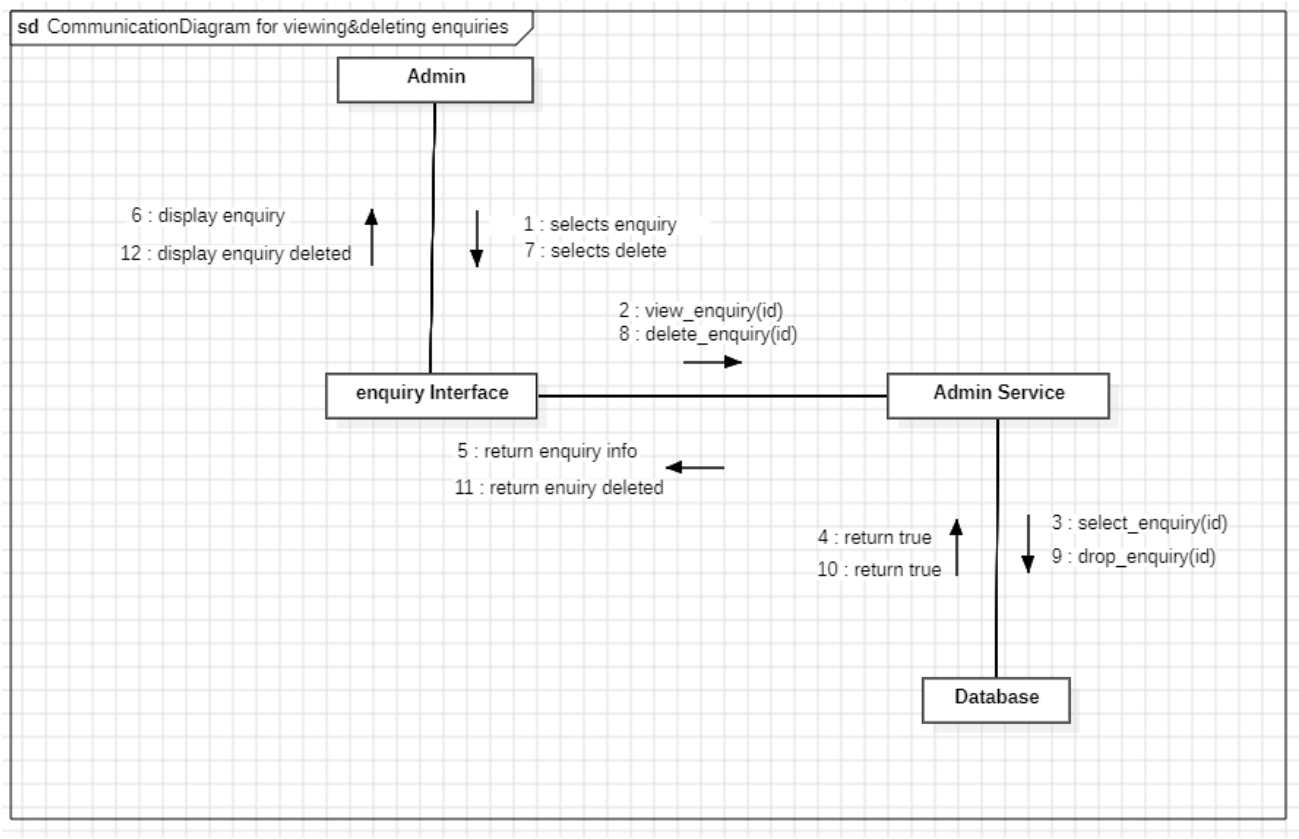


Figure 13 Communication Diagram on view and delete enquiries

3- Class Diagram

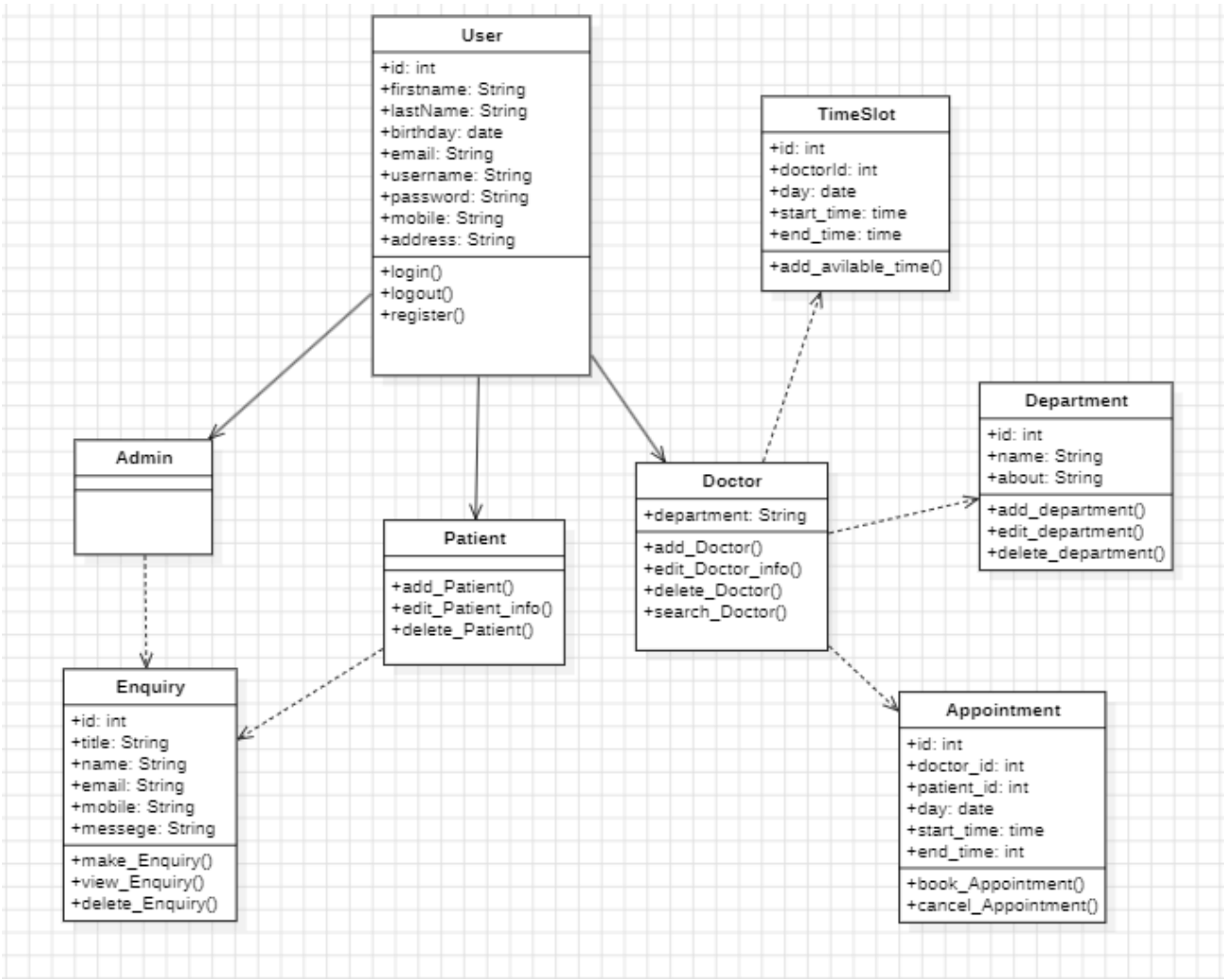


Figure 14 Class Diagram

4- Entity Relation Diagram

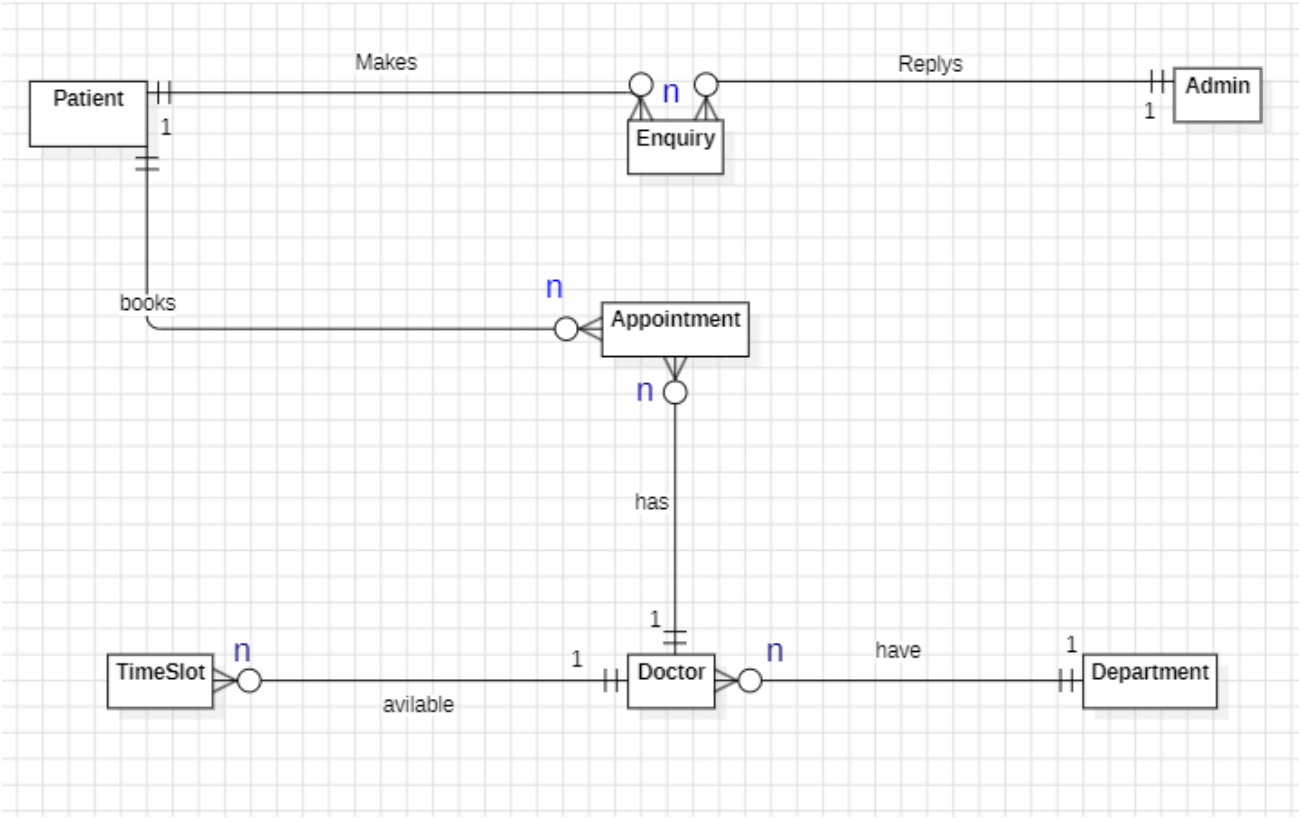


Figure 15 E R Diagram

5- Sequence Diagrams

i. Sequence diagram on Login

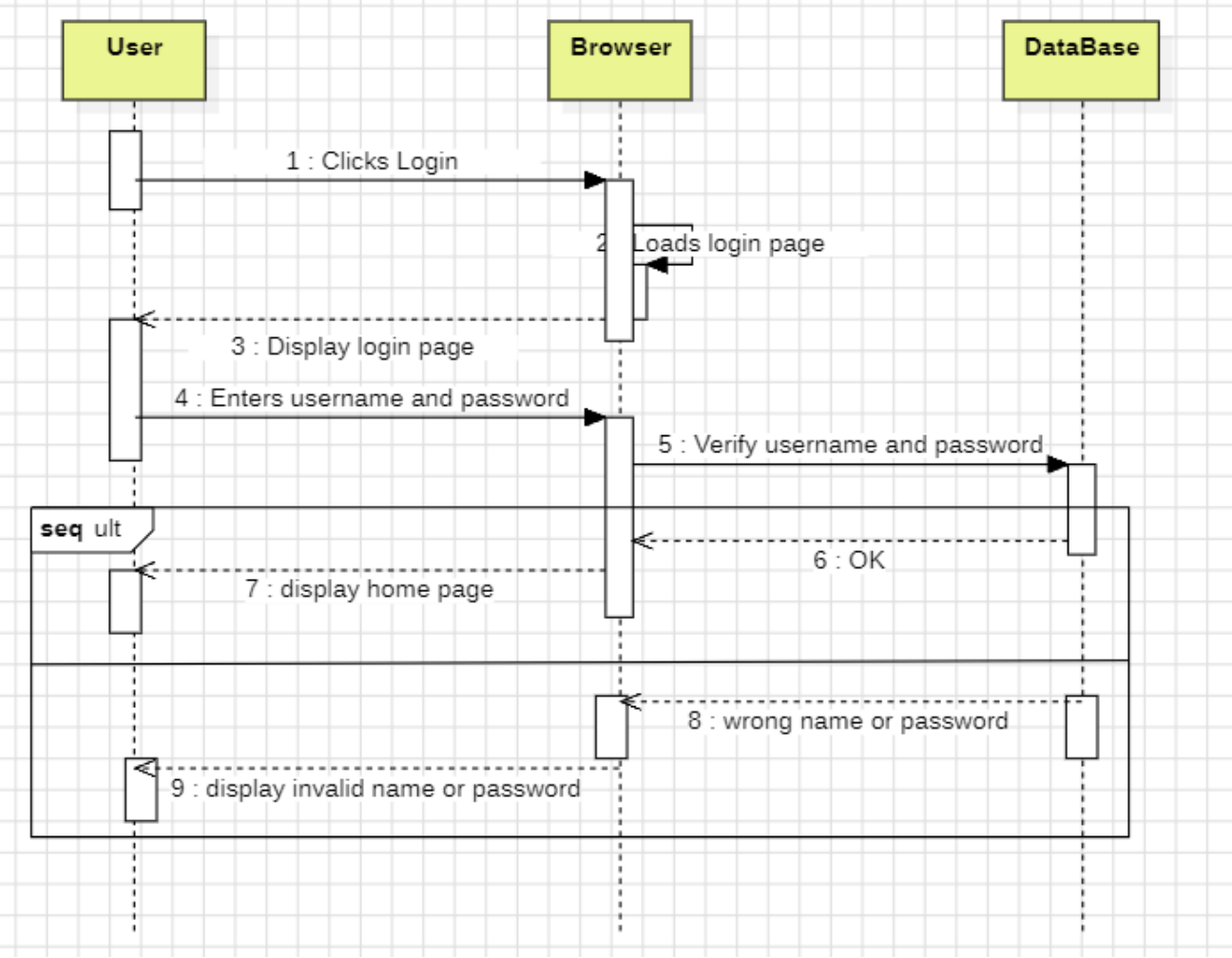


Figure 16 Sequence diagram on Login

ii. Sequence diagram on Registration

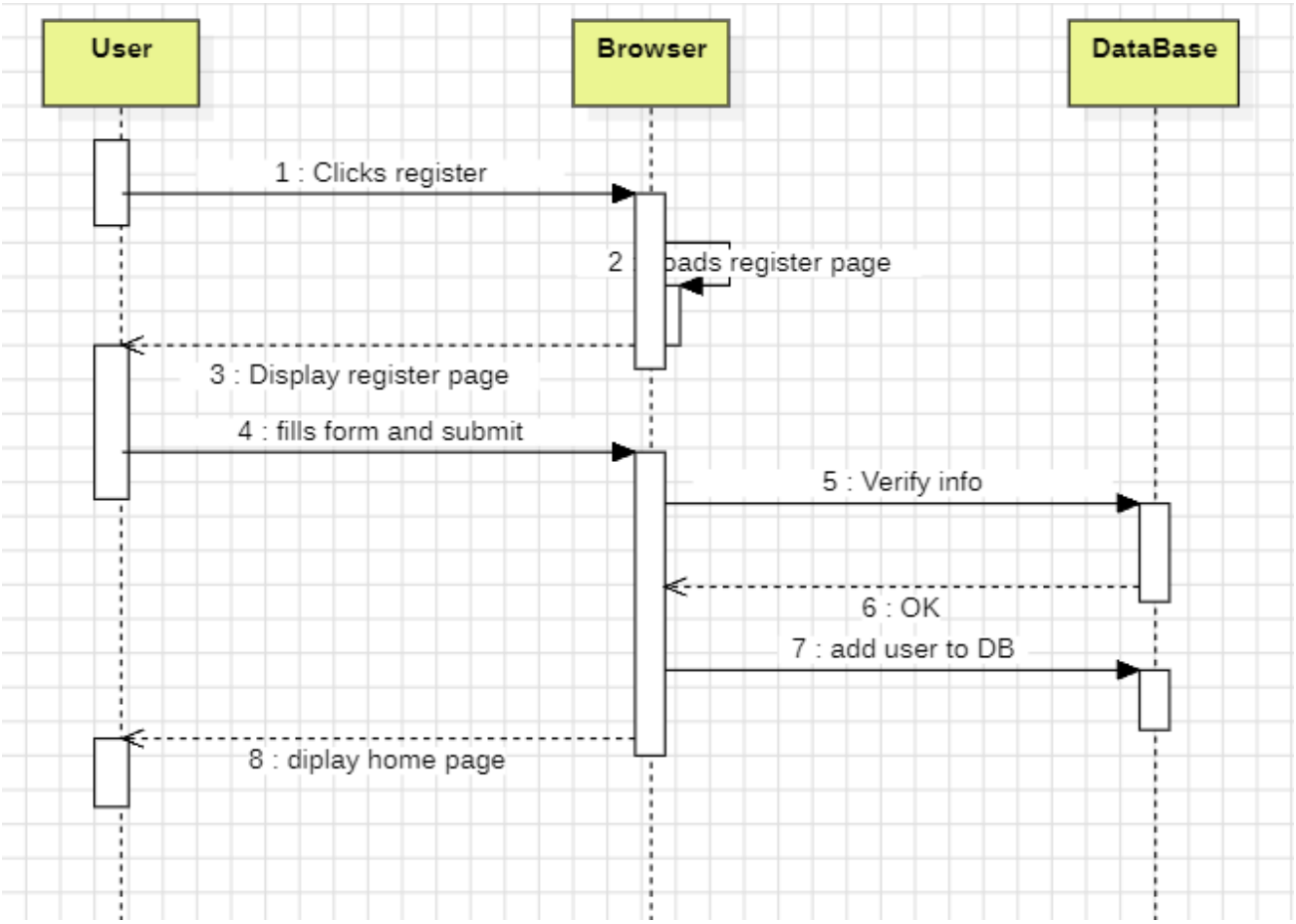


Figure 17 Sequence diagram on Registration

iii. Sequence diagram on search

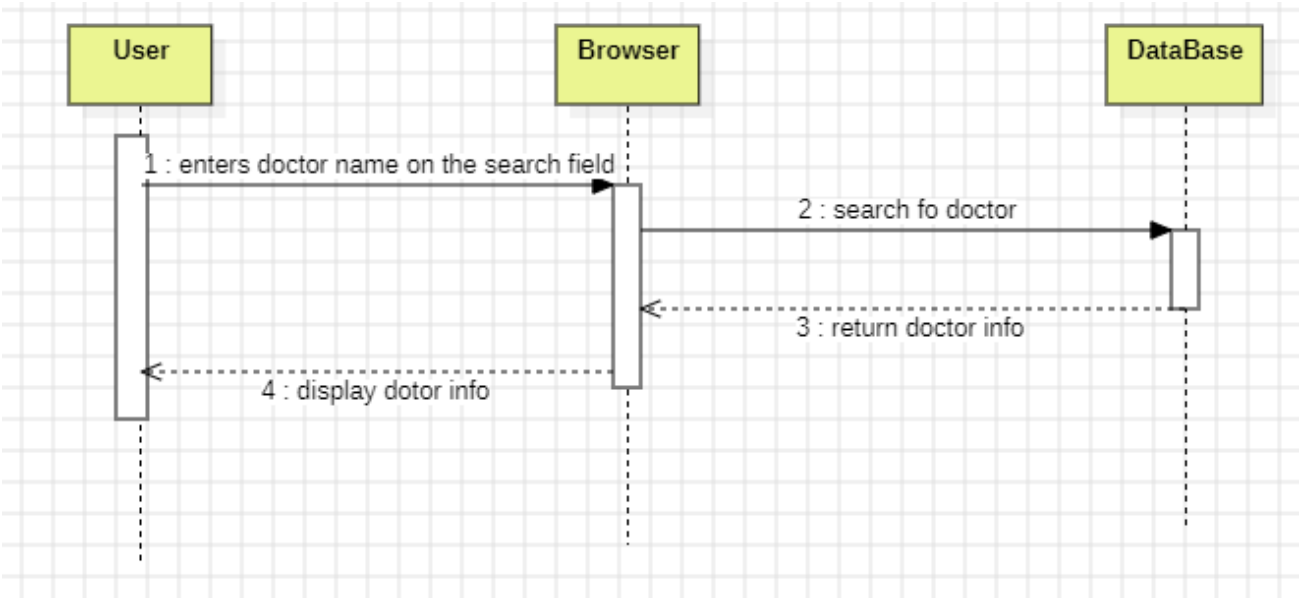


Figure 18 Sequence diagram on search

iv. Sequence diagram on Book appointment

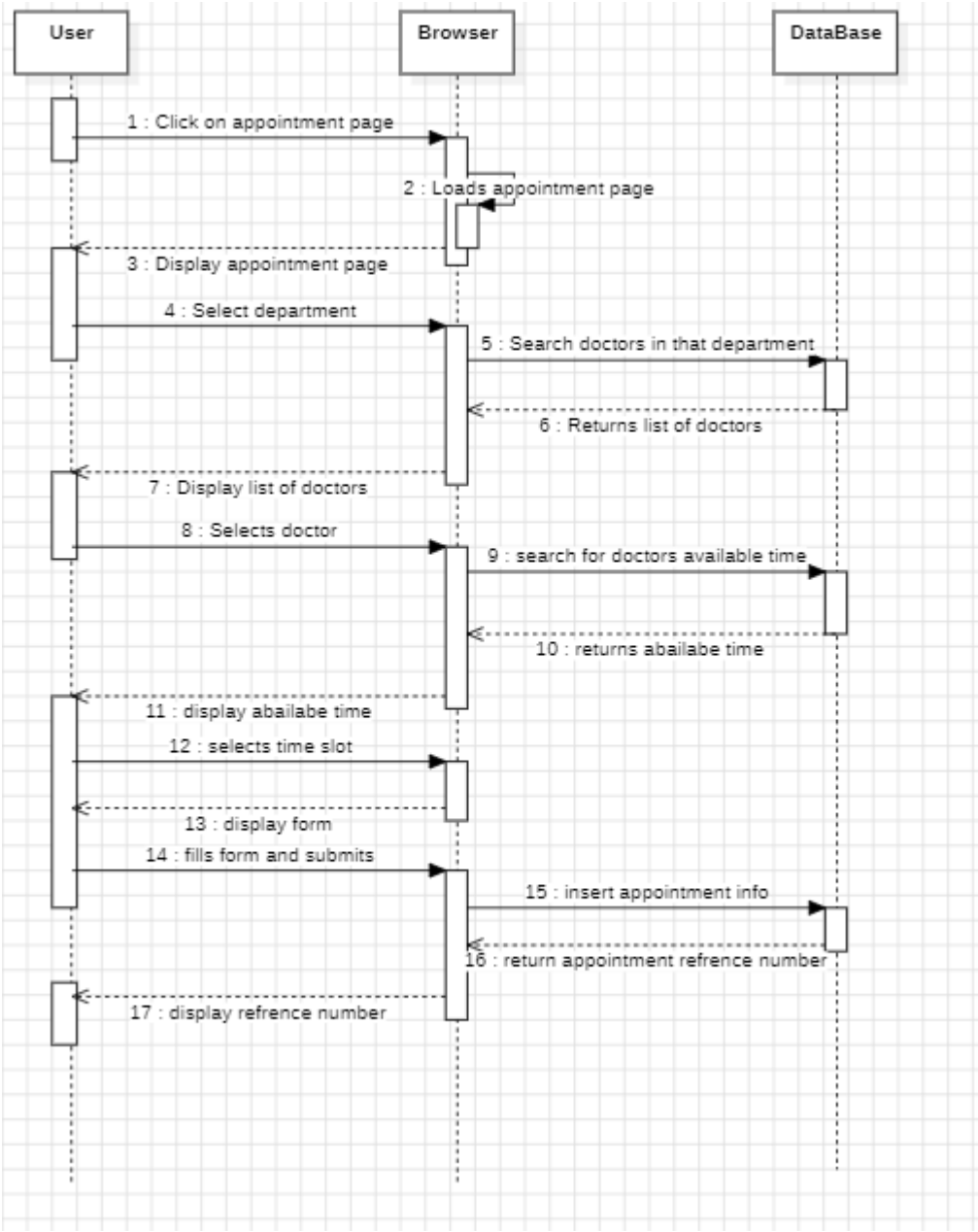


Figure 19 Sequence diagram on Book appointment

v. Sequence diagram on Cancel appointment

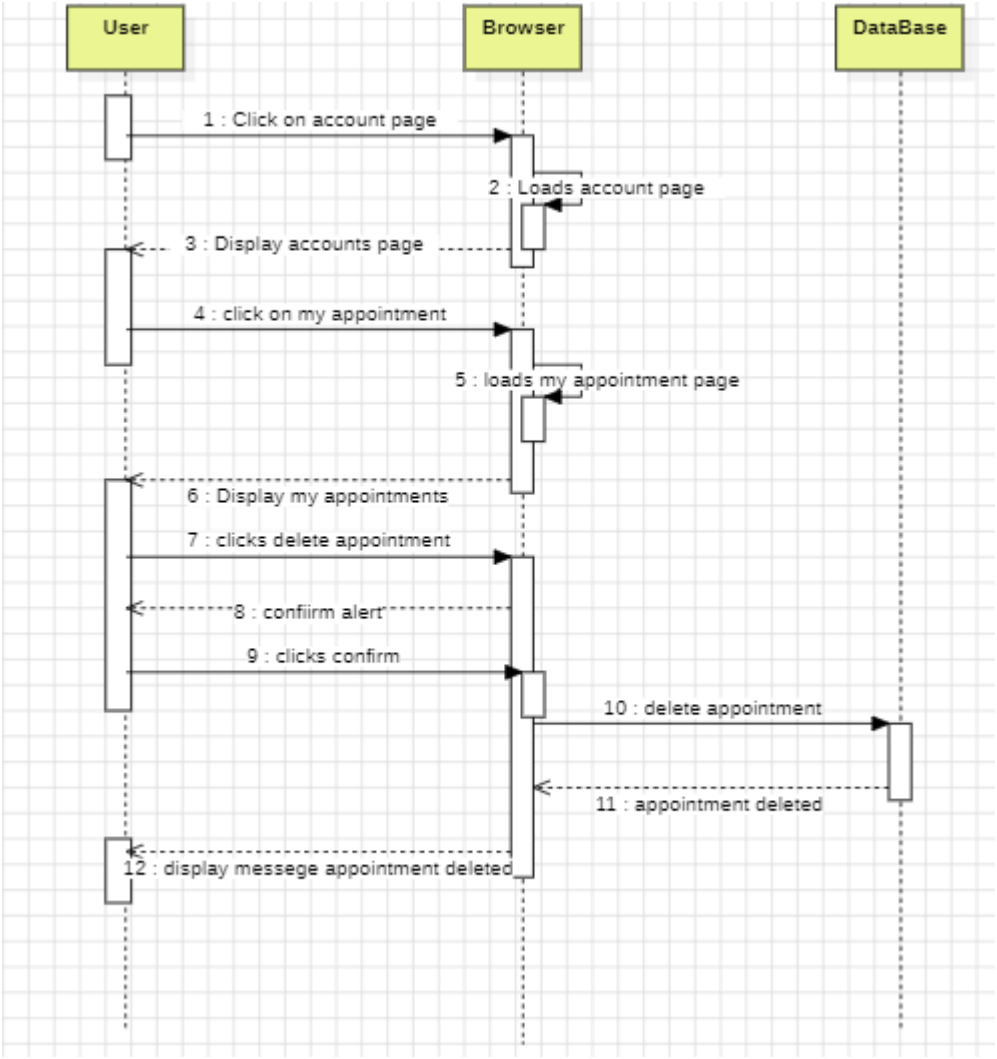
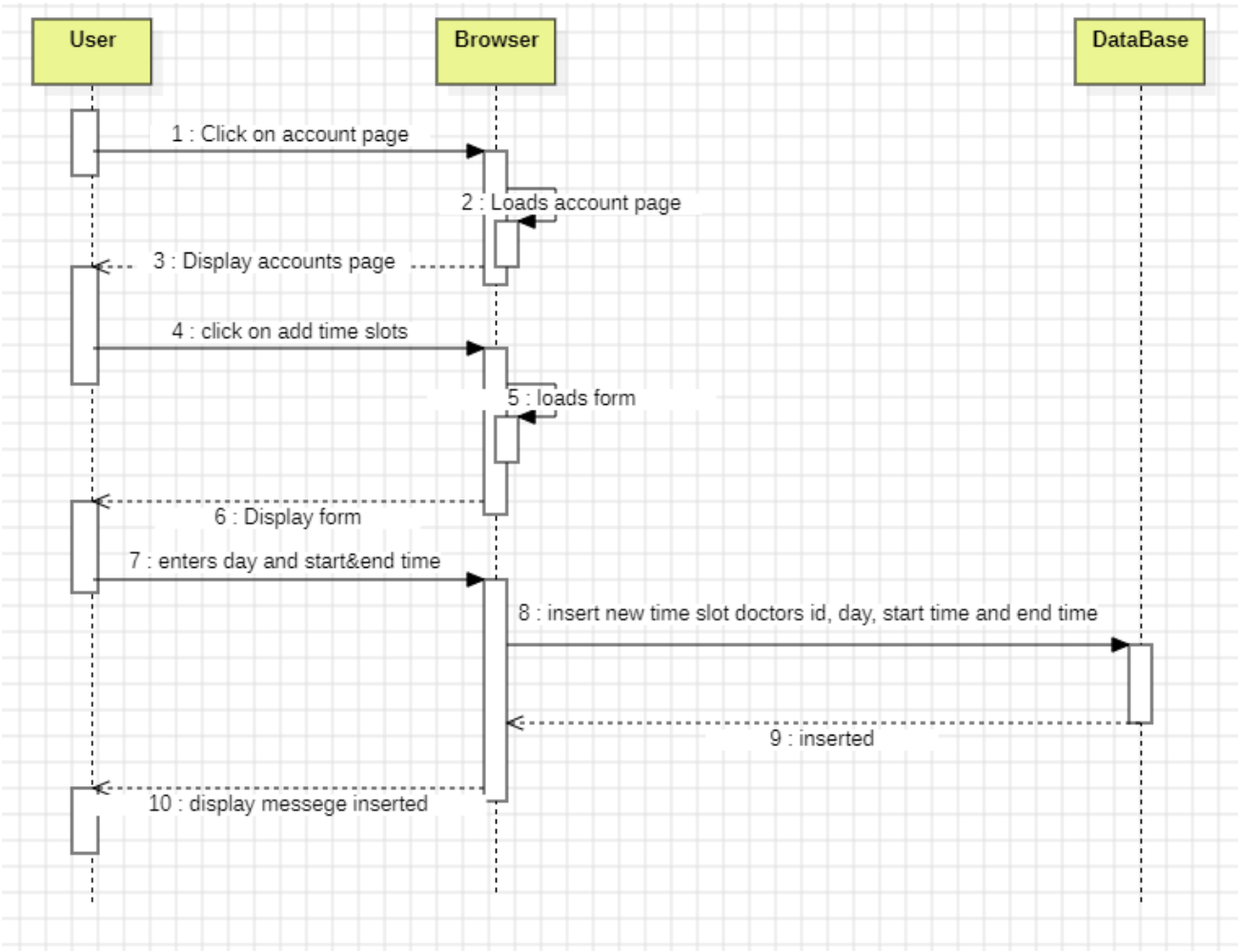


Figure 20 Sequence diagram on Cancel appointment

vi. Sequence diagram on Add available time



21 Sequence diagram on Add available time Figure