

Online Doctor Appointment Application

Final Year Project – Mid Report

Session 2020-2024



A project submitted in partial fulfilment of the
COMSATS University Degree
of
BS in Computer Science / Software Engineering (CUI)



Department of Computer Science
COMSATS University Islamabad, Lahore Campus

09 June 2023

Project Detail

Type (Nature of project)	[<input checked="" type="checkbox"/>] Development [<input type="checkbox"/>] Research [<input type="checkbox"/>] R&D			
Area of specialization	Web Development			
Project Group Members				
Sr.#	Reg. #	Student Name	Email ID	*Signature
(i)	SP20-BCS-110	Rameez Ashraf	Sp20-bcs-110@cuilahore.edu.pk	
(ii)	SP20-BCS-120	Dilawar Hussain	Sp20-bcs-120@cuilahore.edu.pk	

*The candidates confirm that the work submitted is their own and appropriate credit has been given where reference has been made to work of others

Plagiarism Free Certificate

This is to certify that, I am Rameez **Ashraf** S/D/o **Muhammad Ashraf** group leader of FYP under registration no **CIIT/SP20-BCS-110/LHR** at Computer Science Department, COMSATS University Islamabad, Lahore Campus. I declare that my FYP proposal is checked by my supervisor and the similarity index is **16%** that is less than 20%, an acceptable limit by HEC. Report is attached herewith as Appendix A.

Date: **09-jun 2023** Name of Group Leader: Rameez Ashraf Signature: _____

Name of Supervisor: **Abdul Karim Shahid**

Co-Supervisor (if any): Rizwan Qureshi

Designation: **Assistant professor**

Designation: **Assistant professor**

Signature: _____

Signature: _____

Abstract

Online doctor appointment application is a digital platform that enables patients to book appointments with medical professionals remotely. This application helps to streamline the appointment booking process, minimize wait times, and enhance access to medical services. Patients can use the application to select their preferred doctor, schedule an appointment, and receive reminders via notification. The application may also include features such as chat box for consultations, and medical record access. This abstract highlights the benefits of using an online doctor appointment application and the convenience it brings to patients seeking medical care.

This online application will overcome the issue of managing booking appointments. As the traditional method, this task is mostly hectic. By traditional method, a user may call the doctor or to his/her assistant to get the appointment or sometime user physically visit the clinic to get the appointment date, which is totally waste of time and resources.

Our Application is the solution of these problem described above. This application will provide the facility to patient or any other user an easy way to find require / specific doctor on the basis of doctor specialization, location, booking schedule and also on the base of doctor fee for poor user. This application will cover maximum basic functionalities to entertain the user basic requirements. We will keep its interface much simpler with the intention that every user feel easy and comfortable while using this application. The application also allows patients to reschedule or cancel appointments.

This Smart application is not only for the user/patient, it also beneficial for doctors and as well as Admin's of some medical institutes to manage their medical task more efficiently.

Contents

Abstract.....	3
1 Introduction	17
1.1 Goal and Objectives.....	18
1.2 Problem Statement	18
1.3 Assumptions and Constraint	19
1.3.1 Assumptions	19
1.3.2 Constraints.....	19
1.4 Project scope	19
2 Requirements Analysis.....	22
2.1 Literature Review	22
2.1.1 Research based related work:	22
2.1.1 Application based related work:.....	22
2.2 Stake Holders.....	25
2.2.1 Development team / Team members:	25
2.2.2 User:	25
2.2.3 Doctors:.....	25
2.2.4 Admin:.....	25
2.2.5 Project supervisor:	25
2.2.6 Academic Community:.....	25
2.2.7 Investors or sponsors:	25
2.2.8 Competitors:.....	25
2.3 Requirement Elicitation	26
2.3.1 Functional Requirements.....	26
2.3.2 Non-Functional Requirements	34
2.3.3 Requirements traceability matrix	38
2.4 Use case descriptions.....	40

2.5	Use Case Design	67
2.5.1	Patient and Doctor	67
2.5.2	Admin Panel:	68
2.6	Software Development Life Cycle Model:	69
3	System Design	71
3.1	Work Breakdown Structure (WBS)	71
3.2	Activity Diagrams	72
3.2.1	AD-01: User Sign Up	72
3.2.2	AD-02: Email Verification:.....	73
3.2.3	AD-03: User Login:	74
3.2.4	AD-04: Password reset:	75
3.2.5	AD-05: Doctor List:	76
3.2.6	AD-06: Apply for Doctor:.....	77
3.2.1	AD-07: Edit profile:.....	78
3.2.2	AD-08: View profile:	79
3.2.3	AD-09: Schedule Management:.....	80
3.2.4	AD-10: Searching / Filtering:.....	81
3.2.5	AD-11: Appointment Booking:	82
3.2.6	AD-12: Appointment Tracking:.....	83
3.2.7	AD-13: Appointment History:	84
3.2.8	AD-14: Cancel appointment:.....	85
3.2.9	AD-15: Update appointment:	86
3.2.10	AD-16: Notification System:.....	87
3.2.11	AD-17: Chat Box:	88
3.2.12	AD-18: View Candidate profile:.....	89
3.2.13	AD-19: Authority to Accept or reject request:	90
3.2.14	AD-20: Add feedback:.....	91
3.2.15	AD-21: View feedback:	92

3.2.16	AD-22: Add list of doctors:	93
3.2.17	AD-23: Delete list of doctors:	94
3.2.18	AD-24: Logout:	95
3.3	Sequence Diagrams	96
3.3.1	SD-01: User Sign Up	96
3.3.2	SD-02: Email Verification	97
3.3.3	SD-03: Sign In	98
3.3.4	SD-04: Password Reset	99
3.3.5	SD-05: View Doctor List.....	100
3.3.6	SD-06: Apply for Doctor	101
3.3.7	SD-07: Edit Profile	102
3.3.8	SD-08: View Profile.....	103
3.3.9	SD-09: Schedule Management.....	104
3.3.10	SD-10: Searching/Filtering.....	105
3.3.11	SD-11: Appointment Booking	106
3.3.12	SD-12: Appointment Track.....	107
3.3.13	SD-13: Clear Appointment History.....	108
3.3.14	SD-14: Cancel Appointment	109
3.3.15	SD-15: Update Appointment.....	110
3.3.16	SD-16: Notification System	111
3.3.17	SD-17: Chat Box	112
3.3.18	SD-18: View Candidate Profile	113
3.3.19	SD-19: Authority to accept or reject the request.....	114
3.3.20	SD-20: Add Feedback.....	115
3.3.21	SD-21: View Feedback	116
3.3.22	SD-22: Add List of Doctors	117
3.3.23	SD-23: Delete List of Doctors	118
3.3.24	SD-24: Logout.....	119

3.4	Software Architecture.....	120
3.4.1	Types of Client-Server Architecture:.....	120
3.5	Class Diagram.....	121
3.6	Database Diagram	122
3.7	Network Diagram(Gantt Chart)	123
3.8	Collaboration Diagram	124
3.8.1	Sign-Up:	124
3.8.2	Email Verification:	125
3.8.3	Sign In:	126
3.8.4	View Doctor List:	127
3.8.5	Add Feedback:	128
3.8.6	Delete Feedback:	129
3.8.7	View Feedback:	130
3.8.8	Appointment Booking:	131
3.8.9	Delete Appointment:	132
3.8.10	View Appointment:	133
3.8.11	View Profile:	134
3.8.12	Add doctor list:	135
3.8.13	Update doctor List:	136
3.8.14	Delete List of Doctor:	137
3.8.15	Logout:	138
4	System Testing	140
4.1	Test Cases	140
	TC01: Sign Up.....	140
	TC02: Email Verification	141
	TC03: Sign In	142
	TC04: Password Reset	143
	TC05: View Doctor List	144

TC 06: Apply for Doctor	145
TC 08: View Profile	146
TC 09: Set Schedule	147
TC 11: Appointment Booking	148
TC 12: Appointment Track	149
TC 13: Clear Appointment	150
TC 14: Cancel Appointment.....	151
TC 15: Update Appointment	152
TC 16: Notification System	153
TC 18: View Candidate Profile.....	154
TC 20: Add Feedback	155
TC 21: View Feedback.....	156
TC 22: Add List of Doctor.....	157
TC 23: Delete List of Doctors.....	158
TC 24: Logout	159
4.2 Unit Testing:	160
4.3 Integration Testing:	160
4.4 Acceptance Testing:	160
5 Conclusion	162
5.1 Problem faced and lesson learned	162
5.2 Summary of Project	162
5.3 Future Work	162
6 References.....	163
6.1 Online Doctor appointment By R Ashraf	164

List of Tables

Table 1: Related Work (Comparison Table)	24
Table 2: Sign Up.....	26
Table 3: Email Verification	26
Table 4: Log In	27
Table 5: Password Reset.....	27
Table 6: View Doctor	27
Table 7: Apply for Doctor	27
Table 8: Edit Profile	28
Table 9: View Profile	28
Table 10: Schedule Management	28
Table 11: Searching.....	28
Table 12: Appointment Booking	29
Table 13: Appointment Track	29
Table 14: Clear History	29
Table 15: Cancel Appointment.....	30
Table 16: Update Appointment	30
Table 17: Notification System.....	30
Table 18: Chat Box.....	31
Table 19: View Candidate Profile	31
Table 20: Authority to accept or reject request	31
Table 21: Add Feedback.....	32
Table 22: View Feedback.....	32
Table 23: Add list of Doctors	32
Table 24: Delete list of Doctor	33
Table 25: Logout	33
Table 26: NFR-01	34
Table 27: NFR-03.....	34
Table 28: NFR-03.....	34
Table 29: NFR-04.....	35
Table 30: NFR-05.....	35
Table 31: NFR-06.....	35
Table 32: NFR-07.....	36

Table 33: NFR-08.....	36
Table 34: NFR-09.....	36
Table 35: NFR-10.....	36
Table 36: NFR-11.....	37
Table 37: Traceability Matrix.....	39
Table 38: UC-01: Sign Up.....	40
Table 39:UC- 2: Email Verification	41
Table 40:UC-3: Sign In	42
Table 41: UC-4: Password Reset.....	43
Table 42: UC-05: View Doctor List.....	44
Table 43: UC-06: Apply for Doctor	45
Table 44: UC-07: Edit Profile	46
Table 45: UC-08: View Profile	47
Table 46: UC-09: Schedule Management	48
Table 47: UC-19: Searching	49
Table 48 UC-11: Appointment booking.....	50
Table 49 UC-12: Appointment Track.....	51
Table 50 UC-13: Clear Appointment History	52
Table 51 UC-14: Cancel Appointment.....	53
Table 52 UC-15: Update Appointment	55
Table 53 UC-16: Notification System.....	56
Table 54 UC-17: Chat box	57
Table 55 UC-18: View Candidate profile.....	58
Table 56: UC-19: Authority to accept or reject the request	59
Table 57: UC-20: Add Feedback.....	60
Table 58 UC-21: View Feedback	61
Table 59 UC-22: Add List of Doctors	62
Table 60: UC-23: Delete List of Doctors	64
Table 61: UC-24: Logout	66
Table 62: TC-01: Sign Up	140
Table 63: TC-02: Email Verification	141
Table 64: TC-03: Sign In.....	142
Table 65: TC-04: Password Reset	143
Table 66: TC-05: View Doctor List	144
Table 67: TC-06: Apply for Doctor.....	145

Table 68: TC-08: View Profile.....	146
Table 69: TC-09: Set Schedule.....	147
Table 70: TC-11: Appointment Booking	148
Table 71: TC-12: Appointment Track.....	149
Table 72: TC-13: Clear Appointment History.....	150
Table 73: TC-14: Cannel Appointment.....	151
Table 74: TC-15: Update Appointment.....	152
Table 75: TC-16: Notification	153
Table 76: TC-18: View Candidate Profile.....	154
Table 77: TC-20: Add Feedback	155
Table 78 TC-21: View Feedback	156
Table 79 TC-22: Add Doctor by Admin	157
Table 80 TC-23: Delete Doctor	158
Table 81 TC-24: Logout.....	159

List of Figures

Figure 1: use case diagram for user and doctor	67
Figure 2: use case diagram for Admin Portal	68
Figure 3: SDLC Model.....	69
Figure 4: Work Breakdown Structure (WBS)	71
Figure 5: AD- Sign Up	72
Figure 6: AD- Email Verification.....	73
Figure 7: AD- Sign In.....	74
Figure 8: Password Reset	75
Figure 9: AD- View Doctor List	76
Figure 10 AD- Apply for Doctor.....	77
Figure 11 AD-Edit Profile	78
Figure 12 AD- View Profile	79
Figure 13 AD- Schedule Management	80
Figure 14 AD- Searching.....	81
Figure 15 AD- Appointment Booking.....	82
Figure 16 AD- Appointment Track	83
Figure 17 AD- Appointment History	84
Figure 18 AD- Cancel Appointment	85
Figure 19 AD- Update Appointment.....	86
Figure 20 AD- Notification System	87
Figure 21 AD- Chat box	88
Figure 22 AD- View Candidate Profile.....	89
Figure 23 AD- Authority to accept or reject request	90
Figure 24 AD- Add Feedback	91
Figure 25 AD- View Feedback.....	92
Figure 26 AD- Add List of Doctors	93
Figure 27 AD- Delete List of Doctors.....	94
Figure 28 AD- Logout.....	95
Figure 29 SD- Sign Up	96
Figure 30 SD- Email Verification	97
Figure 31 SD- Sign In	98
Figure 32 SD- Password Reset.....	99
Figure 33 SD- View Doctor List	100

Figure 34 SD- Apply for Doctor	101
Figure 35 SD- Edit Profile.....	102
Figure 36 SD- View Profile.....	103
Figure 37 SD- Schedule Management.....	104
Figure 38 SD- Searching	105
Figure 39 SD- Appointment Booking	106
Figure 40 SD- Appointment Track.....	107
Figure 41 SD- Clear Appointment History	108
Figure 42 SD- Cancel Appointment.....	109
Figure 43 SD- Update Appointment.....	110
Figure 44 SD- Notification System	111
Figure 45 SD- Chat box.....	112
Figure 46 SD- View Candidate Profile	113
Figure 47 SD- Authority to accept or reject request.....	114
Figure 48 SD- Add Feedback	115
Figure 49 SD- View Feedback	116
Figure 50 SD- Add list of Doctors	117
Figure 51 SD- Delete list of doctors.....	118
Figure 52 SD- Logout.....	119
Figure 53 Class Diagram.....	121
Figure 54 Database Diagram.....	122
Figure 55 Network Diagram(Gantt Chart)	123
Figure 56 CD- Sign Up	124
Figure 57 CD- Email Verification.....	125
Figure 58 CD- Sign In.....	126
Figure 59 CD- View Doctor List.....	127
Figure 60 Add Feedback	128
Figure 61 Update Feedback.....	128
Figure 62 CD- Add Appointment.....	131
Figure 63 CD- Cancel Appointment.....	132
Figure 64 CD- View Appointment	133
Figure 65 CD- View Profile	134
Figure 66 CD- Add Doctor List.....	135
Figure 67 CD- Update List of Doctor.....	136
Figure 68 CD- Delete List of Doctor.....	137

Figure 69 CD- Logout	138
----------------------------	-----

CHAPTER 1

INTRODUCTION

1 Introduction

Indeed, Health is a very important factor of our life. As the population is increased day by day the health concern rate also increasing. Today people have more awareness about the health as compared to old days. Today, everyone wants to maintain his/her health by any means. Now a days, access to a qualify doctor is a big concern for everyone.

Let's take a look on traditional way, if someone is ill and wants to visit a doctor for check-up, he/she needs to visit the clinic/hospital and has to wait until the doctor is not free/available, or in some cases patient have to stand in the queued to get the token number for check-up. If there is long queue and the time of duty finished, in this case all the time goes to waste. And there are lot of issue in traditional appointment booking system.

To provide the solution of this problem we purposed the online appointment application system. As mentioned above it's not just overcome the problem in traditional system, it will also provide a lot of other facilities to the user/patient. Our application will provide an interactive platform to the user to user to get the appointment with the doctor of its own choice on the basis of Booking slot availability, there nearby location, Specialty of doctor. User also have the option to chat with the doctor in case he/she don't want the appointment but want to little concern only.

After booked the appointment user also have option to resume its appointment or cancel within time limit (Let say 48 hours earlier before the exact appointment time). User can also able to view its appointment history like past appointment, upcoming appointments, Successful appointment, canceled appointment.

Doctor can manage its appointment by using this application like confirm or decline the appointment request from the user. Doctor can see the history of the appointment, can keep record of the patient history. Doctor can edit its profile, Doctor can also manage its appointment schedule (Between what times to what time user can book the appointment).

Our application will provide different rights to the admin. Admin can prove or disprove the doctor hiring request. Add the doctor and delete/remove the doctor. Have list of all doctors, view status of doctor active or not.

All booking, appointments, requests (approved and disapproved) will be notify with the help of notification system. If user book the appointment, doctor will get the notification. User will get the notification on the response of doctor wither if confirmed or declined, Same as Admin will get the notification when someone apply for the doctor and in the response, doctor will get the response either its request approved or disapproved.

1.1 Goal and Objectives

The goal and objectives of this project is to provide the best solution of the problems discussed on above to our three main communities User, Doctor and Admin.

- To redeem your life from traditional hectic appointment mechanism.
- To book your appointment without any waste of time.
- Provide an easy/simple system for the use of normal peoples.
- Every kind of user can get the benefit from this application.
- Don't need to go physically for getting the appointment.
- To develop a user-friendly and an attractive online system for user/patient to search and find require doctor and book appointment.
- User can browse and book the appointment at any time.
- To provide the chat bot for the conversion or consultation.
- To develop the system to manage the schedule of the patient according the availability doctors for their requested day and time.
- To digitalize the patient record for long term use.
- Provide facility to user of cancelation the appointment.
- To provide reminder to the patient about upcoming appointments
- To provide a platform to the doctor where they can provide services to the patients.
- Doctor can create its well perform profile to get more appointments.
- To provide the authority to the admin to manage the records of doctors.
- Make admin able to add, remove any doctor.
- Give authority to admin for edit the details of the doctor.
- Make admin able to have the track of doctor activities.
- To help the user to earn by referring this application to his friends and family.

1.2 Problem Statement

The current process of scheduling doctor appointments is inefficient, leading to long waiting times, missed appointments, and challenges in maintaining the record of previous appointments. There is a need for an online doctor appointment system that streamlines scheduling, provides real-time availability information, offers appointment reminders, and improves appointment record management.

The proposed online doctor appointment system will allow patients to easily book appointments, view real-time availability, receive automated reminders, and provide healthcare providers with centralized appointment records. This system aims to enhance efficiency, and improve overall healthcare experience.

1.3 Assumptions and Constraint

1.3.1 Assumptions

- i. Availability of Internet Connection: The assumption is that users, both patients and doctors, have access to a stable internet connection to interact with the web application effectively.
- ii. Device Compatibility: It is assumed that the web application will be compatible with a wide range of devices, including desktop computers, laptops, tablets, and smartphones, running various operating systems and web browsers.
- iii. Maximum Doctor Participation: The success of the platform relies on a number of sufficient and well professional doctors actively participating and registering on the system to offer their services to patients.
- iv. Timely Data Updates: System will remain in active state any time to update the data timely and book the appointments without any conflicts.
- v. User Data Accuracy: The assumption is that users will provide accurate and up-to-date personal information, medical history to get the better treatment.

1.3.2 Constraints

- i. The user cannot get video consultation.
- ii. User cannot give feedback without booking any appointment.
- iii. User cannot book appointment without log in.
- iv. Application will not work without internet connection.
- v. User can only book the appointment on available slots.

1.4 Project scope

The scope of our project is Development of the complete and working application using MERN Stack Technologies to manage the appointments and records online. The aim is to enhance the overall health care experience for both doctor and admin. Project improve the accessibility to healthcare services.

The proper scope is defined below

- Its online platform that is for health care only
- Provide an efficient way to manage medical related daily task
- Effective for doctor, patient and admin as well
- User can find the nearest doctor, Access through location
- User can find the doctor with respect to the speciality of the doctor
- User can book the appointment only in available slots
- User can cancel the appointment but before 24 hours of exact booking time
- Booking schedule management

- Doctor can accept or decline the booking request
- User, doctor can communicate via chat box
- User can refer this application to his family and friends
- Admin will receive the doctor application requests
- Admin can contact with doctor for the interview (Optional)
- Admin can approve or disapprove the doctor applicant request
- Admin have no authority to edit the user/patient credential
- Admin can only edit the credential of the doctor
- Admin can approve or disapprove the referral bonus.

CHAPTER 2

Requirements Analysis

2 Requirements Analysis

2.1 Literature Review

There are many web applications which provides almost the same functionalities as over platform will offer. Below are the descriptive details of related apps working on the same functionalities.

2.1.1 Research based related work:

There is no research based related work as the system is merely a development project.

2.1.1 Application based related work:

Following are some applications based related work that are much alike our system.

2.1.1.1 Practo

Practo ^[2] by [Shashank ND](#) is an online platform that provides patients with access to doctors, clinics, and hospitals. The platform offers a range of features, including doctor profiles, online booking, and patient reviews. Not available in Pakistan.

2.1.1.2 Healthgrades

Healthgrades ^[3] by [Red Ventures](#) is an online platform that provides patients with information on doctors, hospitals, and healthcare providers. The platform offers a range of features, including doctor profiles, patient reviews, and online booking. Healthgrades also offers a telemedicine service that allows patients to consult with doctors remotely. Not available in Pakistan.

2.1.1.3 Marham

Marham ^[4] by [Ehsan Imam](#) is an online healthcare platform that allows users to book appointment with doctors, search for doctors by specialty, locations and other filters, read doctors reviews by other patients, read health related articles and blogs written by doctors and healthcare professionals and can receive health-related notifications and reminders.

2.1.1.4 MeriSehat

MeriSehat ^[5] by [Babar Rashid Khan](#) is an online healthcare platform based in Pakistan that provides users with access to doctors and healthcare providers. Users can book appointments with doctors and receive online consultations through the platform.

2.1.1.5 Oladoc

Oladoc ^[6] by [Abid Zubari](#) is an online healthcare platform that allows users in Pakistan to book appointments with doctors and healthcare providers. The platform provides users with access to a wide range of healthcare services, including general practitioners, specialists, and diagnostic services.

While these existing applications offer similar functionalities to our proposed web application, we believe that there is still room for improvement. We will design our website as simple as possible so that everyone can book their appointment without any difficulty. In our application we will allow existing users to earn money by referring this application to their family and friends. When a user refers this application to a referee, and referee sign up and book his first appointment through referral link, then the referrer will get

certain amount of bonus on completion of the appointment. This not only helpful for users but also for us, because this method can increase user acquisition, produce better quality users, higher retention rates and also a cost-effective way of marketing.

Comparison Table:

Features	Our Application (Smart Health System)	Oladoc	Marham	MeriSehat	Healthgrades	Practo
Searching for Doctors	Yes	Yes	Yes	Yes	Yes	Yes
Doctors's Details	Yes	Yes	Yes	Yes	Yes	Yes
Reviews	Yes	Yes	Yes	Yes	Yes	Yes
Online Appointment's Booking	Yes	Yes	Yes	Yes	Yes	Yes
Appointment Reminders	Yes	Yes	Yes	Yes	Yes	Yes
Simplicity	Yes	Yes	Yes	Yes	No	No
Appointment History	Yes	Yes	Yes	No	Yes	Yes
Appointment's charges	Low	High	High	High	Very high	Very high
Refund	Yes	Yes	Yes	No	No	No
Cancellation fees	Low or none	High	High	High	High	High
Referral Program	Yes	No	No	No	No	No
Availability (Pakistan)	Yes	Yes	Yes	Yes	No	No

Table 1: Related Work (Comparison Table)

2.2 Stake Holders

2.2.1 Development team / Team members:

Each of the individual member working on this project is stake holder. They will provide their talents and knowledge to the project and have a stake in its success because it is a component of their academic work.

2.2.2 User:

Every End user who will use this online application are our end users. The popularity or failure of the product among the user have direct impact to our project. End user are our direct stake holders.

2.2.3 Doctors:

Doctor are our direct stake holders. They have direct effect with failure or success of the project. If application does not perform the functionalities properly it and book the appointment with conflicts will became cause of problem for both patients and doctors.

2.2.4 Admin:

Admin will also cause the problem if application does not provide the basics functionalities.

2.2.5 Project supervisor:

The Project supervisor, who supervise this project also have some kind of link with the success and failure of the project.

2.2.6 Academic Community:

The academic community might also be regarded as stakeholders because the initiative is an academic endeavour. They could be curious about the project's conclusions, the methods employed, and the research design.

2.2.7 Investors or sponsors:

If your project needs finance, investors or sponsors would be regarded as stakeholders since they will put money into it and hope to get something back.

2.2.8 Competitors:

If our project the huge popularity among the people this might affect the business of competitors.

2.3 Requirement Elicitation

2.3.1 Functional Requirements

FR01: Sign Up:

Req. No.	Functional Requirement
FR01-01	System will allow the user/patient and Doctor to sign up and get registered using his/her name, Gmail, email and password.
FR01-02	A system must allow the user and doctor to sign Up with Google.
FR01-03	The system must display the message to the user that he/she has been registered.

Table 2: Sign Up

FR 02: Email verification:

Req. No.	Functional Requirement
FR02-01	Application must send a verification email to the user only once time on Sign Up.
FR02-02	Not register the user without verification.
FR02-03	User can be log In after successful verification.

Table 3: Email Verification

FR 03: Log In:

Req. No.	Functional Requirement
FR03-01	Application must allow the user and doctor to login by password and username.
FR03-02	Application must allow the user to log in on more than one devices(Multiple systems).
FR03-03	By clicking on the login button system must log In the user.
FR03-04	Application must provide the option of forget password.

*Table 4: Log In***FR 04: Password Reset:**

Req. No.	Functional Requirement
FR04-01	Application allow the user to reset the password.

*Table 5: Password Reset***FR 05: View Doctor List:**

Req. No.	Functional Requirement
FR05-01	After Login application show the list of the doctors(Almost all).

*Table 6: View Doctor***FR 06: Apply for Doctor:**

Req. No.	Functional Requirement
FR06-01	A normal user can apply for doctor.
FR06-02	User can get the status of doctor after approval from the admin.

Table 7: Apply for Doctor

FR 07: Edit Profile:

Req. No.	Functional Requirement
FR07-01	Doctor can create its profile.
FR07-02	Application will not allow the Doctor to delete or edit the profile of other doctors.

*Table 8: Edit Profile***FR 08: View Profile:**

Req. No.	Functional Requirement
FR08-01	After Login, user can view his/her profile.
FR08-02	User can also view the profile of the doctor by click on “Profile View”.

*Table 9: View Profile***FR 09: Schedule Management:**

Req. No.	Functional Requirement
FR09-01	Doctor can update or edit its schedule (Work timing).
FR09-02	User can view the schedule of the doctor.

*Table 10: Schedule Management***FR 10: Searching/Filtering:**

Req. No.	Functional Requirement
FR10-01	User can search the doctor on the basis of different filters like Location, specially etc.

Table 11: Searching

FR 11: Appointment booking:

Req. No.	Functional Requirement
FR11-01	User / patient can book the appointment according to the schedule of the require doctor.
FR11-02	Application not allow to book the appointment if doctor is not free in that particular time.

*Table 12: Appointment Booking***FR 12: Appointment Track (Past and Upcoming):**

Req. No.	Functional Requirement
FR12-01	User and doctor can view the upcoming appointments.
FR12-02	User and doctor can also view history appointment.

*Table 13: Appointment Track***FR 13: Clear Appointment History:**

Req. No.	Functional Requirement
FR13-01	Application must allow its users to clear the appointment history.
FR13-02	Only allow to delete past appointments, not upcoming.

Table 14: Clear History

FR 14: Cancel Appointment:

Req. No.	Functional Requirement
FR14-01	Application must allow the user to cancel the upcoming appointments.
FR14-02	Patient can only cancel the appointment before 48 hours.
FR14-03	Same for doctor can also cancel the appointment.

*Table 15: Cancel Appointment***FR 15: Update Appointment:**

Req. No.	Functional Requirement
FR15-01	The application must provide an interface to update the appointment.
FR15-02	The application must send the notification to the doctor for approval.
FR15-03	If updated appointment approved, the application should show the updated appointment.

*Table 16: Update Appointment***FR 16: Notification System:**

Req. No.	Functional Requirement
FR16-01	Application must push the notification when certain activity performed. Like
FR16-02	When user request for book the appointment, doctor get the notification.
FR16-03	User get the notification on the response of doctor(Either accept or decline).
FR16-04	Admin get the notification when a user apply for the doctor.

Table 17: Notification System

FR 17: Chat Box:

Req. No.	Functional Requirement
FR17-01	The application will allow the patient and Doctor to communicate using this platform.
FR17-02	A patient can ask the question from doctor using this application without using any personal account.
FR17-03	It also helpful if a doctor does not share its contact.

*Table 18: Chat Box***FR 18: View candidate profiles:**

Req. No.	Functional Requirement
FR18-01	Application must allow Admin to view the profile of the candidates who apply for the doctor on our platform.

*Table 19: View Candidate Profile***FR 19: Authority to accept or reject the request:**

Req. No.	Functional Requirement
FR19-01	Application must allow the doctor to book the appointment request or decline the request.
FR19-02	Admin can approve or disprove the received requests.
FR19-03	User get the notification on the response of doctor (Either accept or decline).
FR19-04	Admin get the notification when a user applies for the doctor.

Table 20: Authority to accept or reject request

FR 20: Add Feedback:

Req. No.	Functional Requirement
FR20-01	The application will provide the interface to patient to add the feedback.
FR20-02	The application will allow the student to add feedback by commenting or by rating through stars.

*Table 21: Add Feedback***FR 21: View Feedback:**

Req. No.	Functional Requirement
FR21-01	The application will allow the patient and Admin to view the doctor feedback.
FR21-02	The application will allow patient to view feedback provided by his/herself and by other patients about the doctors.
FR21-03	Feedback will be helpful for the patient about choosing the doctor.

*Table 22: View Feedback***FR 22: Add list of Doctors:**

Req. No.	Functional Requirements
FR022-01	The system will allow Admin to add a doctor.
FR022-02	The system will ensure that the added doctor is not already present in the list.
FR022-03	The system will allow admin to view the list before adding a new doctor.

Table 23: Add list of Doctors

FR 23: Delete list of Doctors:

Req. No.	Functional Requirements
FR023-01	The system will enable admin to delete the doctor account from the list of teachers.
FR023-02	The system will display a confirmation message before deleting the specific doctor.
FR023-03	The system will display a message after deleted the doctor.

*Table 24: Delete list of Doctor***FR 24: Logout:**

Req. No.	Functional Requirement
FR24-01	The application must provide an option on the interface so that the student, teacher and can logout after sign in.
FR24-02	The application must allow the student, teacher and admin to log out by clicking Logout Button
FR24-03	The application will display a message about successfully log out to all.

Table 25: Logout

2.3.2 Non-Functional Requirements

NFR 01: Performance:

Req. No.	Non-Functional Requirement
NFR01-01	The application should be able to handle a large number of users and appointments without significant delays or downtime.
NFR01-02	The system must notify the student and teacher about his/her appointment before time.
NFR01-03	The system shall give a quick response about finding the required doctor.

Table 26: NFR-01

NFR 02: Maintainability:

Req. No.	Non-Functional Requirement
NFR02-01	The Application should be maintainable and easy to modify the changes.
NFR02-02	The Application should perform without any major failure.

Table 27: NFR-03

NFR 03: Reliability:

Req. No.	Non-Functional Requirement
NFR03-01	The system should be reliable and available to users at all times.
NFR03-02	The design of the system must be easy to use.
NFR03-03	The failure rate of the system should be less.
NFR03-04	The system shall provide some action control in the user's hand.
NFR03-05	The load time of images and page should be less as possible.

Table 28: NFR-03

NFR 04: Availability:

Req. No.	Non-Functional Requirement
NFR04-01	The application should be available 24/7.
NFR04-02	Application should be available on most of smart devices like smartphone, tablet, Pc Laptop etc.

*Table 29: NFR-04***NFR 05: Accessibility:**

Req. No.	Non-Functional Requirement
NFR05-01	The application will be accessible to all across the Pakistan areas who have internet access.
NFR05-02	Application will be accessible to almost all people educated, not well educated.

*Table 30: NFR-05***NFR 06: Usability:**

Req. No.	Non-Functional Requirement
NFR06-01	System should be easy to use and easy to learn and remember.
NFR06-02	User should be able to perform the desired action right away.
NFR06-02	Application should be available on most of smart devices like smartphone, tablet, Pc Laptop etc.

Table 31: NFR-06

NFR 07: portability:

Req. No.	Non-Functional Requirement
NFR07-01	The website can be used on any service engine like Chrome, Mozilla Firefox, etc.
NFR07-02	The Responsive design of the website allows user to use the website on tablets mobile and many other portable devices.

*Table 32: NFR-07***NFR 08: Scalability:**

Req. No.	Non-Functional Requirement
NFR08-01	The system should be scalable and able to handle increasing numbers of users.

*Table 33: NFR-08***NFR 09: Referral System:**

Req. No.	Non-Functional Requirement
NFR09-01	The system should be scalable and able to handle increasing numbers of users.

*Table 34: NFR-09***NFR 10: Security:**

Req. No.	Non-Functional Requirement
NFR10-01	Security and Privacy will be maintained.
NFR10-02	Every user's data should be secured, and every user should be able to login using their own credentials
NFR10-03	A user should not be able to see any other user's data

Table 35: NFR-10

NFR 11: Browser Compatibility:

Req. No.	Non-Functional Requirement
NFR11-01	The application should be compatible with different web browsers, including well-known ones like Google Chrome, Mozilla Firefox, and Safari

Table 36: NFR-11

2.3.3 Requirements traceability matrix

Requirement's Traceability Matrix					
Project Name	Online Doctor Appointment Application				
Project Supervisor	Sir Abdul Karim Shahid				
Project Description	Online Doctor Appointment system is Web based application designed to overcome the problem of traditional appointment system. Application will provide series of facilities to user to book the appointment easily on the bases of own preferences.				
Requirement No.	Requirement Description	Use Cases	Test Cases	Activity Diagram	Sequence Diagram
FR01	The Admin and user should be able to login through required credentials.	UC-01	TC-01	AD-01	SD-01
FR02	The user will receive email verification on Signup.	UC-02	TC-02	AD-02	SD-02
FR03	User Login.	UC-03	TC-03	AD-03	SD-03
FR04	Password Reset.	UC-04	TC-04	AD-04	SD-4
FR05	View Doctor List	UC-05	TC-05	AD-05	SD-05
FR06	Apply for Doctor.	UC-06	TC-06	AD-06	SD-06
FR07	Edit Profile	UC-07	TC-07	AD-07	SD-07
FR08	View Profile.	UC-8		AD-13	SD-07
FR09	Schedule Management	UC-09	TC-09	AD-09	SD-15
FR10	Searching	UC-10		AD-10	SD-10
FR11	Appointment Booking	UC-11	TC-11	AD-11	SD-11

FR12	Appointment Track	UC-12	TC-12	Ad-12	SD-12
FR13	Clear Appointment History.	UC-13	TC-13	AD-13	SD-13
FR14	Cancel Appointment.	UC-14	TC-14	AD-14	SD-14
FR15	Update Appointment.	UC-15	TC-15	AD-15	SD-15
FR16	Notification System.	UC-16	TC-16	AD-16	SD-16
FR17	Chat Box.	UC-17		AD-17	SD-17
FR18	View Candidate Profile.	UC-18	TC-18	AD-18	SD-18
FR19	Authority to accept or reject the request.	UC-19		AD-19	SD-19
FR20	Add Feedback	UC-20	TC-20	AD-20	SD-20
FR21	View Feedback.	UC-21	TC-21	AD-21	SD-21
FR22	Add List of doctors.	UC-22	TC-22	AD-22	SD-22
FR23	Delete list of Doctors.	UC-23	TC-23	AD-23	SD-23
FR24	Logout.	UC-24	TC-24	AD-24	SD-24

Table 37: Traceability Matrix

2.4 Use case descriptions

UC-01: Sign Up

Use case name:	Sign Up
Use case id:	1
Actors:	Patient and Doctor
Use Case Summary	User visits the website open Signup page, fill the signup form by providing his/her credentials and get registered. The user can also Signup using Google.
Pre-conditions:	user shall not be registered
Normal Course of Event (Basic Flow):	
Steps#	Description
1.	User visit the web site
2.	User fill the sign-up form using credentials Name, Contact Number, Email, Address and Password
3.	User click on sign up button to get registered
4.	User click on ok button and submit the form
4.	The system will send verification Email.
Alternate Scenario:	
a. If the user will not provide at least one information the system will generate an error	
b. If the user will not provide valid email the system will generate an error message	
c. If the user is already registered the system will generate an error	
d. If the user will add the email address which is already exist the system will generate an error	
Post-conditions:	User is registered.
Exceptions:	Nil
UC cross reference:	FR-01

Table 38: UC-01: Sign Up

UC-02: Email Verification

Use case name:	Email Verification
Use case id:	1
Actors:	Patient and Doctor
Use Case Summary	User visits the website open Signup page, fill the signup form by providing his/her credentials and get registered. The user can also Signup using Google.
Pre-conditions:	user shall not be registered
Normal Course of Event(Basic Flow):	
Steps#	Description
1.	User visit the web site
2.	User fill the sign up form using credentials Name, Contact Number, Email, Address and Password
3.	User click on sign up button to get registered
4.	User click on ok button and submit the form
4.	The system will send verification Email.
Alternate Scenario:	
a. If the user will not provide at least one information the system will generate an error	
b. If the user will not provide valid email the system will generate an error message	
c. If the user is already registered the system will generate an error	
d. If the user will add the email address which is already exist the system will generate an error	
Post-conditions:	User is registered.
Exceptions:	Nil
UC cross reference:	FR-01

Table 39:UC- 2: Email Verification

UC-03: Sign In

Use case name:	Sign In
Use case id:	1
Actors:	Patient, Doctor and Admin
Use Case Summary	Users visit the site and open login form and fill it by entering password and Email. User then click on the login button. System will check the credential in the database and login it if matched.
Pre-conditions:	user must be registered
Normal Course of Event(Basic Flow):	
Steps#	Description
1.	User visit the web site and click on login button
2.	User will enter the Email and password for login
3.	User then click on the login button to send the login request
4.	System will verify and validate the entered information is either correct or not. If given information are correct then user will be shown the successfully logged in message
Alternate Scenario:	
a. If the user will not provide at least one information the system will generate an error	
b. If the user will not provide valid email the system will generate an error message	
c. If the user is already registered the system will generate an error	
d. If the user will add the email address which is already exist the system will generate an error	
Post-conditions:	User is registered.
Exceptions:	Nil
UC cross reference:	FR-03

Table 40:UC-3: Sign In

UC-04: Password Reset

Use case name:	Password Reset
Use case id:	4
Actors:	User: The individual who needs to reset the password
Use Case Summary	Password reset is a vital functionality for user authentication purposes. Its allow to set new password
Pre-conditions:	The user must have an existing account The user must have the access to the email associated with the account.
Normal Course of Event (Basic Flow):	
Steps#	Description
1.	User visit the login page of the application and click on the 'Forget password' option.
2.	Application redirect the user to password reset page
3.	User enter his/her registered email and submit the form
4.	The user gets the Email that have link of password reset
5.	User can set the new password
6.	Now user can login with new password
Alternate Scenario:	
a. If the user enter the wrong email, application gives the error 'email not registered'.	
Post-conditions:	The user password is reset successfully. User can login with new password.
Exceptions:	If user also forget the email address then he/she make new account with new Email address.
UC cross reference:	FR-04

Table 41: UC-4: Password Reset

UC-05: View Doctor List

Use case name:	View Doctor List
Use case id:	5
Actors:	Patient and Admin
Use Case Summary	This use case helps the user to view the teacher.
Pre-conditions:	Number of doctors should be registered.
Normal Course of Event(Basic Flow):	
Steps#	Description
1.	User visit the login page of the application and click on the 'Forget password' option.
2.	Application redirect the user to password reset page
3.	User enter his/her registered email and submit the form
4.	The user get the Email that have link of password reset
5.	User can set the new password
6.	Now user can login with new password
Alternate Scenario:	
a. If the user enters the wrong email, application gives the error 'email not registered'.	
Post-conditions:	The user password is reset successfully. User can login with new password.
Exceptions:	If user also forget the email address then he/she make new account with new Email address.
UC cross reference:	FR-04

Table 42: UC-05: View Doctor List

UC-06: Apply for doctor

Use case name:	Apply for Doctor
Use case id:	6
Actors:	User
Use Case Summary	Give facility to user to apply for the doctor Account.
Pre-conditions:	User have Medical degree or license.
Normal Course of Event(Basic Flow):	
Steps#	Description
1.	User log in the application as normal user.
2.	Then from the module 'Apply for Doctor' user can fill the form for doctor account.
3.	The request will be send to the Admin.
4.	Admin will prove or decline the request.
5.	User can registered as doctor(role-doctor) if request approved.
6.	Now user can use the application as doctor and can set his/her schedule.
Alternate Scenario:	
a. Doctor Application form will have some compulsory field, like license number, Medical degree Certificate. b. If there is any compulsory field is missing then user can not apply for doctor. c. If there is some technical issue, user can contact with supporting team.	
Post-conditions:	User role changed to doctor as user recive the confirmation notification from the Admin.
Exceptions:	Admin can add the reason about rejection the application.
UC cross reference:	FR-06

Table 43: UC-06: Apply for Doctor

UC-07: Edit Profile

Use case name:	Edit Profile
Use case id:	7
Actors:	Doctor
Use Case Summary	Option to Edit the profile.
Pre-conditions:	a) The doctor must be registered and have an active account in the system. b) The doctor must be logged into the system.
Normal Course of Event(Basic Flow):	
Steps#	Description
1.	The doctor logs into the system using their credentials.
2.	The system presents the doctor with their profile page or a designated section for profile management.
3.	The doctor selects the "Edit Profile" option.
4.	The system displays the doctor's existing profile information in editable fields.
5.	The doctor modifies the desired profile fields.
Alternate Scenario:	
If the doctor wants to change their password: <ol style="list-style-type: none"> The system prompts the doctor to enter their current password. The doctor enters their current password. The doctor enters the new password and confirms it. The doctor saves the changes, and the system updates their password accordingly. 	
Post-conditions:	a) The doctor's profile information is updated with the edited details.
Exceptions:	Admin can add the reason about rejection the application.
UC cross reference:	FR-07

Table 44: UC-07: Edit Profile

UC-08: View Profile

Use case name:	Edit Profile
Use case id:	8
Actors:	Student, teacher and admin
Use Case Summary	User visits the website, open profile page and view his/her profile.
Pre-conditions:	User shall be registered
Normal Course of Event(Basic Flow)	
Steps#	Description
1.	User visit the web site.
2.	User click on profile button.
3.	User open his/her profile, view his/her profile or the other users.
Alternate Scenario	
a) If the user is not registered and want to view profile of other user the system will generate an error	
Post-conditions:	Profile is viewed.
Exceptions:	Nil
UC cross reference:	FR-08

Table 45: UC-08: View Profile

UC-09: Schedule Management

Use case name:	Schedule Management
Use case id:	9
Actors:	Doctor
Use Case Summary	Doctor can manage his/her schedule for the appointments.
Pre-conditions:	Doctor should have the active account. Doctor should be log In.
Normal Course of Event(Basic Flow)	
Steps#	Description
1.	Doctor Log In the account
2.	User Click on the Profile
3.	Here Doctor has field of Schedule.
4.	Doctor will set the schedule here.
Alternate Scenario	
a) If there is off by Admin in Some day . b) Then Doctor cannot manage the schedule for that day.	
Post-conditions:	Schedule will be set.
Exceptions:	Doctor may disable the booking for a day
UC cross reference:	FR-09

Table 46: UC-09: Schedule Management

UC-10: Search/Filter

Use case name:	Search/Filter
Use case id:	10
Actors:	User and Admin
Use Case Summary	User and Admin can search the specific or desire doctor by applying the filter. Filter can be based on location, Doctor speciality, fee etc.
Pre-conditions:	User and Admin Should Have active account. There should be list of doctors in the application.
Normal Course of Event(Basic Flow)	
Steps#	Description
1.	User or Admin will log In the account.
2.	Click on the doctor List
3.	Write the description in search field
4.	List of all doctor with respect to search will shows.
Alternate Scenario	
a) If No doctor list match with the search keywords then it show nothing(Empty Screen) b) Or can show the message 'No match found'.	
Post-conditions:	Doctor will Show according to search. In case of no Match message will show.
Exceptions:	Application is just new and don't have doctor list.
UC cross reference:	FR-10

Table 47: UC-19: Searching

UC-11: Appointment Booking

Use case name:	Appointment Booking
Use case id:	11
Actors:	Patient
Use Case Summary	“Patient login to the system. He/she can add appointments from the portal/dashboard.”
Pre-conditions:	Student must login.
Normal Course of Event(Basic Flow)	
Steps#	Description
1.	Patient visit the web site and login to his/her account.
2.	He/she selects the appointment.
3.	He/she clicks submit button.
Alternate Scenario	
a) If selected appointment slot is already booked then system not allow the booking to the patient. b) System may say select other slots.	
Post-conditions:	Appointment will be added on both doctor and patient side.
Exceptions:	Nil
UC cross reference:	FR-11

Table 48 UC-11: Appointment booking

UC-12: Appointment Track

Use case name:	Appointment Track(Past and Upcoming)
Use case id:	12
Actors:	Patient and Doctor
Use Case Summary	To view your appointment list past and upcoming.
Pre-conditions:	Patient and Doctor Should be have attest one appointment.
Normal Course of Event(Basic Flow)	
Steps#	Description
1.	Login to the account.
2.	Go to the appointment section
3.	View your appointments
Alternate Scenario	
a) If there is no appointment booked yet system show message “No appointment booked yet”.	
Post-conditions:	User will know the appointment history.
Exceptions:	Nil
UC cross reference:	FR-12

Table 49 UC-12: Appointment Track

UC-13: Clear Appointment History

Use case name:	: Clear Appointment History
Use case id:	13
Actors:	Patient
Use Case Summary	From the Appointment track section patient can clear the appointment history.
Pre-conditions:	There should be one past appointment.
Normal Course of Event(Basic Flow)	
Steps#	Description
1.	Patient visit the web site and login to his/her account.
2.	He/she will click on the appointment section.
3.	He/she will see the appointment
4.	There will be option to clear the history.
5.	Patient can clear the history here.
Alternate Scenario	
a) If there is no past appointment user then there will be nothing to clear or remove.	
Post-conditions:	Past appointments will be removed.
Exceptions:	Patient may not have any appointment history
UC cross reference:	FR-13

Table 50 UC-13: Clear Appointment History

UC-14: Cancel Appointment

Use case name:	Cancel Appointment
Use case id:	14
Actors:	Patient and Doctor
Use Case Summary	Patient and Doctor can cancel the upcoming appointment
Pre-conditions:	Cancel the appointment before 48 hours of appointment time.
Normal Course of Event (Basic Flow)	
Steps#	Description
1.	Patient or Doctor visit the web site and login to his/her account.
2.	He/she will click on the appointment section.
3.	He/she will see the appointment
4.	User can cancel his/her upcoming appointment.
Alternate Scenario	
a) If time of upcoming appointment is less than 48 hours, system will not allow to cancel the appointment.	
Post-conditions:	Appointment should be cancelled. That specific slot will become free. Patient or Doctor may get the notification on cancel the appointment.
Exceptions:	If user have no appointment booked yet, then he cannot perform this activity.
UC cross reference:	FR-14

Table 51 UC-14: Cancel Appointment

UC-15: Update Appointment

Use case name:	Update Appointment
Use case id:	15
Actors:	Student, teacher
Use Case Summary	User visits the website, open make an appointment page to update his/her appointment
Pre-conditions:	User had already made an appointment
Normal Course of Event(Normal Flow)	
Steps#	Description
1.	User visit the web site
2.	User click on make an appointment button
3.	User open his/her appointment
4.	User selects the option to update his/her appointment
5.	User update his appointment
6.	The system will display a confirmation message to the user to update his/her appointment
7.	User click on the ok button and then save his/her appointment
Alternate Scenario	
a. If the user will not fill at least one section the system will generate an error	
b. If the user will mistakenly add wrong information the system will generate an error	
c. If the user will add the service which is not available the system will generate an error	
Post-conditions:	Appointment is updated

Exception:	Nil
UC cross reference:	FR-15

Table 52 UC-15: Update Appointment

UC-16: Notification System

Use case name:	Notification System
Use case id:	16
Actors:	Student, teacher
Use Case Summary	“User can receive notifications of any type example, cancelation appointment and confirmation appointment.
Pre-conditions:	User must login.
Normal Course of Event(Normal Flow)	
Steps#	Description
1.	User visit the web site and login his/her account.
2.	When a notification arrives, it will be added in notification queue.
3.	User clicks notification button.
4.	After clicking the updates/notification button all latest notifications will be shown.
Alternate Scenario:	Nil
Post-conditions:	User will get all Notification. There will be no new(Unseen) notifications.
Exception:	Nil
UC cross reference:	FR-16

Table 53 UC-16: Notification System

UC-17: Chat Box

Use case name:	Chat Box Interaction
Use case id:	17
Actors:	User(Patient, Admin, Doctor)
Use Case Summary	The goal of this use case is to enable users to interact with a doctor to do some queries and same for the doctors.
Pre-conditions:	The chat box system is accessible and operational.
Normal Course of Event (Normal Flow)	
Steps#	Description
1.	User login with his/her credentials.
2.	Click on the chat box and select the doctor name to start the conversation.
Alternate Scenario:	Nil
Post-conditions:	Receiver will receive the messages.
Exception:	receiver restrict the account with 'No one can message me'.
UC cross reference:	FR-17

Table 54 UC-17: Chat box

UC-18: View candidate profile

Use case name:	View candidate profile
Use case id:	18
Actors:	Admin
Use Case Summary	Admin can view the candidate profile (Who apply for doctor profile).
Pre-conditions:	User Submit the doctor Application form successfully.
Normal Course of Event(Normal Flow)	
Steps#	Description
1.	As user submit the doctor application the Admin will get the notification.
2.	To response on the received request admin can view the candidate profile.
Alternate Scenario: Nil	
Post-conditions:	Nil.
Exception:	Admin may ignore the notification and don't view the candidate profile.
UC cross reference:	FR-18

Table 55 UC-18: View Candidate profile

UC-19: Authority to accept or reject the request

Use case name:	Authority to accept or reject the request
Use case id:	19
Actors:	Admin
Use Case Summary	Admin can accept or decline the candidate request.
Pre-conditions:	Admin must receive the request first.
Normal Course of Event(Normal Flow)	
Steps#	Description
1.	As user submit the doctor application the Admin will get the notification.
2.	Admin can review the candidate profile, medical qualification etc.
3.	After review the profile Admin can prove or decline the candidate request.
Alternate Scenario: Nil	
Post-conditions:	Candidate get the notification either accept or reject the application. In case of accept, user role will changed to doctor.
Exception:	Admin may ignore the notification and don't view the candidate profile(Pending request).
UC cross reference:	FR-19

Table 56: UC-19: Authority to accept or reject the request

UC-20: Add Feedback

Use case name:	Add Feedback
Use case id:	20
Actors:	Patient
Use Case Summary	Patient visits the website, open feedback page and add feedback about this teachers through commenting or by rating through stars.
Pre-conditions:	Patient must be registered
Normal Course of Event:	
Steps#	Description
1.	Patient visit the web site
2.	Patient click on services button
3.	Patient click on feedback button to add feedback
4.	Patient add feedback by commenting or by rating through stars
5.	Patient click on the submit button
Alternate Scenario:	
a. If the patient will not write anything in the commenting box the system will generate an error. b. If the patient double clicks on the star the star will be unfilled, the feedback will not submit.	
Post-conditions:	Feedback is added
Exceptions: Nil	
UC cross Reference:	FR-20

Table 57: UC-20: Add Feedback

UC-21: View Feedback

Use case name:	View Feedback
Use case id:	21
Actors:	Patient and Admin
Use Case Summary	User visits the website, open feedback page and view feedback about the Doctor given by him/herself or by other users
Pre-conditions:	User must be registered
Normal Course of Event(Normal Flow)	
Steps#	Description
1.	User visit the web site
2.	User click on feedback button to view feedback of the website
3.	User click on particular doctor to view the feedback of that specific doctor
4.	User click on feedback button to view feedback of that doctor
5.	User click on the home button after viewing the feedback
Alternate Scenario:	
a. If the user will not choose any teacher the system will not show any feedback	
Post-conditions:	Feedback is shown to the user according to the option selected by him/her.
Exceptions:	Nil
UC cross reference	FR-21

Table 58 UC-21: View Feedback

UC-22: Add List of Doctors

Use case name:	Add List of Doctors
Use case id:	22
Actors:	Admin
Use Case Summary	This use case helps the admin to add a list of doctors.
Pre-conditions:	This list of the doctor should not exist already.
Normal Course of Event(Normal Flow)	
Steps#	Description
1.	The use case starts with the admin wanting to make changes in the list of doctors.
2.	The admin wants to add a doctor that does not exist on the list.
3.	The admin wants to make changes in the description of a doctor.
Alternative course of event:	
a: The Admin does not want to add a doctor that does not exist on the list.	
Post-conditions:	New doctor account will be added in doctor list.
Exceptions:	Nil
UC cross reference	FR-22

Table 59 UC-22: Add List of Doctors

UC-23: Delete List of Doctors

Use case name:	Delete list of doctors
Use case id:	23
Actors:	Admin
Use Case Summary	This use case helps the admin to delete a specific doctor from the list of doctors
Pre-conditions:	If admin try to delete doctor by searching the name then that doctor should be exist in the system.
Normal Course of Event(Normal Flow)	
Steps#	Description
1.	The use case starts with the admin wanting to make delete doctor from the list of a doctors.
2.	The admin visits the list of a doctors.
3.	The admin views the doctor's name.
4.	The admin selects the delete option.
5.	The system will display a confirmation message.
6.	The admin will select either yes or no
7.	The Particular doctor is deleted
8.	The system will display changes.
Alternative course of event	
a. Amin does not want to do any change.	
b. Cancel the action.	

Post-conditions:	Existing doctor account will be deleted from doctor list.
Exceptions:	Nil
UC cross reference	FR-23

Table 60: UC-23: Delete List of Doctors

UC-24: Logout

Use case name:	Logout
Use case id:	25
Actors:	User (Patient, Doctor, Admin)
Use Case Summary	The admin and user must be logging into the system
Pre-conditions:	If admin try to delete doctor by searching the name then that doctor should be exist in the system.
Normal Course of Event(Normal Flow)	
Steps#	Description
1.	The use case starts when the user and admin want to exit from the application.
2.	The admin and user click on the logout option.
3.	The system will display a confirmation message that “Are you sure you want to Logout”.
Alternative course of event	
a. The page doesn't open.	
b. The admin and user don't want to logout.	
c. Cancel action(Don't confirm the logout).	
Post-conditions:	User will be log outed. Application will redirect the user to login page.
Exceptions:	Nil

UC cross reference	FR-25
-------------------------------	-------

Table 61: UC-24: Logout

2.5 Use Case Design

2.5.1 Patient and Doctor

If the user is a patient, then user will have an option to signup/login and logout, user will be able to give feedback, view the schedule, add an appointment, will get notifications, will be able to view and edit profile and get/edit and delete an appointment.

If the user is a doctor, then user will be able to signup/login and logout, user will get notifications user will be able to view and edit profile, update and delete an appointment.

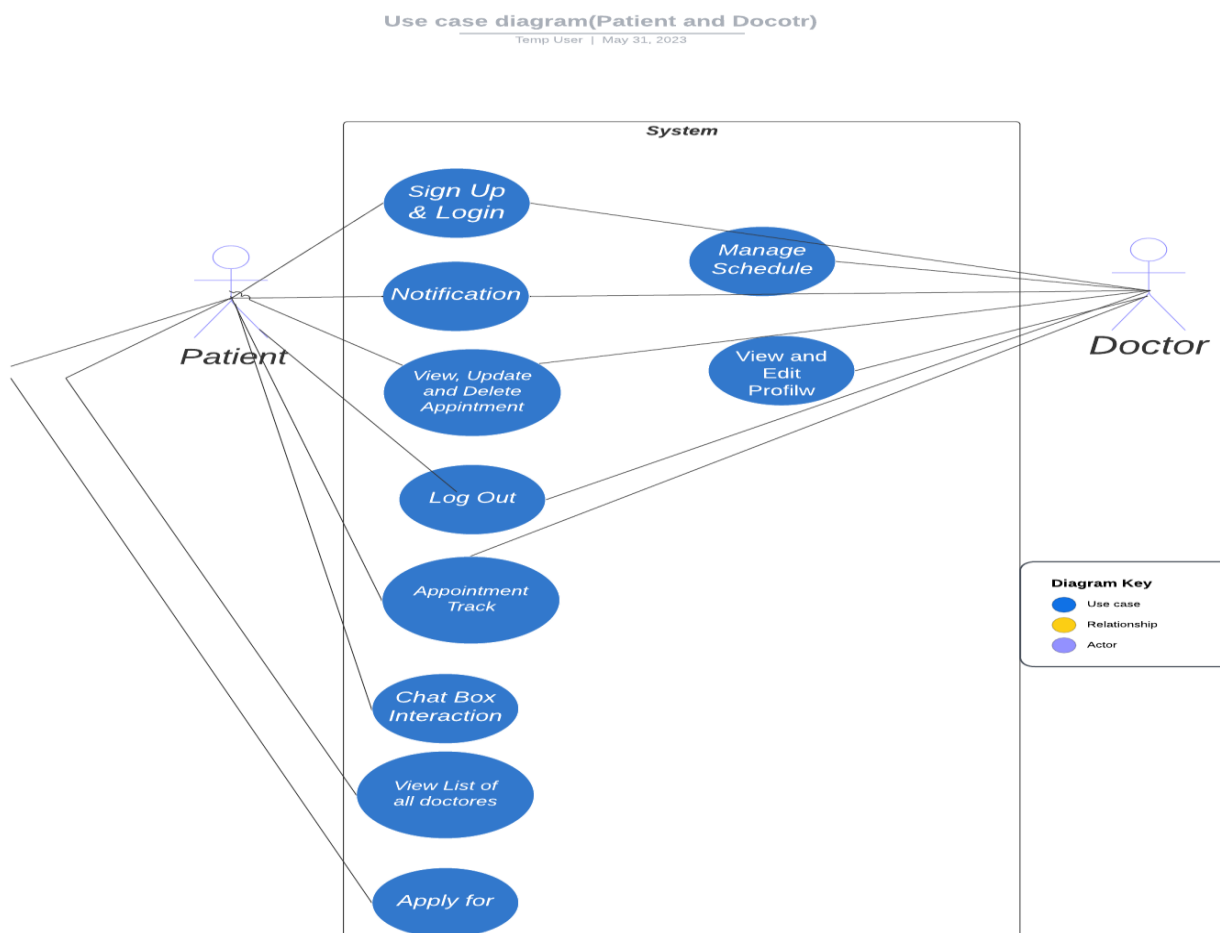


Figure 1: use case diagram for user and doctor

2.5.2 Admin Panel:

Admin will be able to login, view schedule, add a doctor, view profile, delete/update doctor list, view feedback and logout.

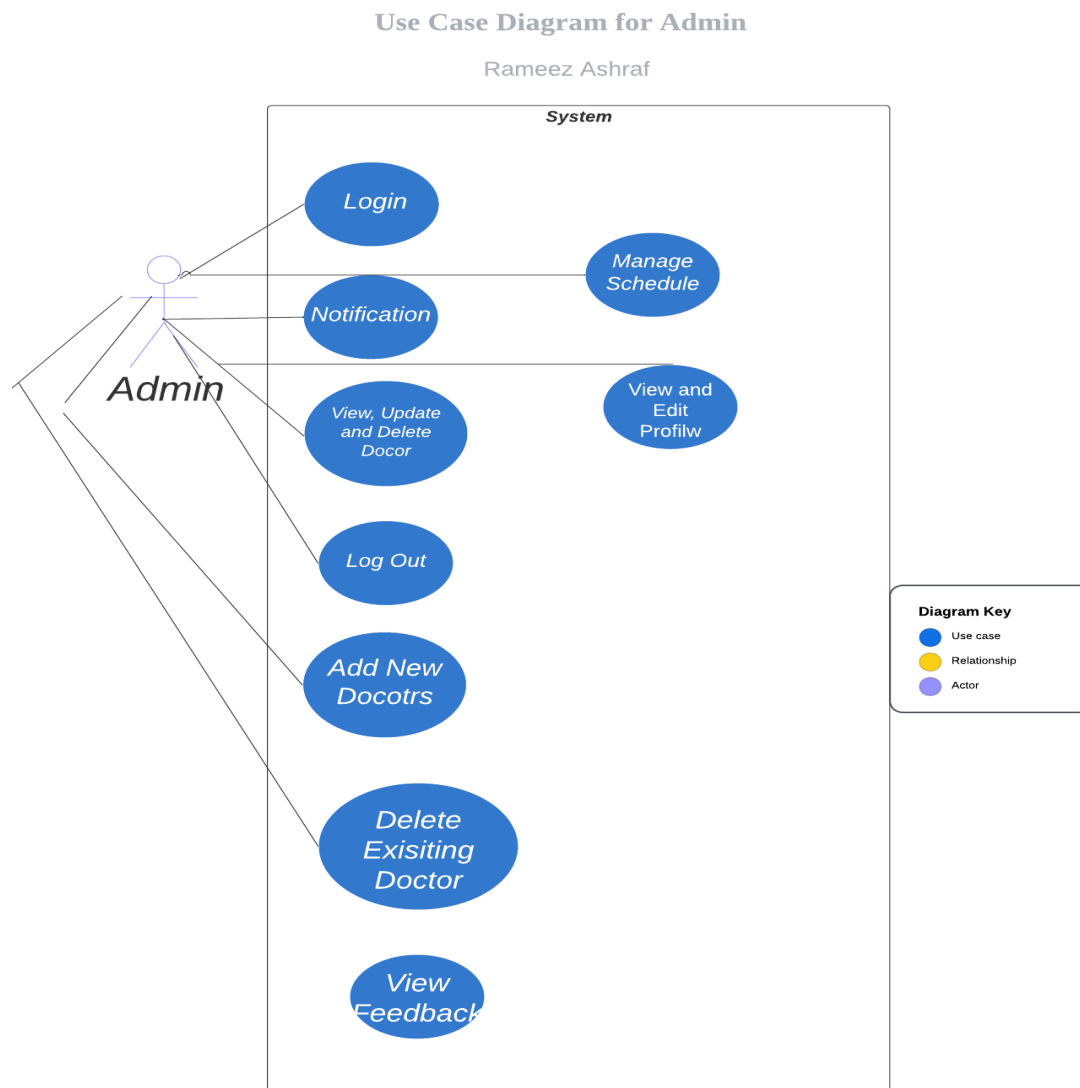


Figure 2: use case diagram for Admin Portal

2.6 Software Development Life Cycle Model:

In our project Software Development Life Cycle (SDLC) we are using is **Incremental Model**. We divide our project into builds/increments. The incremental model first build/develop the prioritizes requirements and then add the new/advance functionalities.

Reasons:

Some basic reasons to use the Incremental Model are:

- Requirements are clearly defined and understood.
- Major Requirements are provided, some functionalities may apply with the passage of time.
- This model break the project into individual modules.
- We can publish the Product as we complete at initial and basic requirements

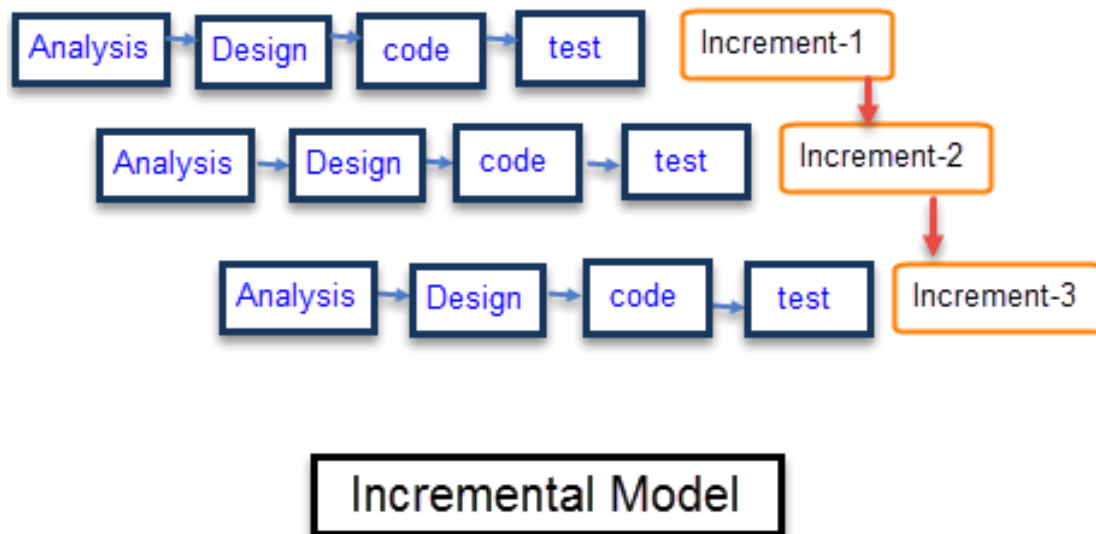


Figure 3: SDLC Model

Chapter 3

System Design

3 System Design

3.1 Work Breakdown Structure (WBS)

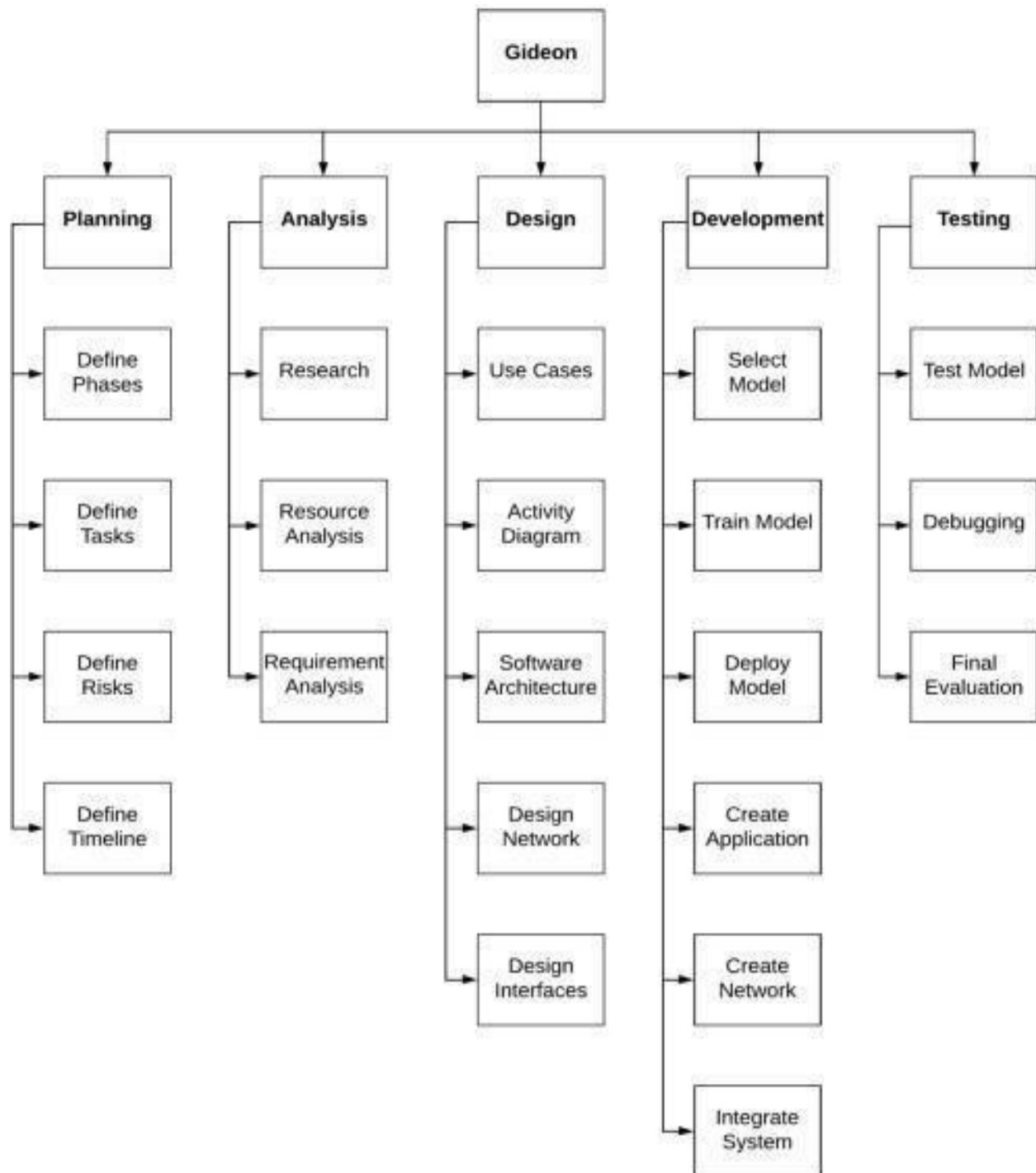


Figure 4: Work Breakdown Structure (WBS)

3.2 Activity Diagrams

3.2.1 AD-01: User Sign Up

A user who wants to register will have two options for signup i.e., enter his credentials or he can sign up from his google account. Once user submit the credentials and they are invalid or already in use then the user will be asked to enter again otherwise he/she will be registered.

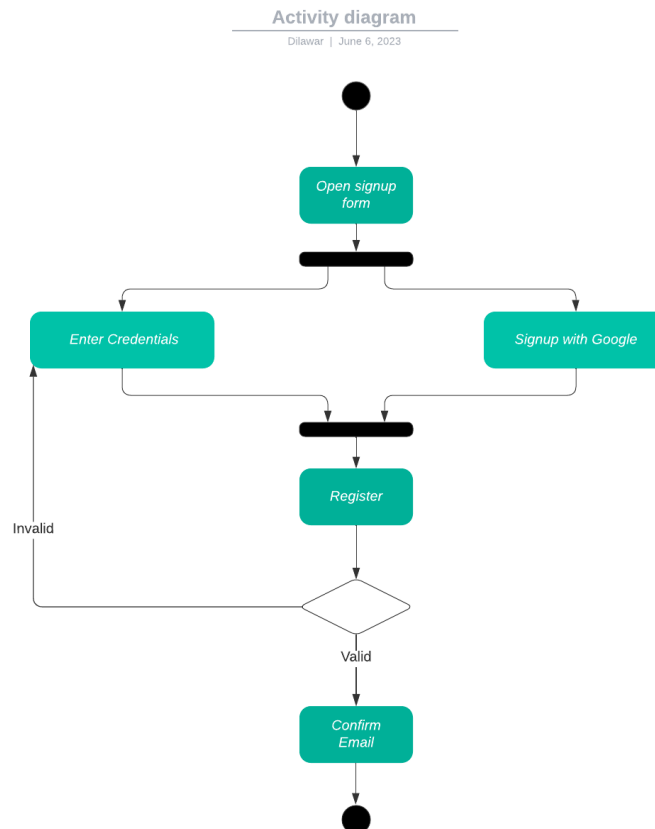


Figure 5: AD- Sign Up

3.2.2 AD-02: Email Verification:

A user who wants to register will have two options for signup i.e., enter his credentials or he can sign up from his google account. Once user submit the credentials he/she will receive the Verification Email. By clicking on this the user will get registered.

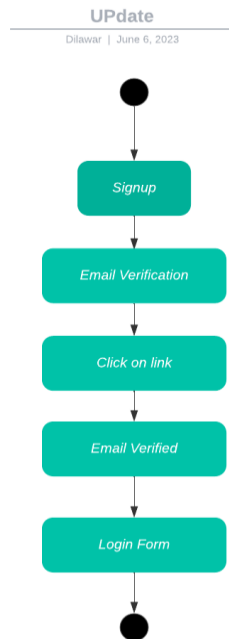


Figure 6: AD- Email Verification

3.2.3 AD-03: User Login:

If the user is already register then he will open login and enter his name and password. If the credentials are correct the user will be led to user's homepage, else the user will be asked to enter the credentials again.

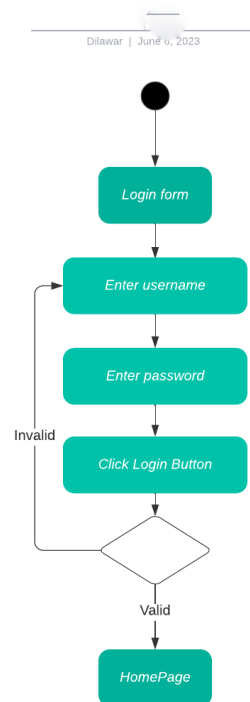


Figure 7: AD- Sign In

3.2.4 AD-04: Password reset:

A user can reset his password by clicking on the forgot password in login form. A link will be provided to him on his registered email. By clicking the link, he can set new password for the account.

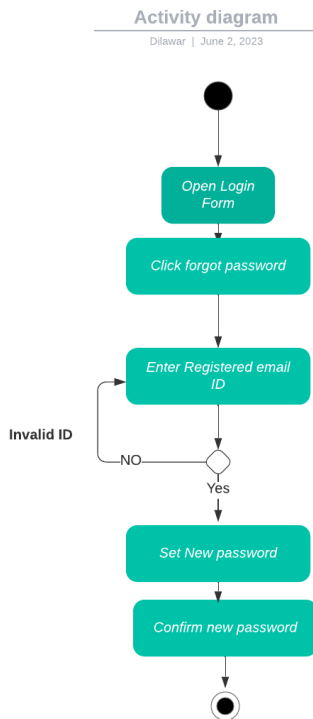


Figure 8: Password Reset

3.2.5 AD-05: Doctor List:

A user will first login and will be led to home page from where the user can see doctor list by clicking on the Doctor button.

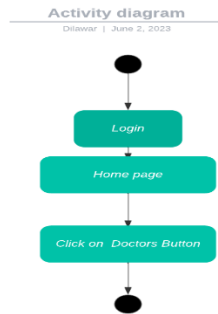


Figure 9: AD- View Doctor List

3.2.6 AD-06: Apply for Doctor:

A user will first login and will be led to home page from where the user will click on the apply for doctor option on the side menu. Then user will fill the form and click the submit button. The request will be forwarded to the admin.

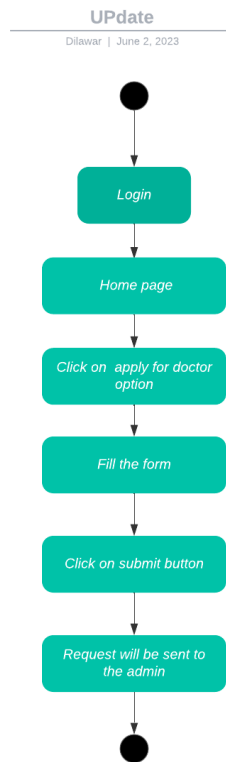


Figure 10 AD- Apply for Doctor

3.2.1 AD-07: Edit profile:

A user will first login and will be led to home page from where the user will view profile by clicking on the profile Icon on top bar. Then user will click the edit button to edit his profile. Then he will click on save button to save changes.

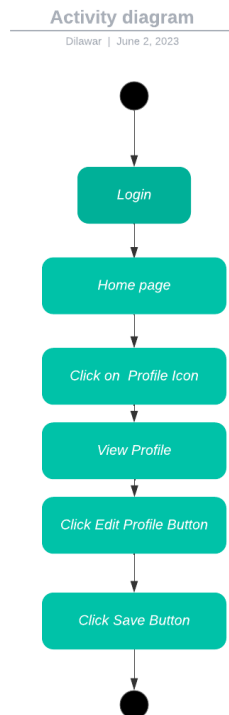


Figure 11 AD-Edit Profile

3.2.2 AD-08: View profile:

A user will first login and will be led to home page from where the user can view profile by clicking on the profile Icon on top bar.

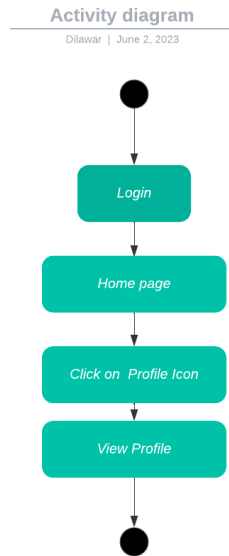


Figure 12 AD- View Profile

3.2.3 AD-09: Schedule Management:

A user will first login and will be led to home page from where the user will click on profile icon. Then he will click on schedule field and will manage his schedule.

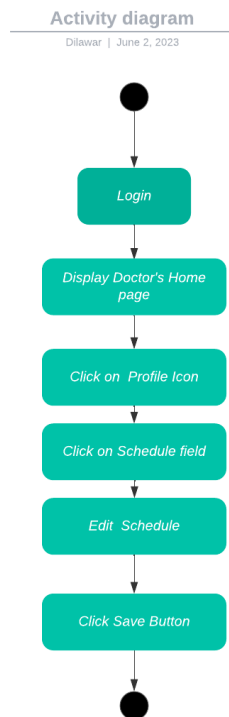


Figure 13 AD- Schedule Management

3.2.4 AD-10: Searching / Filtering:

A user will first login and will be led to home page from where the user can search for doctors by clicking on search bar and can also filter the searching by clicking on the filter button next to it.

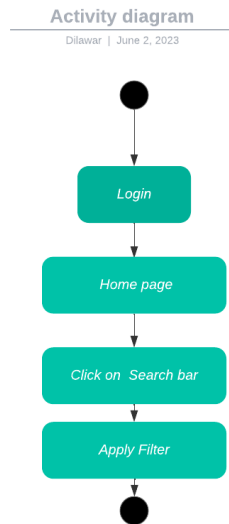


Figure 14 AD- Searching

3.2.5 AD-11: Appointment Booking:

A user will first login and will be led to home page from where the user will search for specific doctor using search bar. Then he will click on booking, to check available slots. After choosing the slots he will click on confirm booking button to book the appointment.

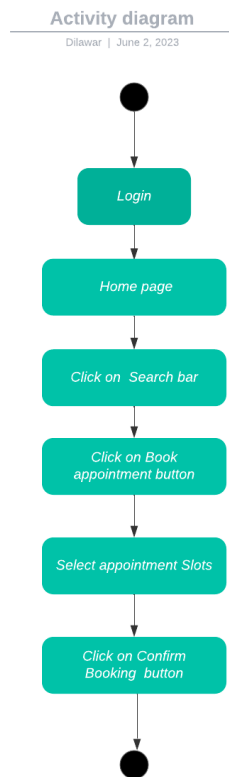


Figure 15 AD- Appointment Booking

3.2.6 AD-12: Appointment Tracking:

A user will first login and will be led to home page. Then he will click on appointment option from the side menu. There he can track the upcoming appointment.

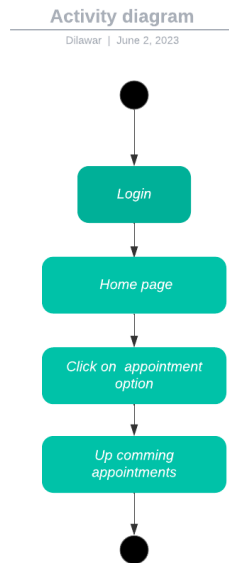


Figure 16 AD- Appointment Track

3.2.7 AD-13: Appointment History:

A user will first login and will be led to home page. Then he will click on appointment option from the side menu. There he can clear the history of previous appointment by clicking on clear history button.

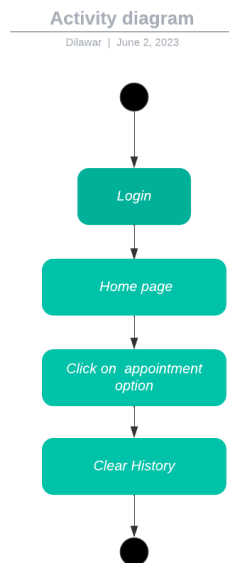


Figure 17 AD- Appointment History

3.2.8 AD-14: Cancel appointment:

A user will first login and will be led to home page. Then he will click on appointment option from the side menu. There he can cancel the appointment by clicking on cancel button (placed on right side of that appointment).

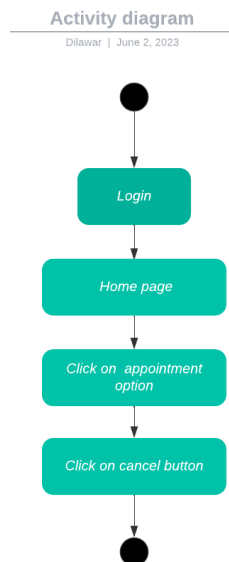


Figure 18 AD- Cancel Appointment

3.2.9 AD-15: Update appointment:

A user will first login and will be led to home page. Then he will click on appointment option from the side menu. There he will click on update button which is placed on the appointment he wants to update. Then he will make required changes and will click on confirm updation button to update the appointment.

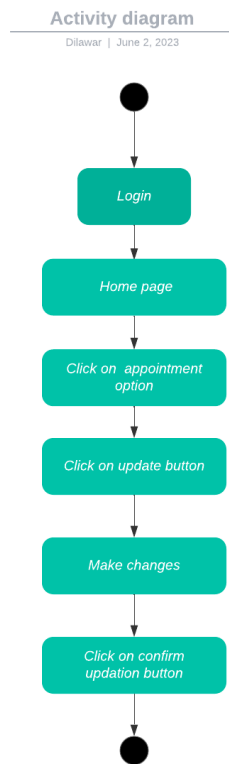


Figure 19 AD- Update Appointment

3.2.10 AD-16: Notification System:

A user will first login and will be led to home page. Then he will click on the notification icon on the top bar to see notifications.

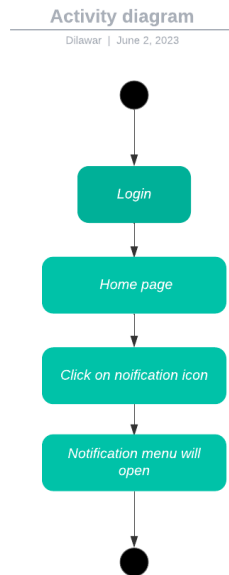


Figure 20 AD- Notification System

3.2.11 AD-17: Chat Box:

A user will first login and will be led to home page. Then he will click the chat button placed at the bottom. The chat box will open.

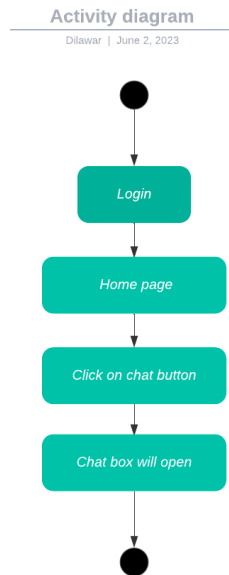


Figure 21 AD- Chat box

3.2.12 AD-18: View Candidate profile:

An admin will first login and will be led to home page. Then he will click on pending request option from the side menu. On clicking he will see candidate profile who apply for doctor's role.

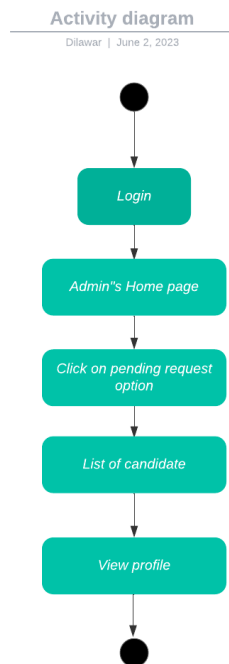


Figure 22 AD- View Candidate Profile

3.2.13 AD-19: Authority to Accept or reject request:

An admin will first login and will be led to home page. Then he will click on pending request option from the side menu. On clicking he will see candidates' profile who apply for doctor's role. There he can approve or disapprove the requests.

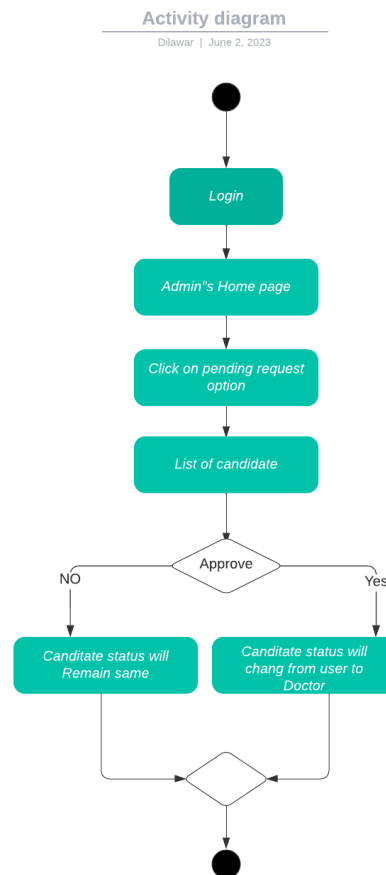


Figure 23 AD- Authority to accept or reject request

3.2.14 AD-20: Add feedback:

A user will first login and will be led to home page. Then he will go to appointment section. If appointment is completed he will click on the feedback button (Placed on the appointment card) to add feedback.

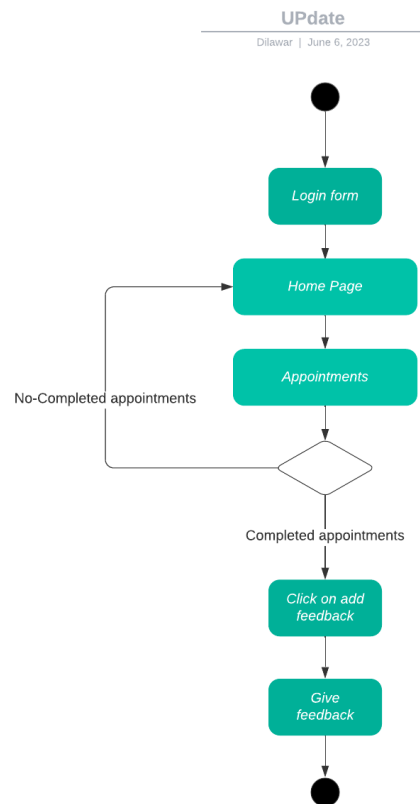


Figure 24 AD- Add Feedback

3.2.15 AD-21: View feedback:

A user will first log in and will be led to home page from where the user will click on doctors' button. A list of doctors will be displayed. There the user will click on the view profile button next to doctor name. There he will click on view feedback button to view the feedbacks.

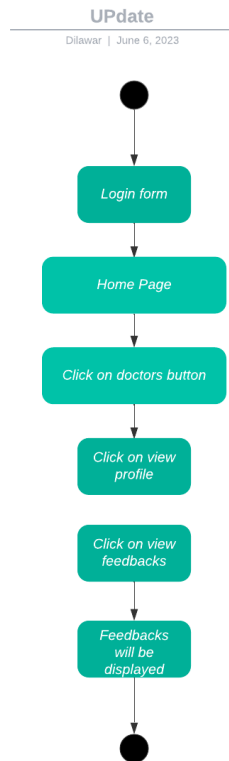


Figure 25 AD- View Feedback

3.2.16 AD-22: Add list of doctors:

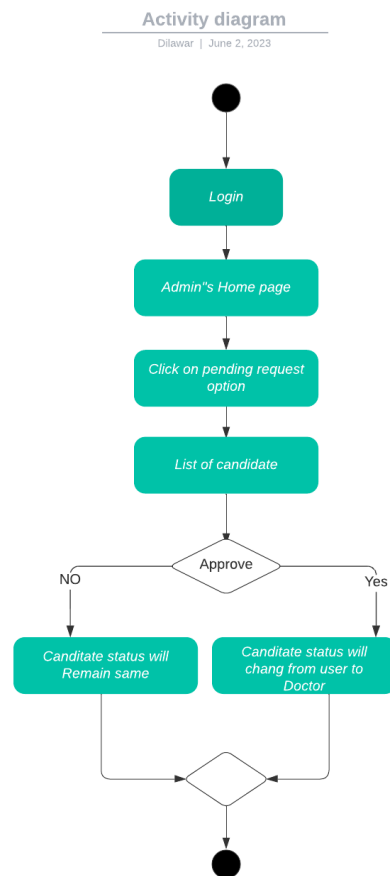


Figure 26 AD- Add List of Doctors

3.2.17 AD-23: Delete list of doctors:

An Admin will first login and will be led to home page from where he will click on Doctors button. There he will click on delete button next to doctors' profile to delete that doctor.

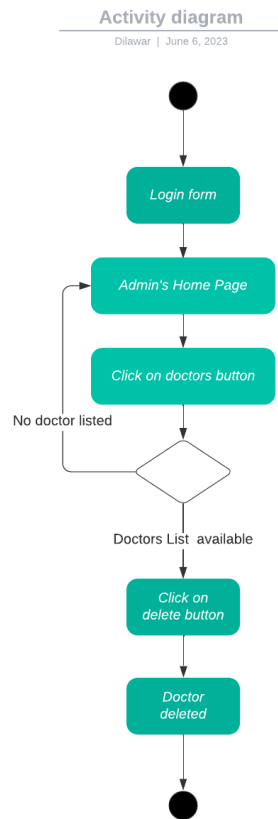


Figure 27 AD- Delete List of Doctors

3.2.18 AD-24: Logout:

A user will first login and will be led to home page from where the user can logout by click on the logout button from the side menu.

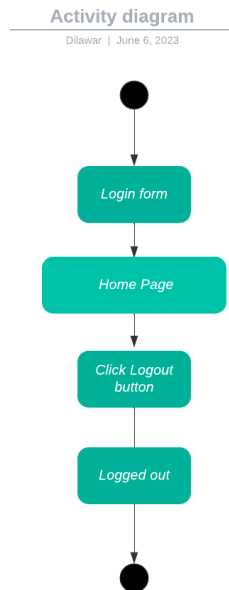


Figure 28 AD- Logout

3.3 Sequence Diagrams

3.3.1 SD-01: User Sign Up

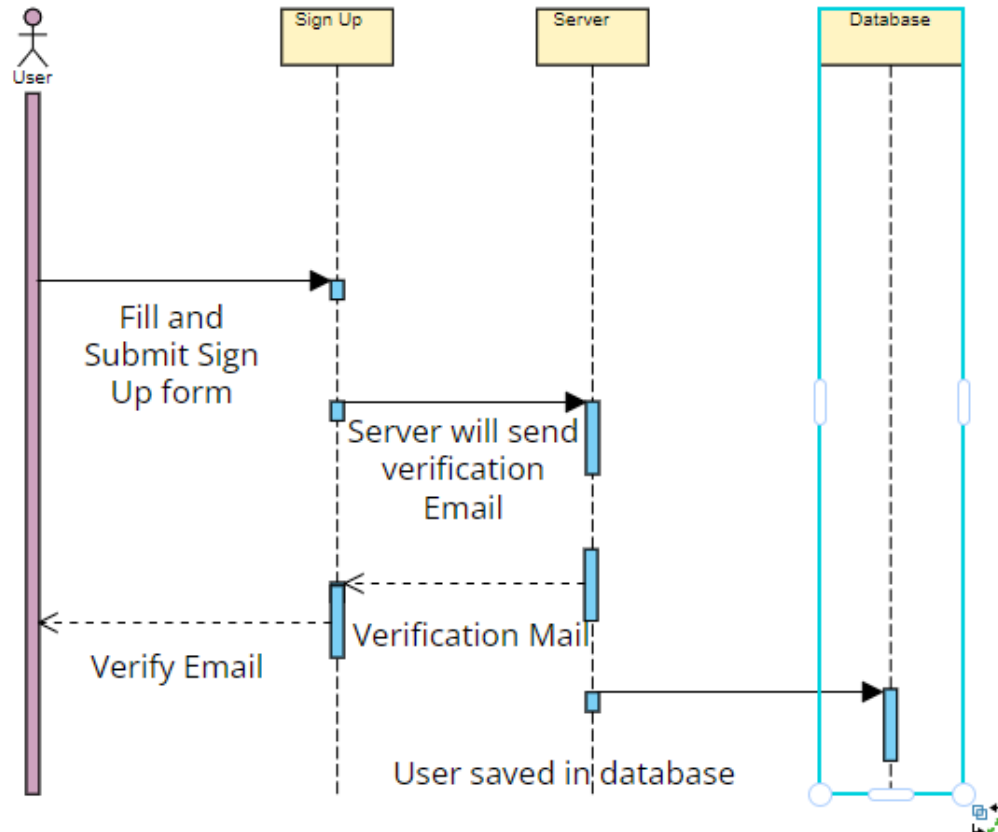


Figure 29 SD- Sign Up

3.3.2 SD-02: Email Verification

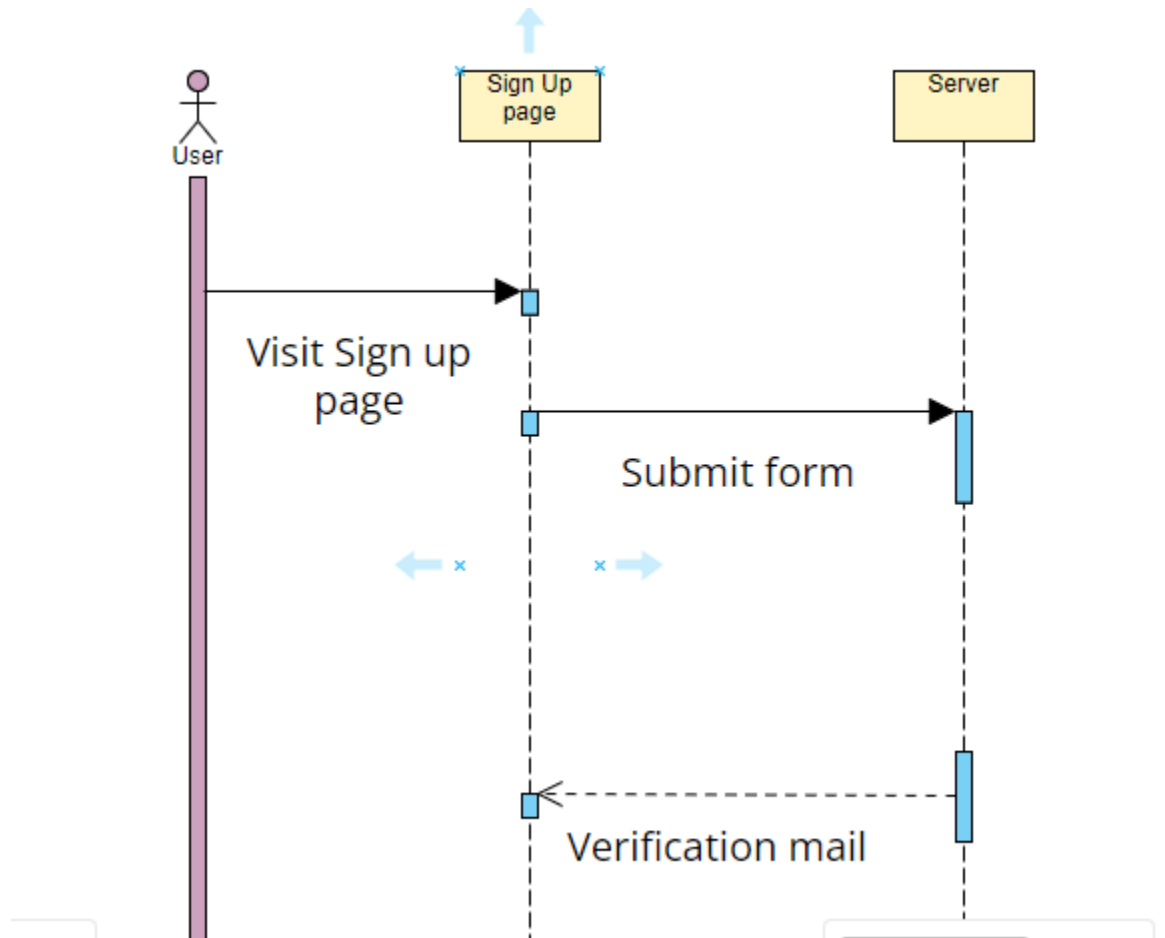


Figure 30 SD- Email Verification

3.3.3 SD-03: Sign In

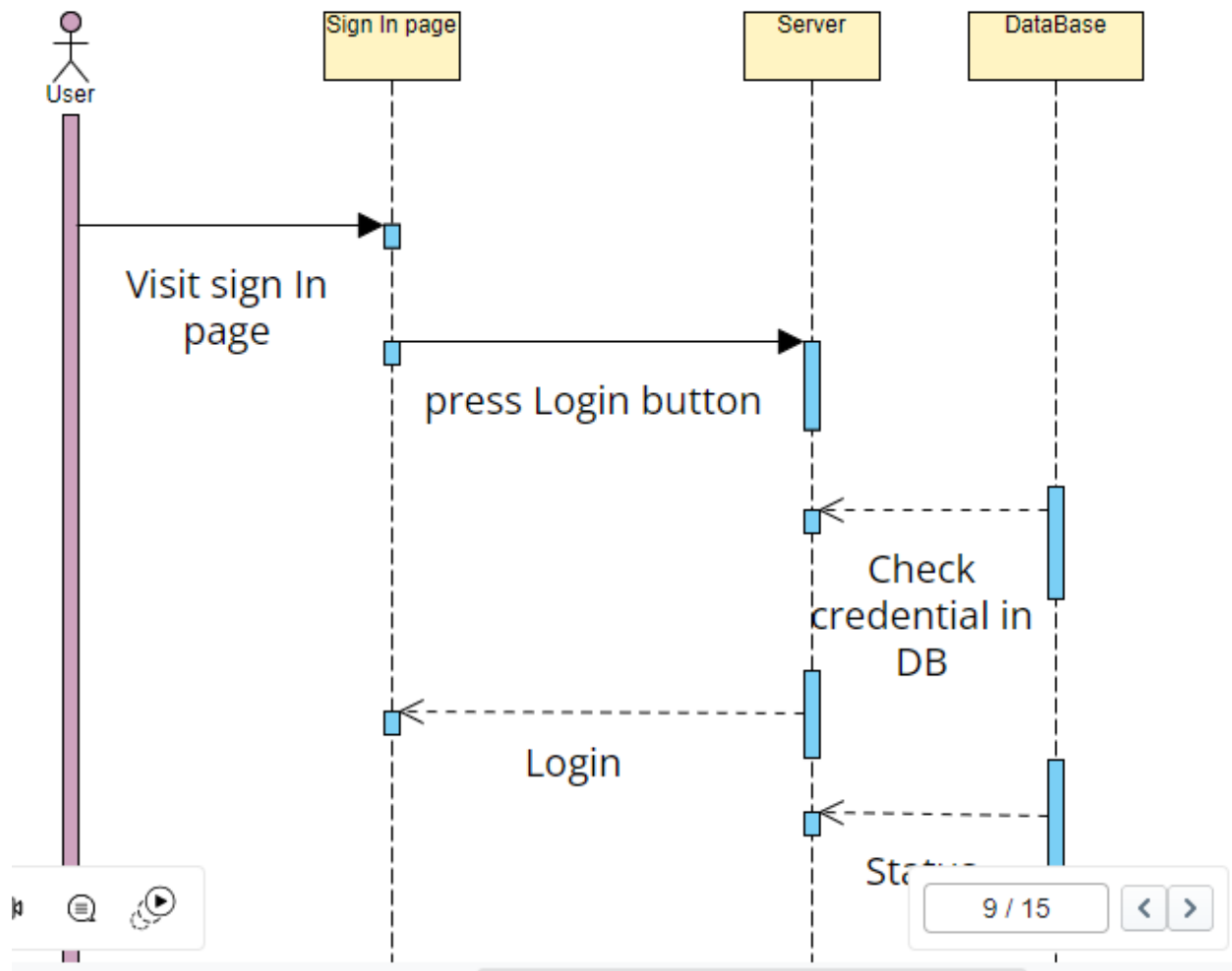


Figure 31 SD- Sign In

3.3.4 SD-04: Password Reset

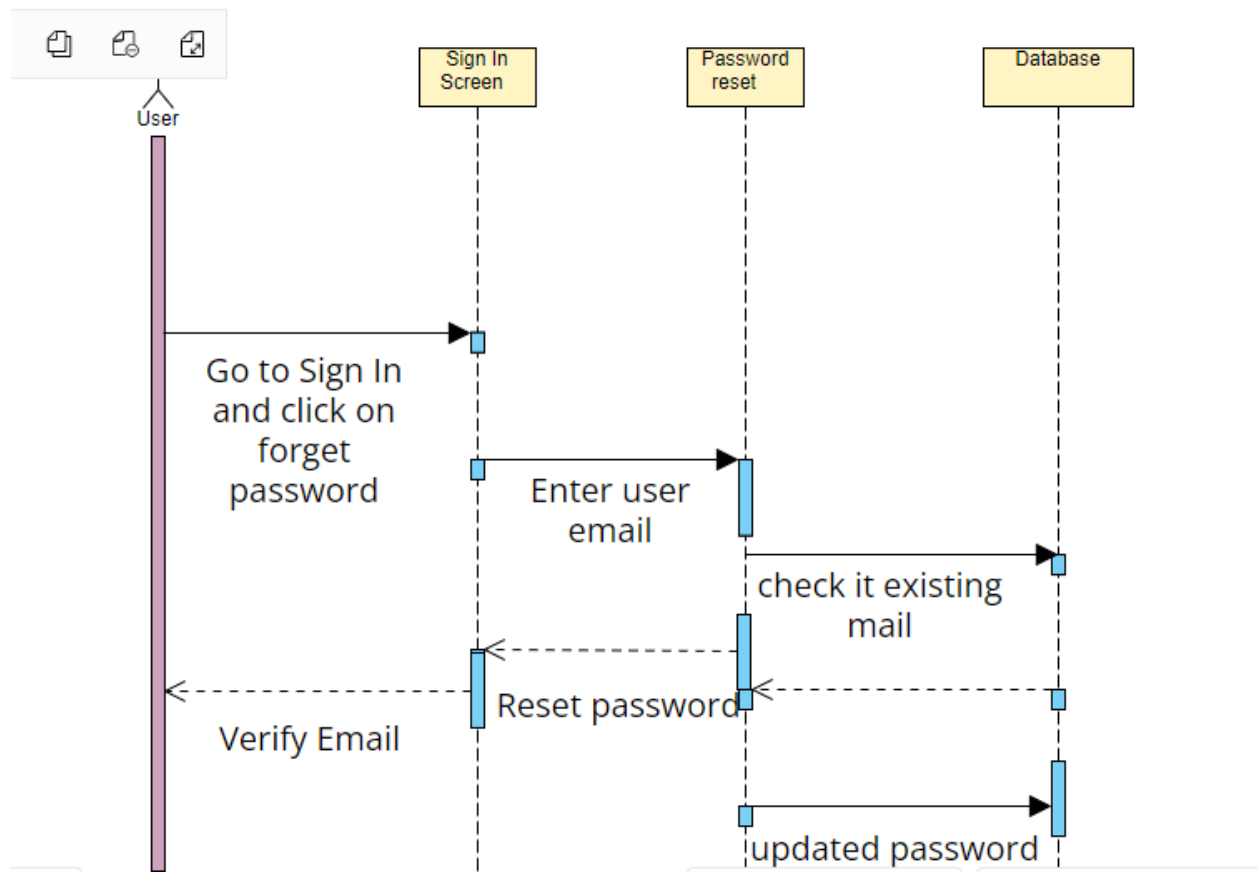


Figure 32 SD- Password Reset

3.3.5 SD-05: View Doctor List

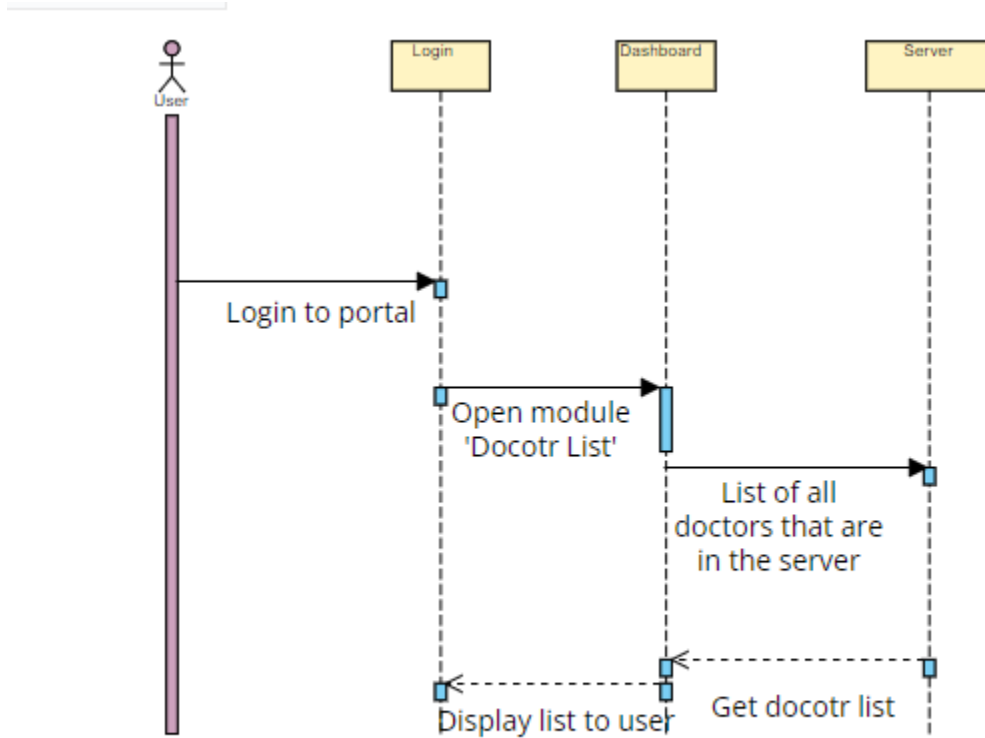


Figure 33 SD- View Doctor List

3.3.6 SD-06: Apply for Doctor

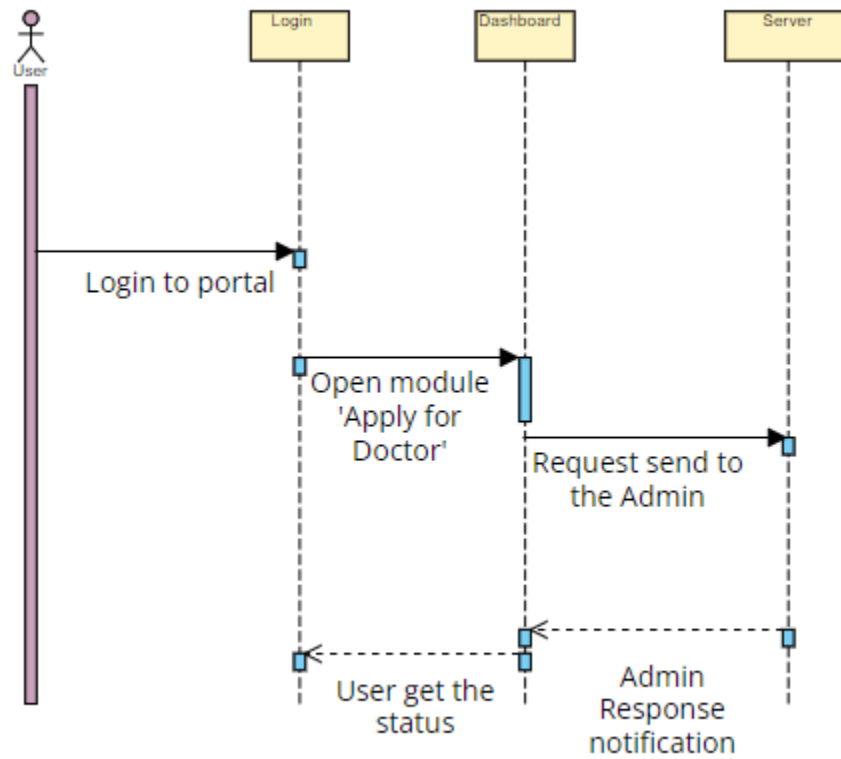


Figure 34 SD- Apply for Doctor

3.3.7 SD-07: Edit Profile

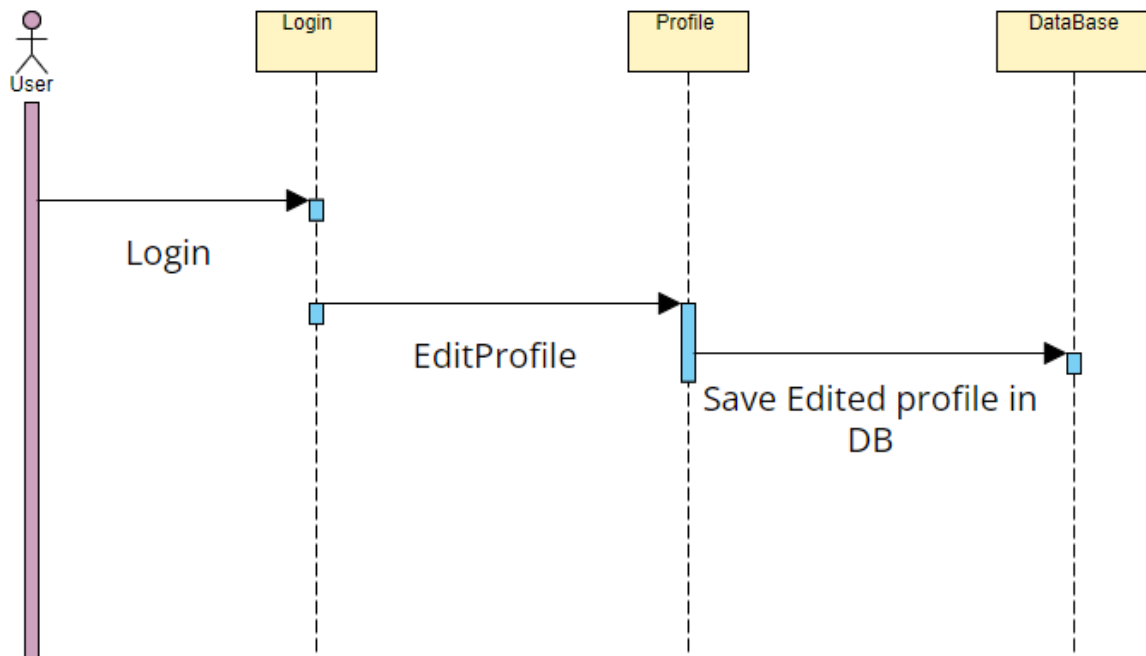


Figure 35 SD- Edit Profile

3.3.8 SD-08: View Profile

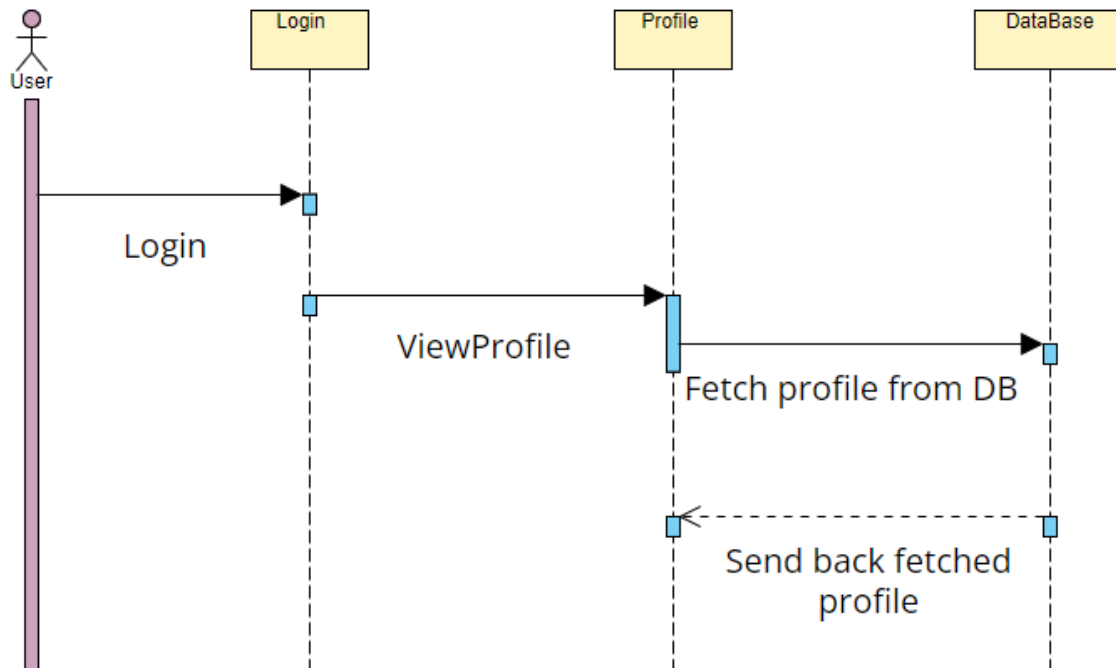


Figure 36 SD- View Profile

3.3.9 SD-09: Schedule Management

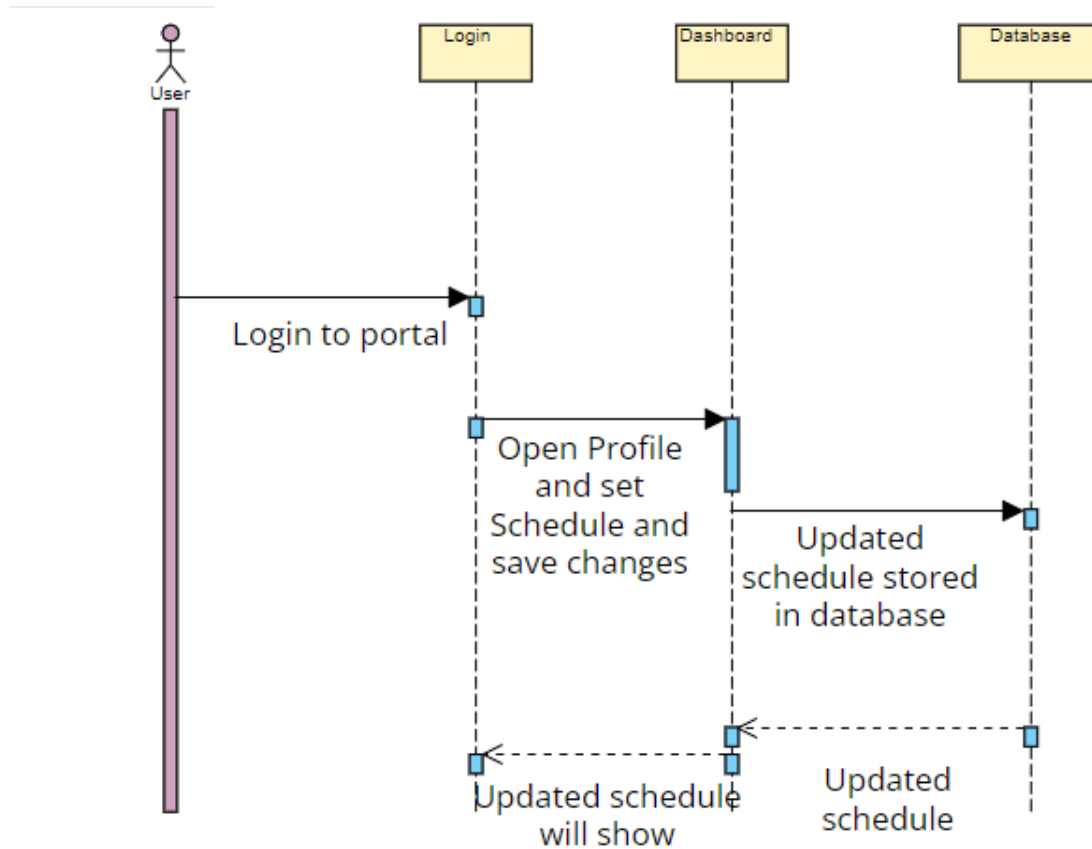


Figure 37 SD- Schedule Management

3.3.10 SD-10: Searching/Filtering

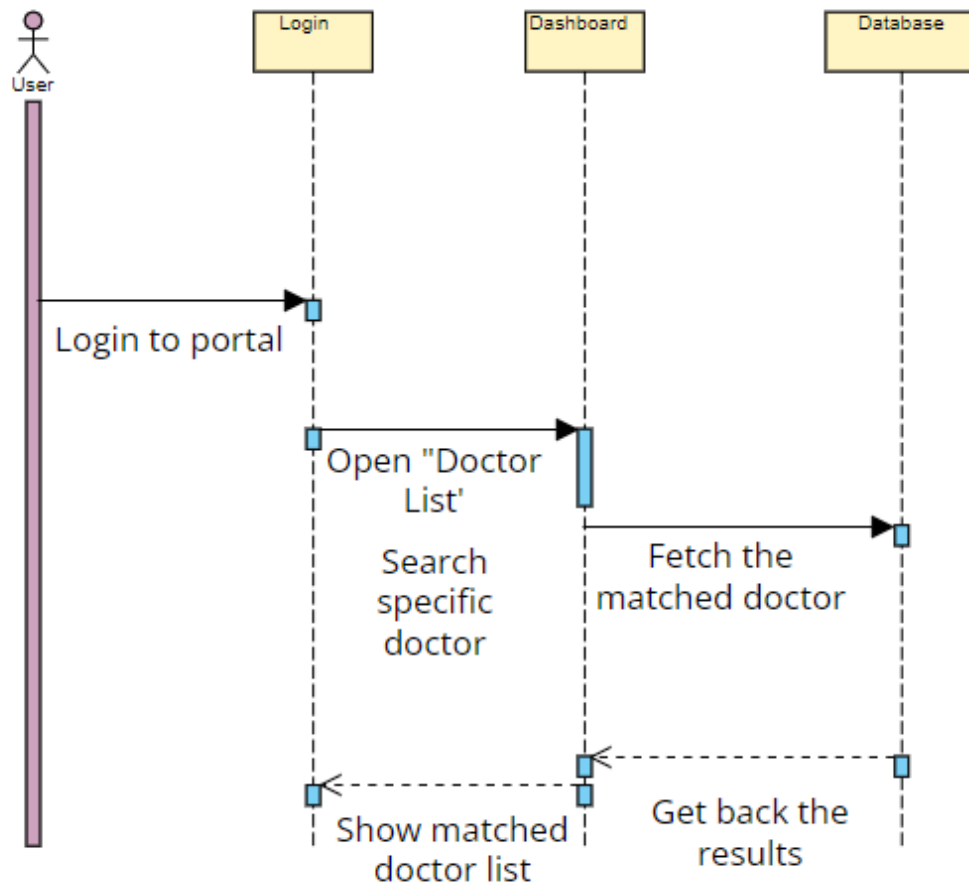


Figure 38 SD- Searching

3.3.11 SD-11: Appointment Booking

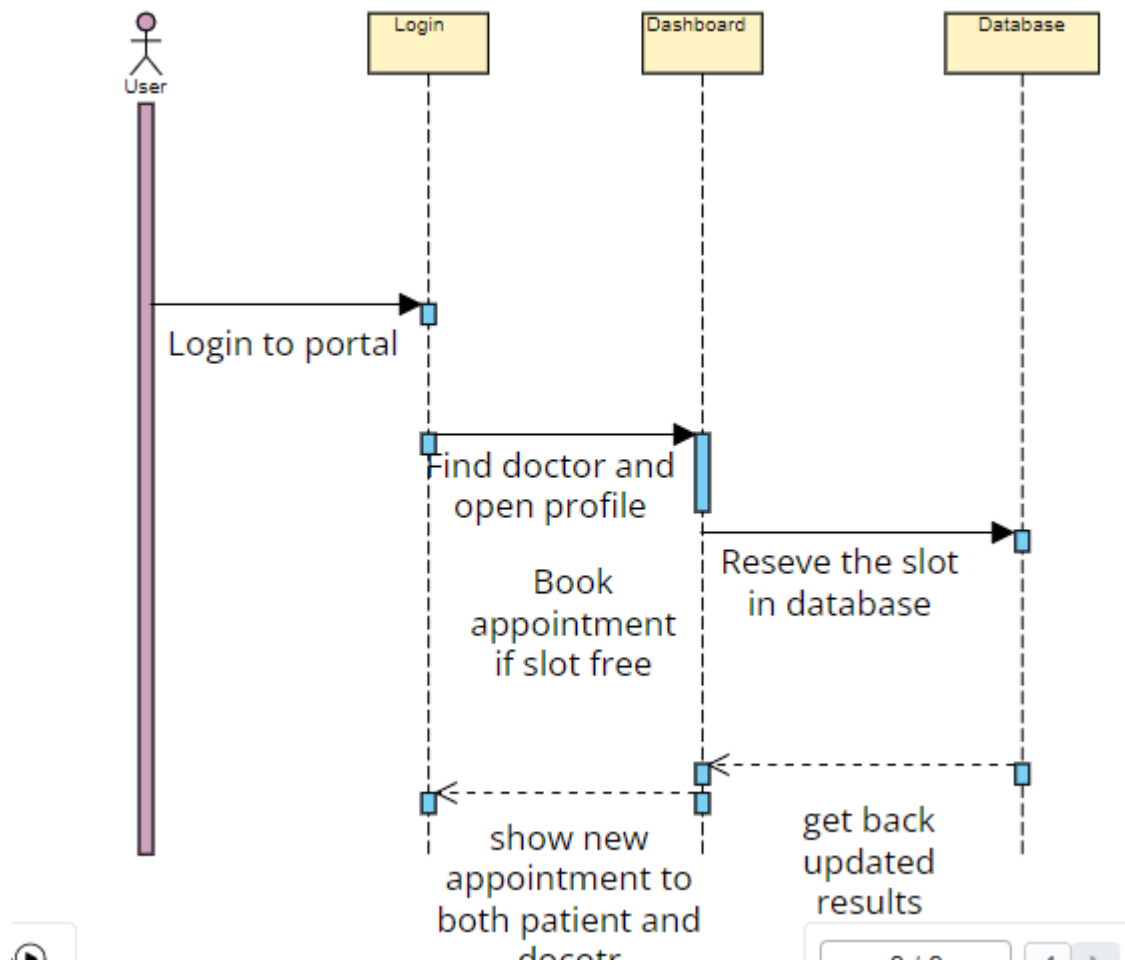


Figure 39 SD- Appointment Booking

3.3.12 SD-12: Appointment Track

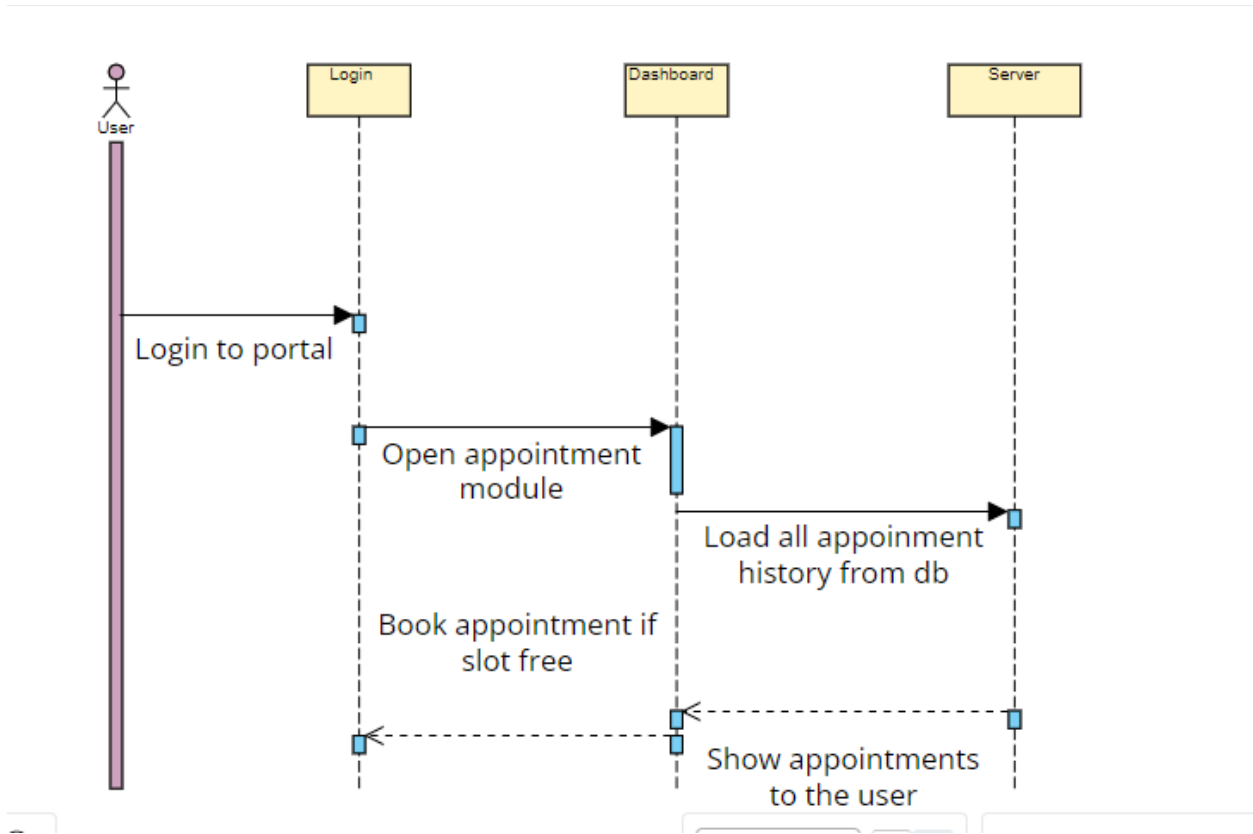


Figure 40 SD- Appointment Track

3.3.13 SD-13: Clear Appointment History

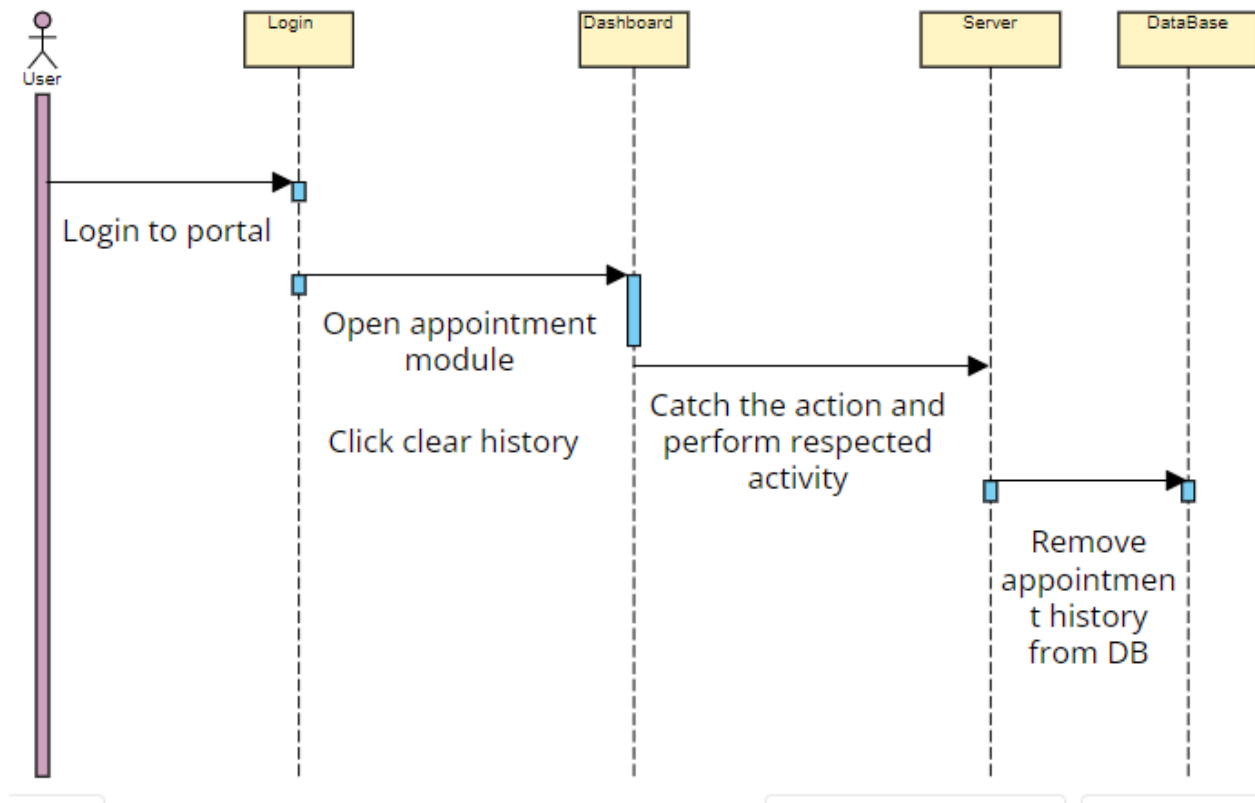


Figure 41 SD- Clear Appointment History

3.3.14 SD-14: Cancel Appointment

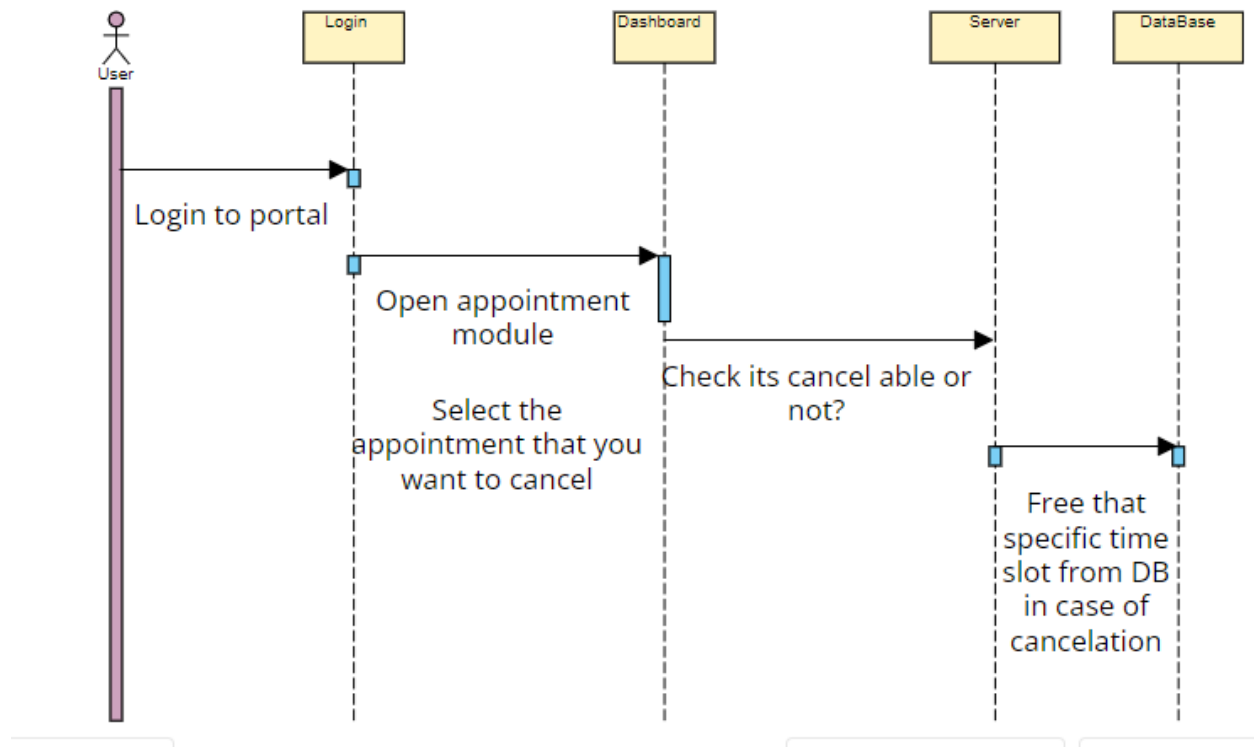


Figure 42 SD- Cancel Appointment

3.3.15 SD-15: Update Appointment

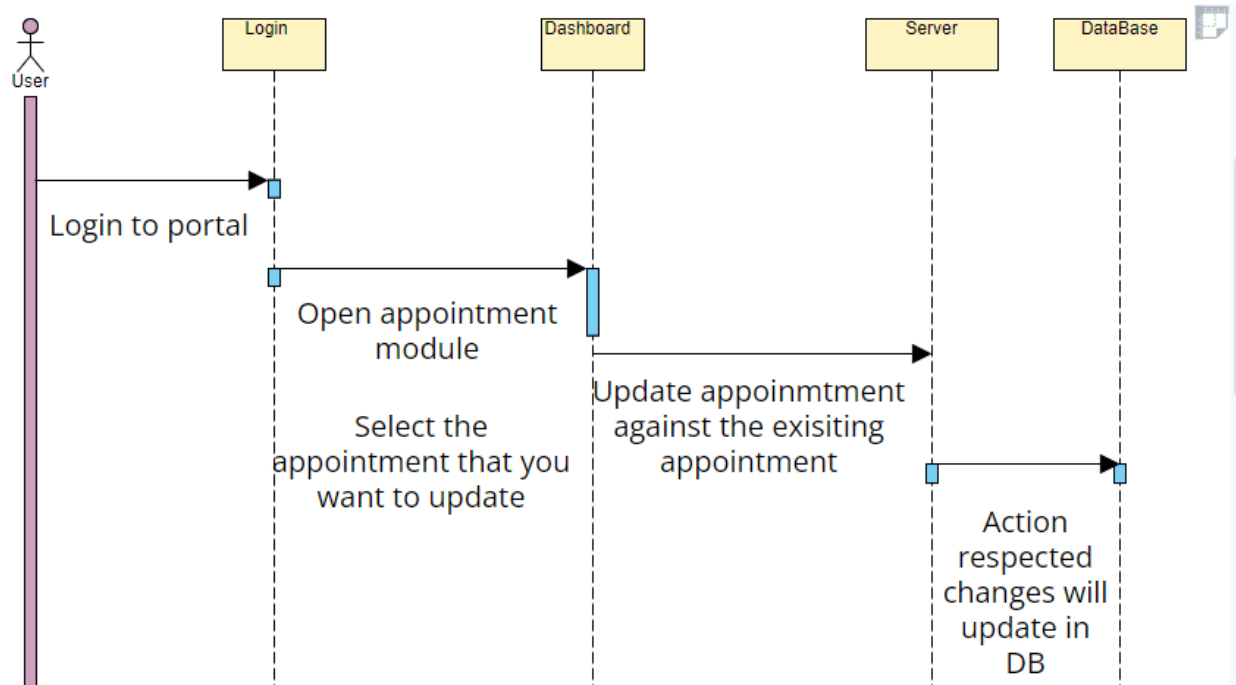


Figure 43 SD- Update Appointment

3.3.16 SD-16: Notification System

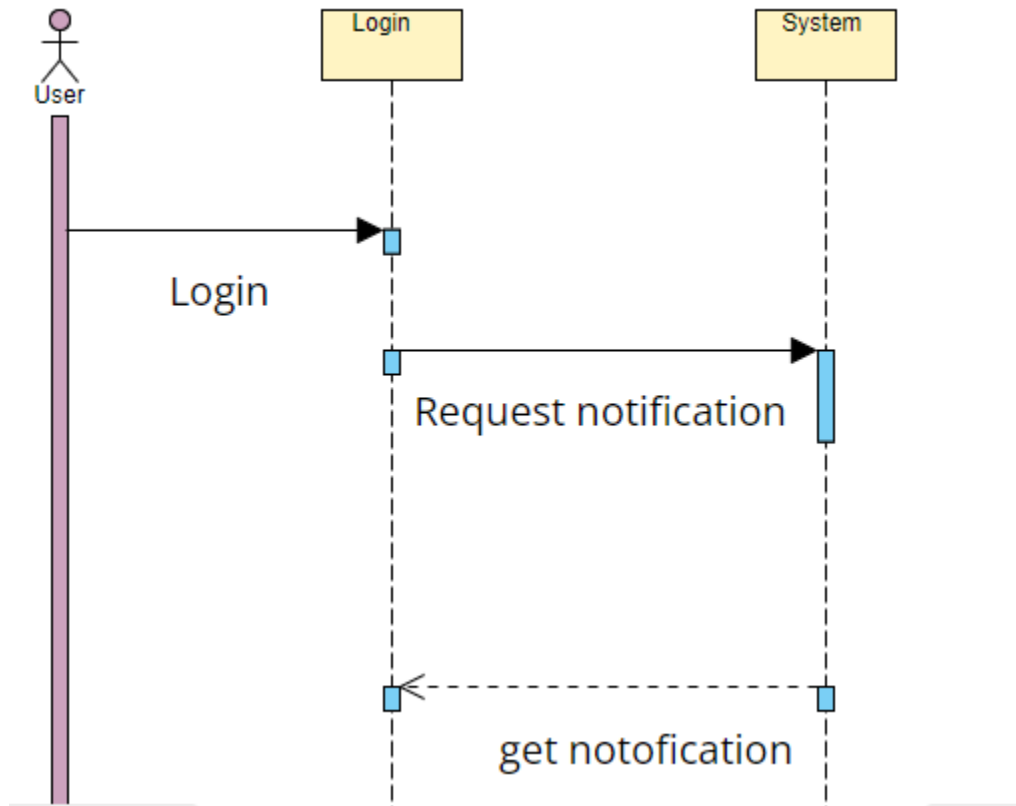


Figure 44 SD- Notification System

3.3.17 SD-17: Chat Box

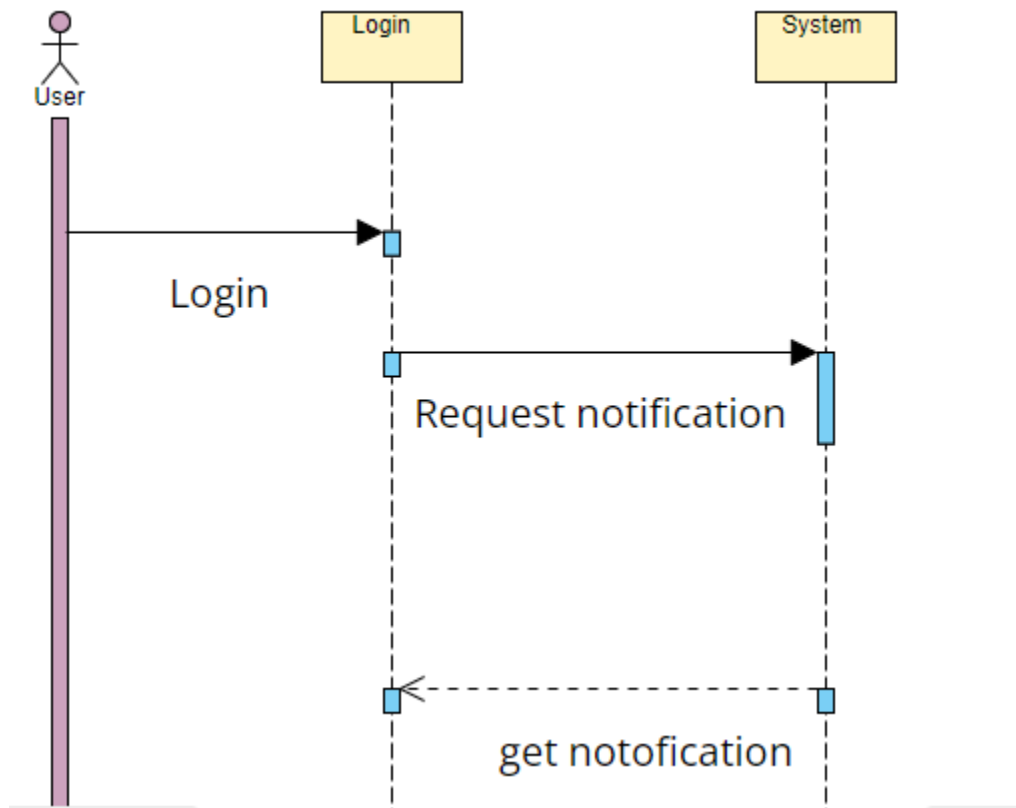


Figure 45 SD- Chat box

3.3.18 SD-18: View Candidate Profile

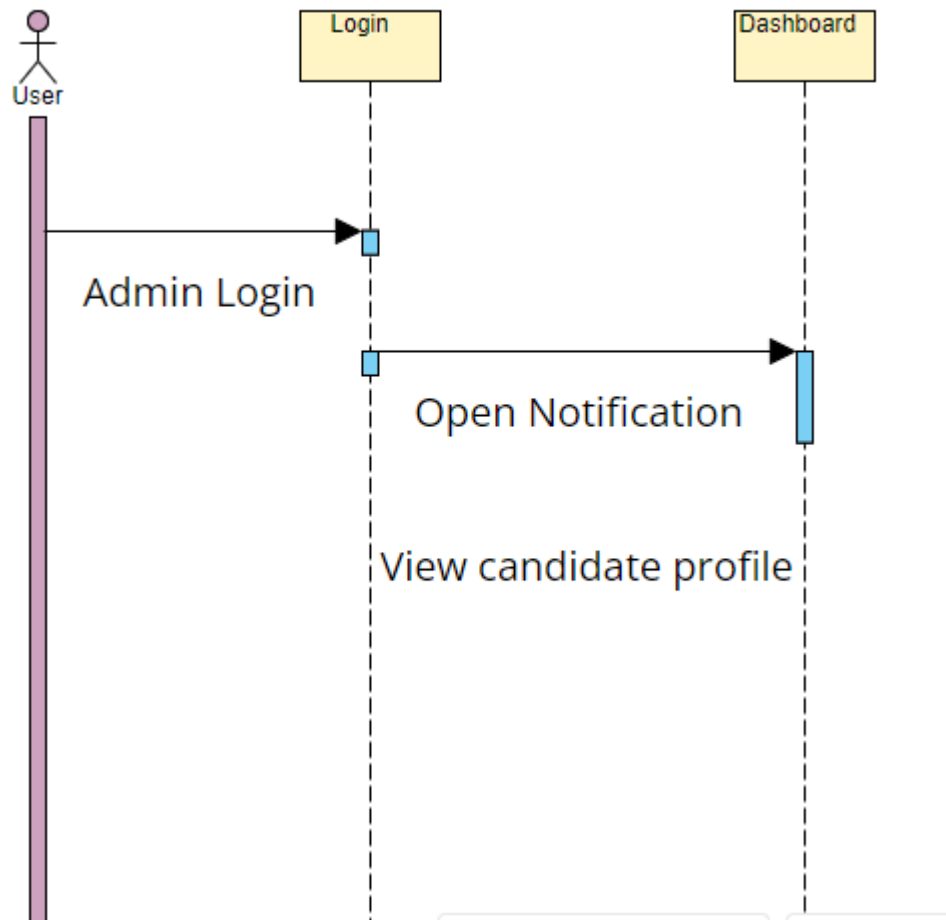


Figure 46 SD- View Candidate Profile

3.3.19 SD-19: Authority to accept or reject the request

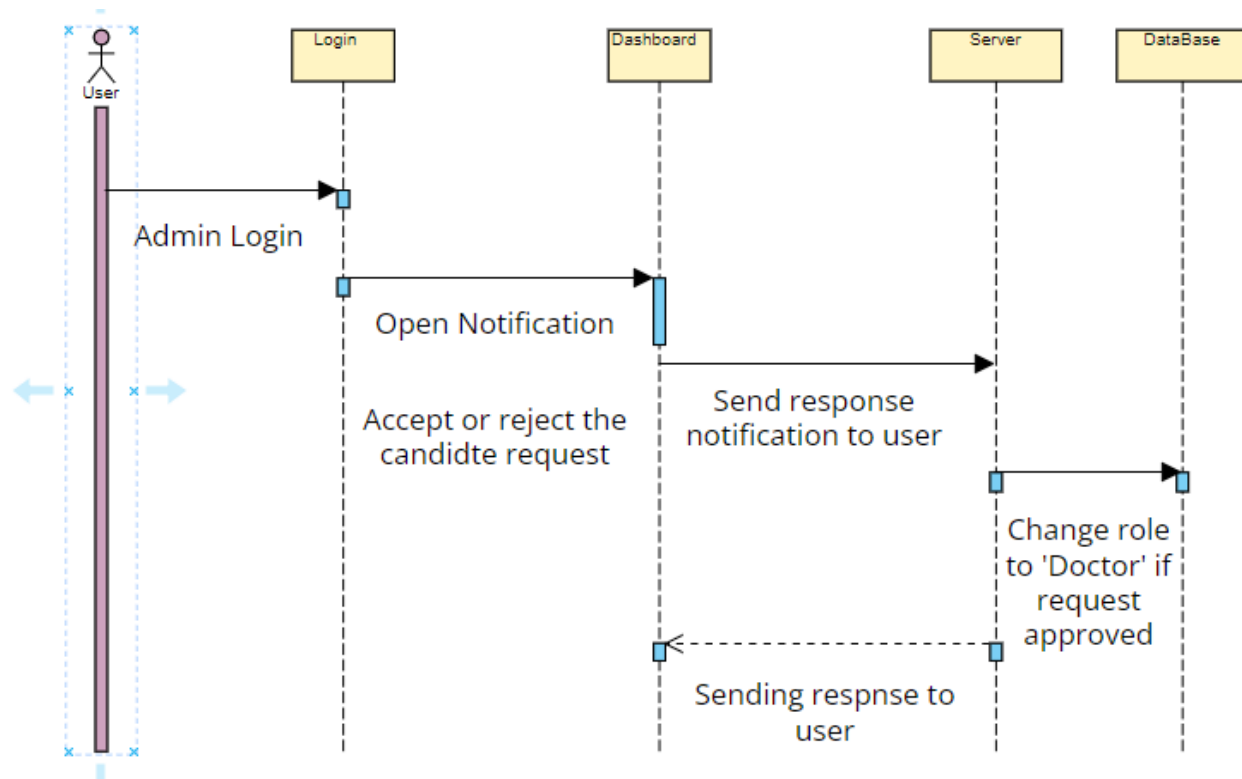


Figure 47 SD- Authority to accept or reject request

3.3.20 SD-20: Add Feedback

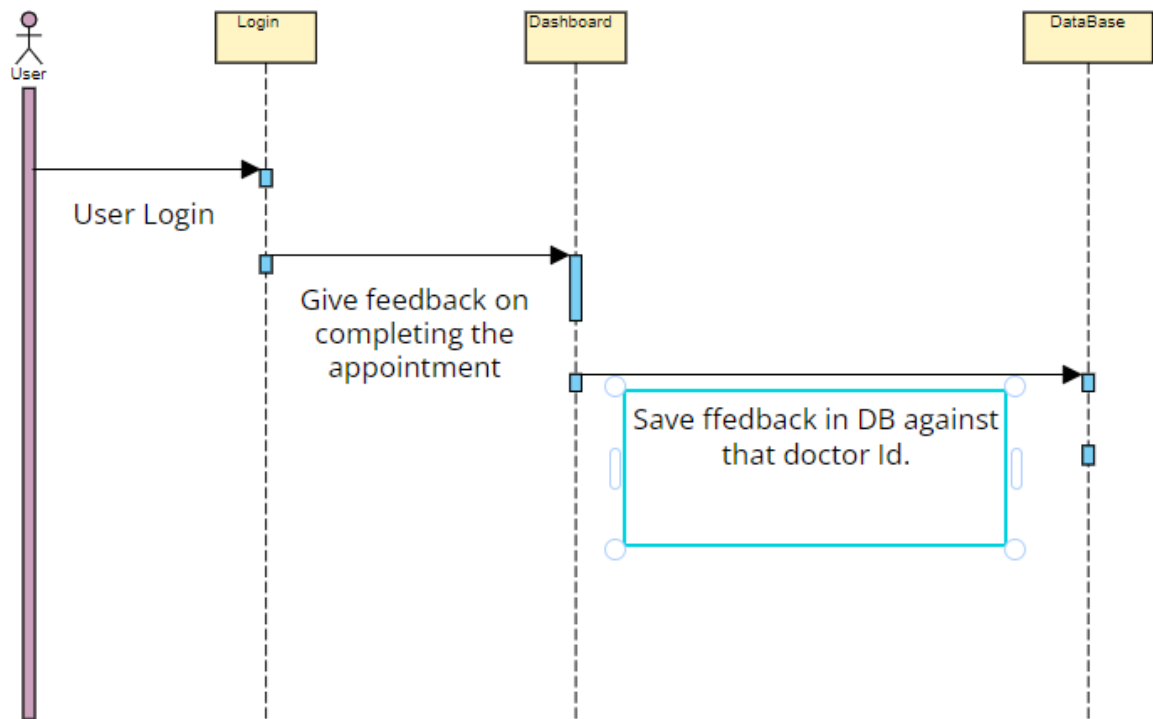


Figure 48 SD- Add Feedback

3.3.21 SD-21: View Feedback

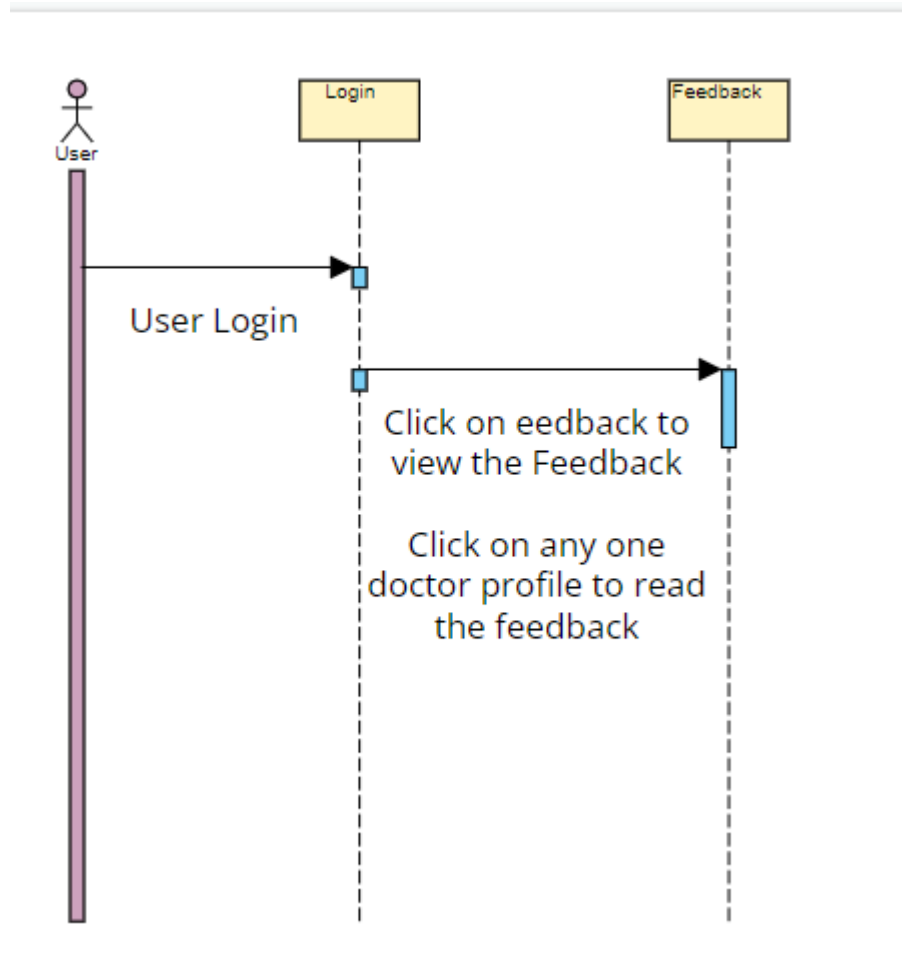


Figure 49 SD- View Feedback

3.3.22 SD-22: Add List of Doctors

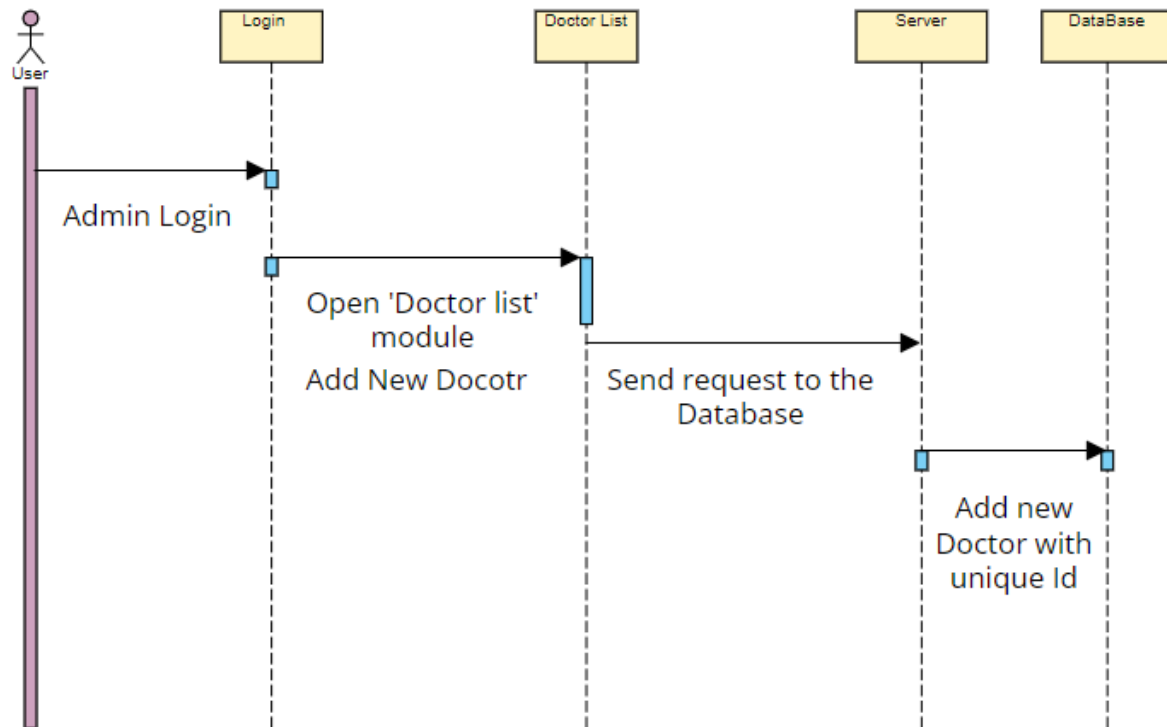


Figure 50 SD- Add list of Doctors

3.3.23 SD-23: Delete List of Doctors

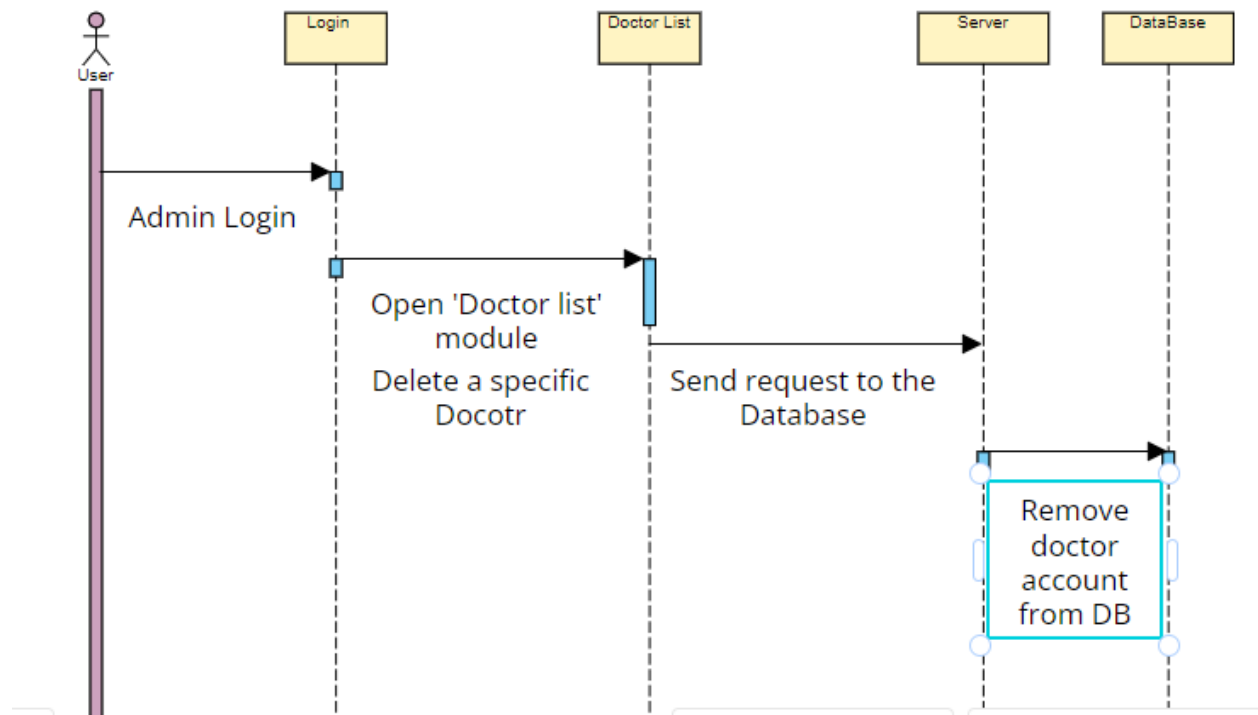


Figure 51 SD- Delete list of doctors

3.3.24 SD-24: Logout

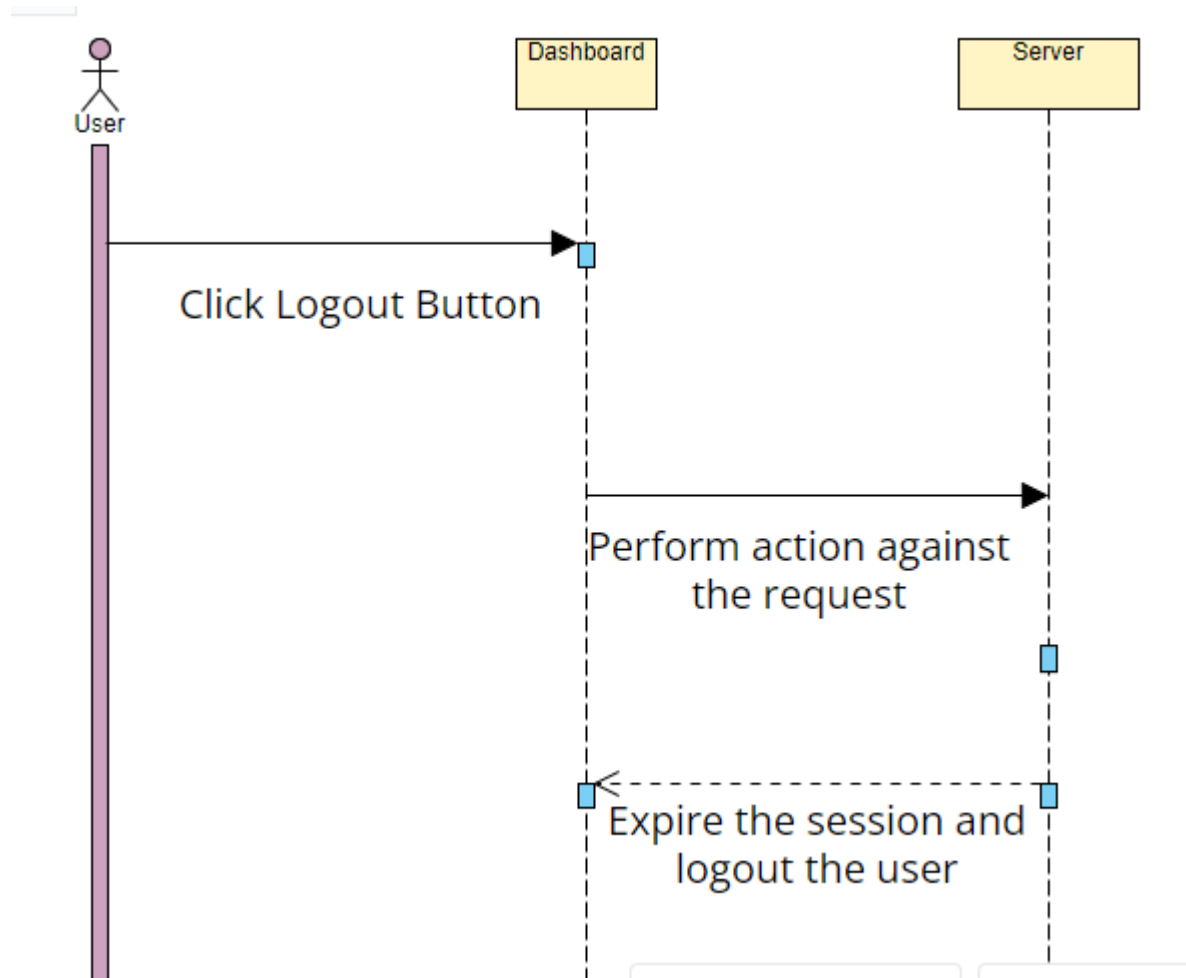


Figure 52 SD- Logout

3.4 Software Architecture

In our project we are using Client Server software architecture. It is a common design frame work where client server heaps are separated. In the client server all the assets are controlled by the servers. The server gets different demonstration to share assets to the client when stated. Client and server are on the same forum. The server is firm and adaptable to answer the client. Thus, this architecture is a service concerned with the client administration that implies won't be interrupted. Client-server architecture overwhelms arranging traffic by responding to client requests as opposed to moving documents entirely.

The client PCs will make a connection to allow the client PC to request server administration and talk to the results that the server returns. Servers believe that tasks from Users will appear and then bring them back.

3.4.1 Types of Client-Server Architecture:

There are three basics types of client-Server Architecture

3.4.1.1 1-Tier Architecture:

In this type of architecture, the data is directly provided to user and user can easily use the database through the computer. Any type of changes made by user will directly reflect to the database. This architecture is used when the programmer is on one-to-one communication with the database to make its response fast. It is also used for the improvement of the local application.

3.4.1.2 2- Tier Architecture:

A 2-Tier Architecture in DBMS is a Database architecture where the presentation layer runs on a client (PC, Mobile, Tablet, etc.), and data is stored on a server called the second tier. Two tier architecture provides added security to the DBMS as it is not exposed to the end-user directly. Its main benefit is that it makes the communication faster.

3.4.1.3 3- Tier Architecture

A 3-tier Architecture is the one of the useful client server architectures in which the development and the main functional processes, logic, data access, data storage and users' interface is done individually as distinct segments. This includes these three components:

1. Presentation layer (your PC, Tablet, Mobile, etc.)
2. Application layer (server)
3. Database Server

3.5 Class Diagram

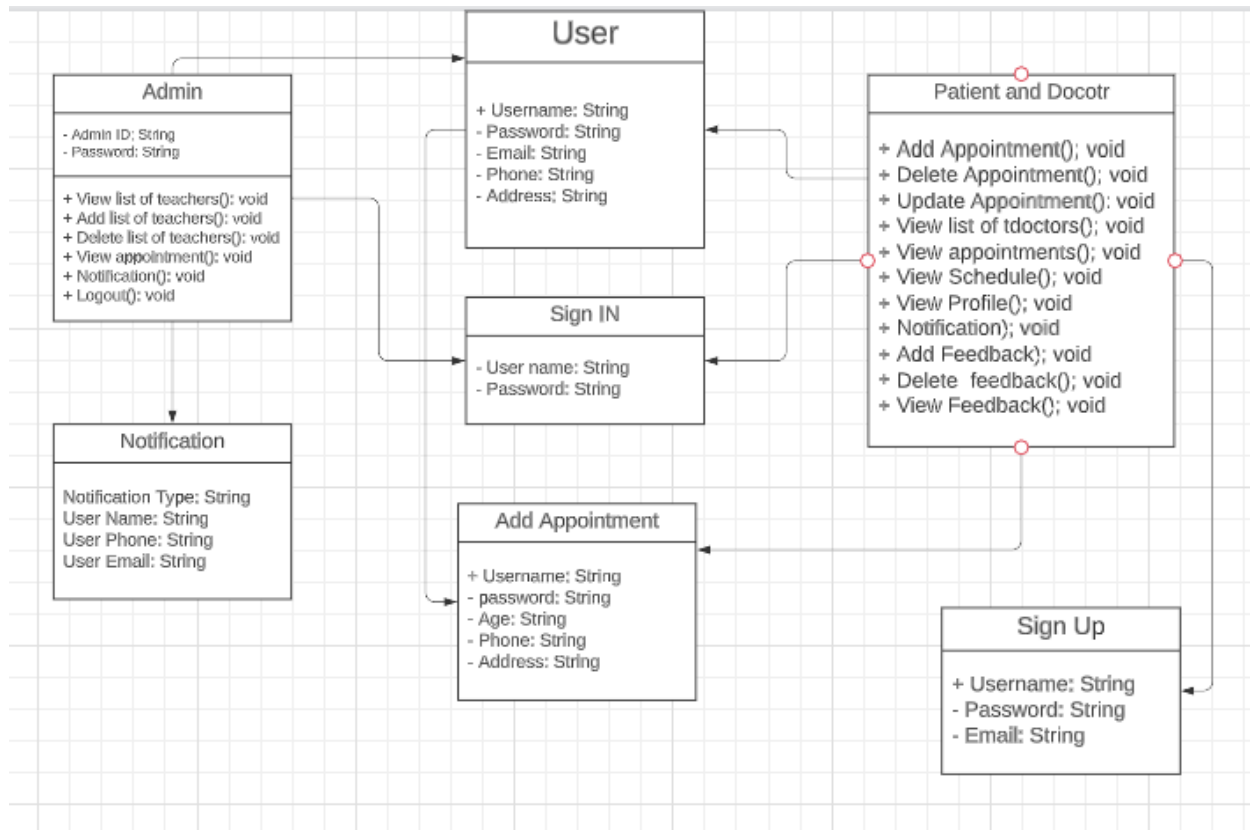


Figure 53 Class Diagram

3.6 Database Diagram

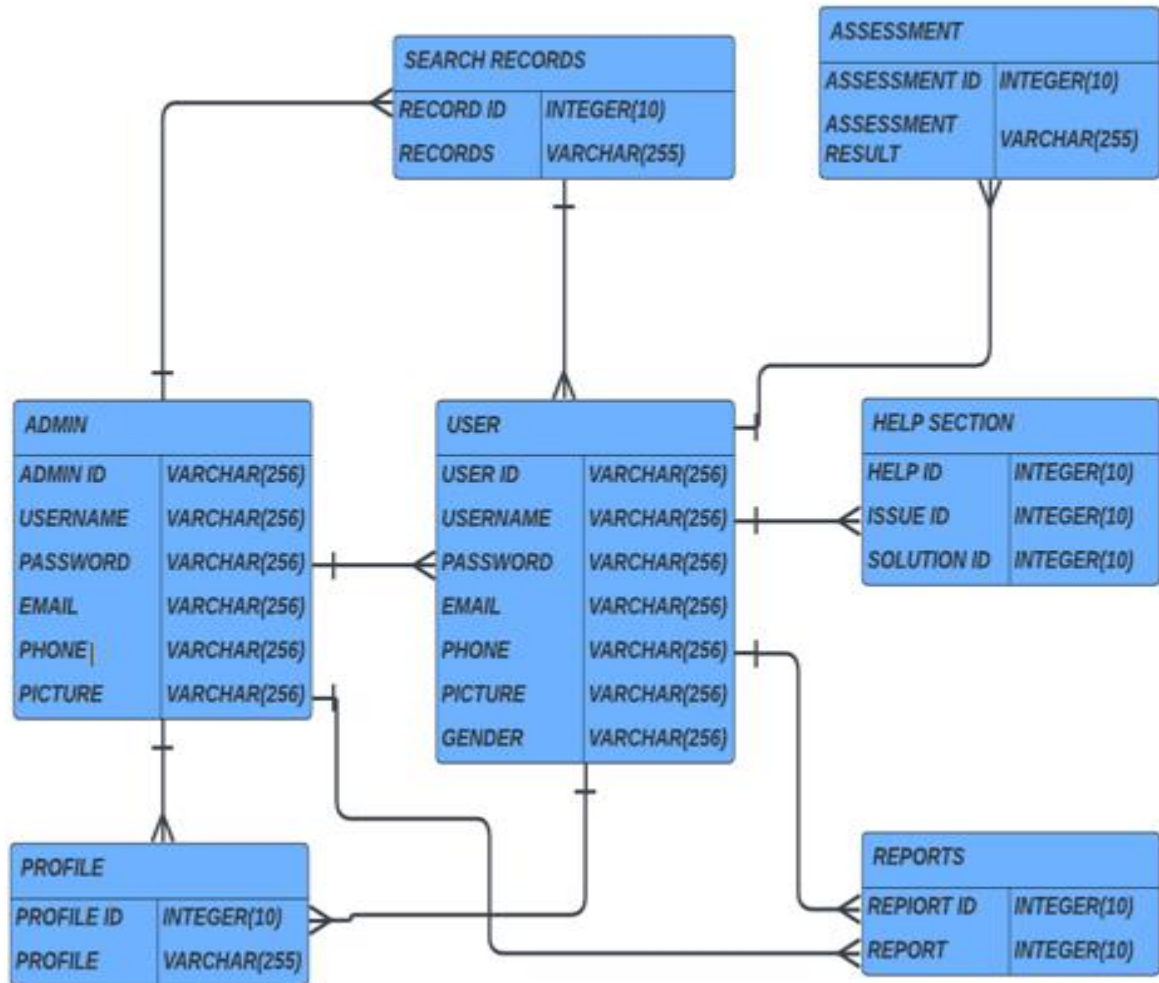


Figure 54 Database Diagram

3.7 Network Diagram(Gantt Chart)



Figure 55Network Diagram(Gantt Chart)

3.8 Collaboration Diagram

3.8.1 Sign-Up:

The user will enter the sign in credentials in the sign-up page. Those credentials will be stored and verified in the database and the user will be signed and directed to the homepage.

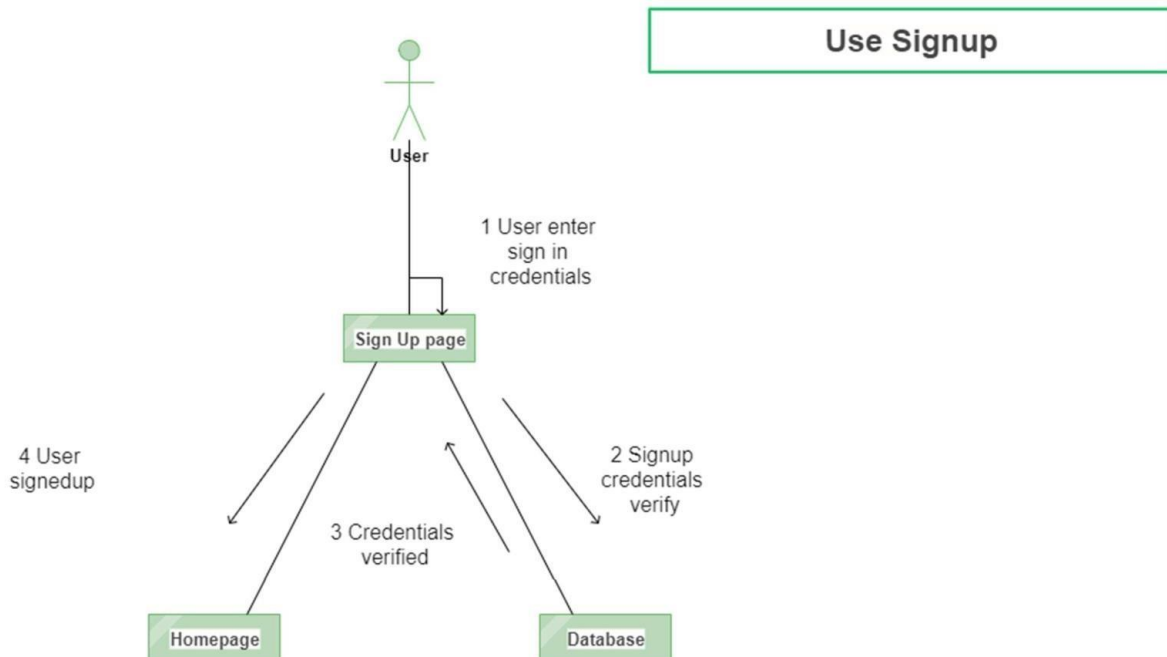


Figure 56 CD- Sign Up

3.8.2 Email Verification:

User go to the sign up page and create the account, after providing all credential when user click on the sign up button, System will send the verification email.

User can only create if Email verification done successfully.

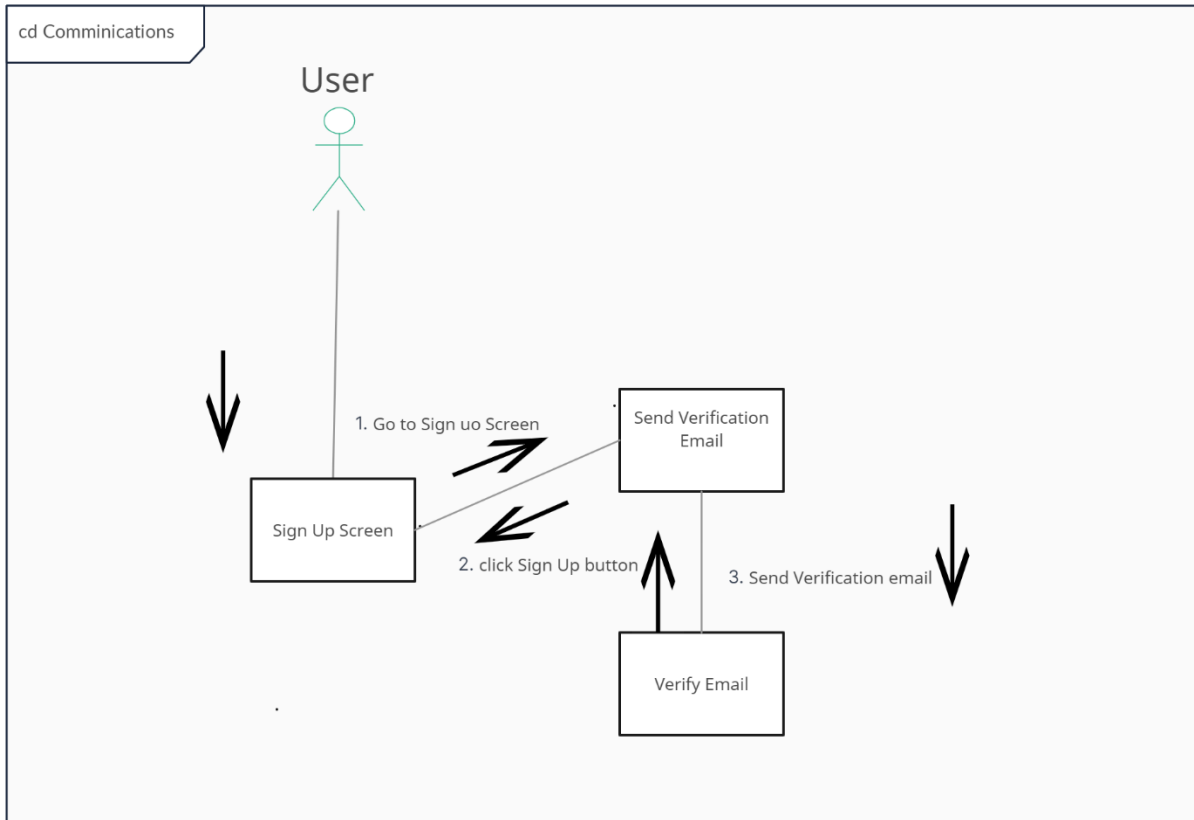


Figure 57 CD- Email Verification

3.8.3 Sign In:

User will enter login credentials and those credentials will be verified through the database. Once the credentials are verified the user will be logged in and directed to the dashboard.

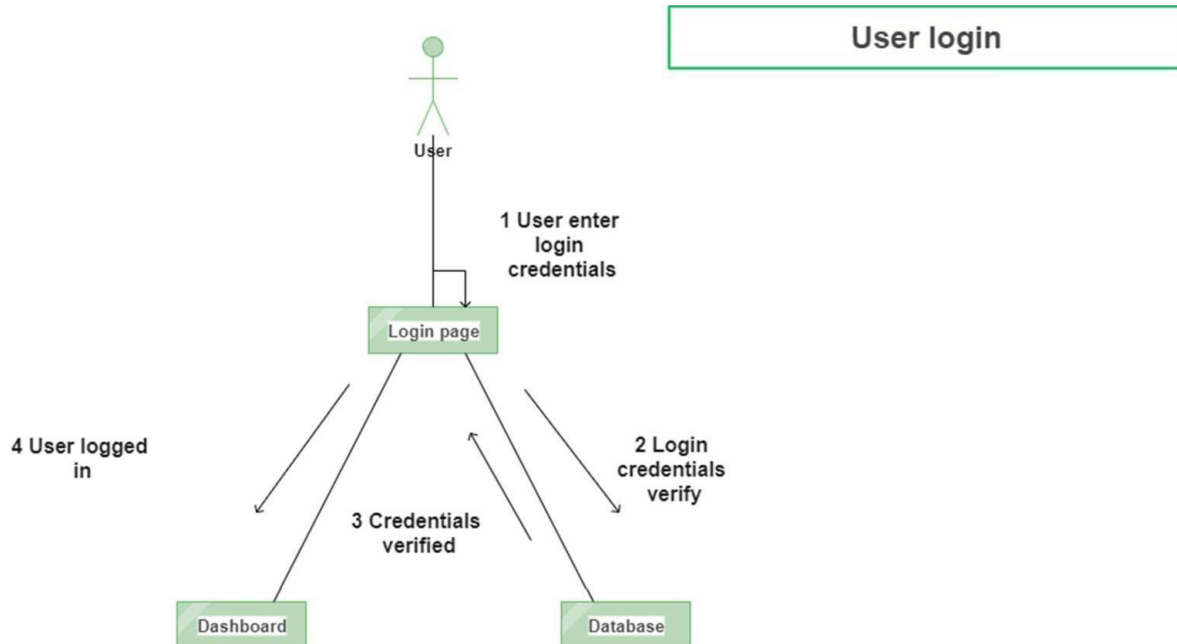


Figure 58 CD- Sign In

3.8.4 View Doctor List:

The user will login to the portal and select the list of teachers from the dashboard. The request will be sent to the server and after the reply from server the list of doctors will be displayed.

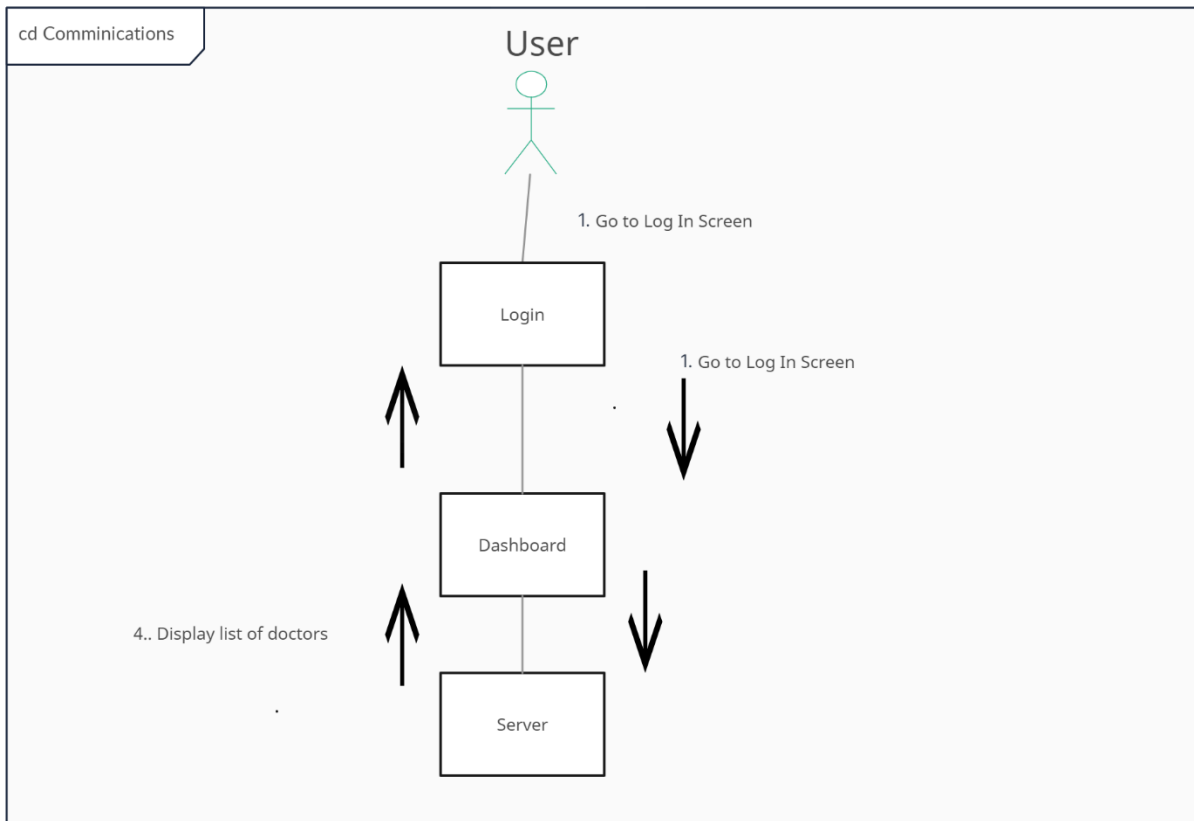


Figure 59 CD- View Doctor List

3.8.5 Add Feedback:

First user will select add feedback option from the homepage then enter the feedback. The entered feedback data will be saved in the database and feedback will be added to the homepage.

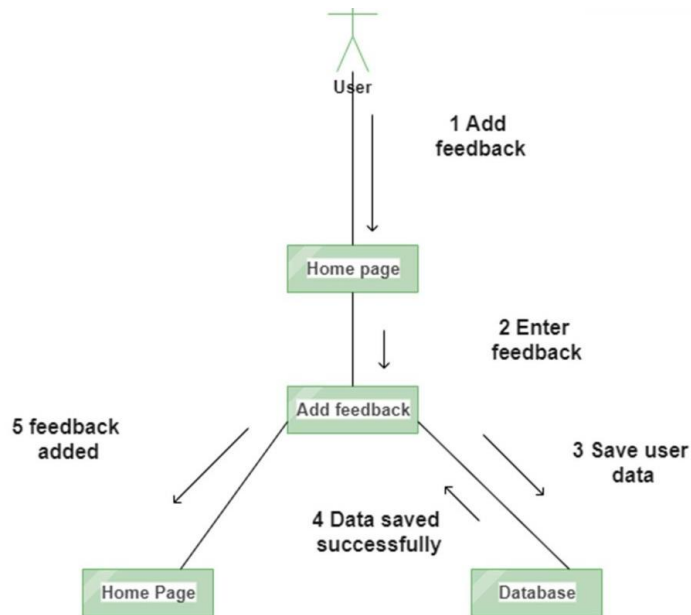
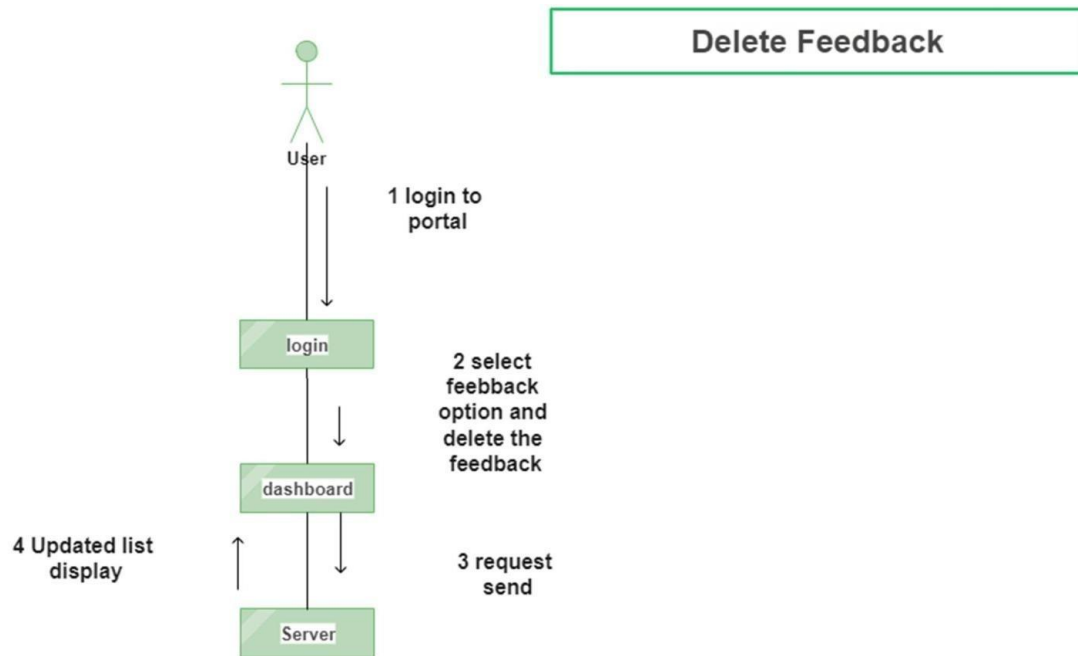


Figure 60 Add Feedback

Figure 61 Update Feedback

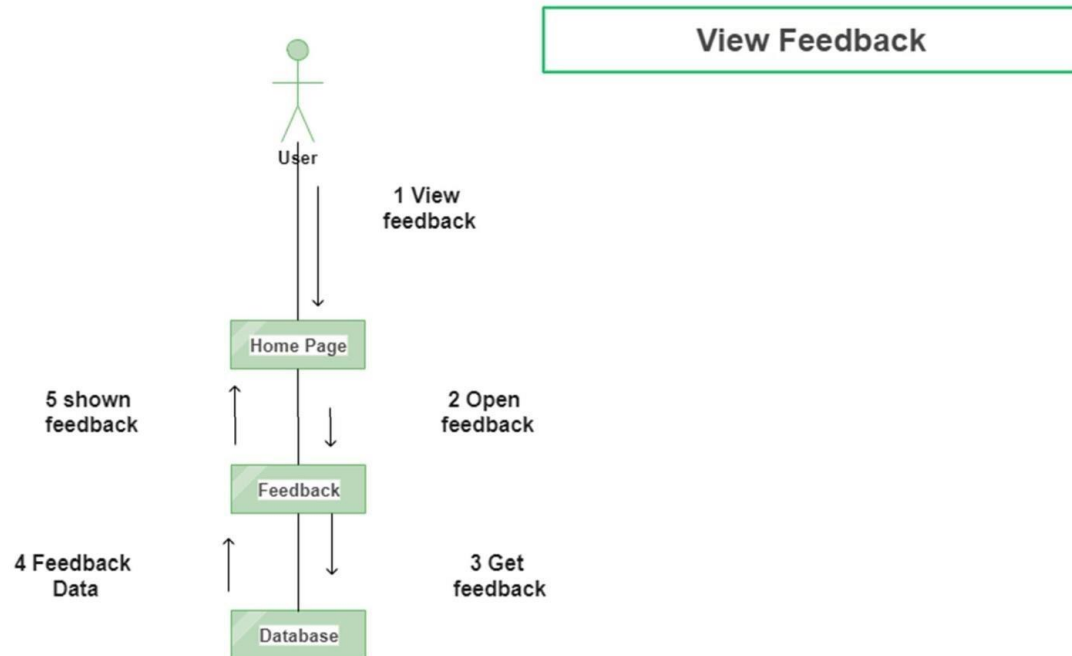
3.8.6 Delete Feedback:

The user will login first and then from the dashboard user will select the feedback option and then delete the feedback option. The request will be sent to server and the updated feedback list will be displayed.



3.8.7 View Feedback:

The user will select view feedback option from the homepage and then open the feedback. The feedback data will be fetched from the database and then the feedback will be shown.



3.8.8 Appointment Booking:

User will first select appointments and then user will be directed to add appointment page. The added appointment will be verified and after successful verification the appointment will be added.

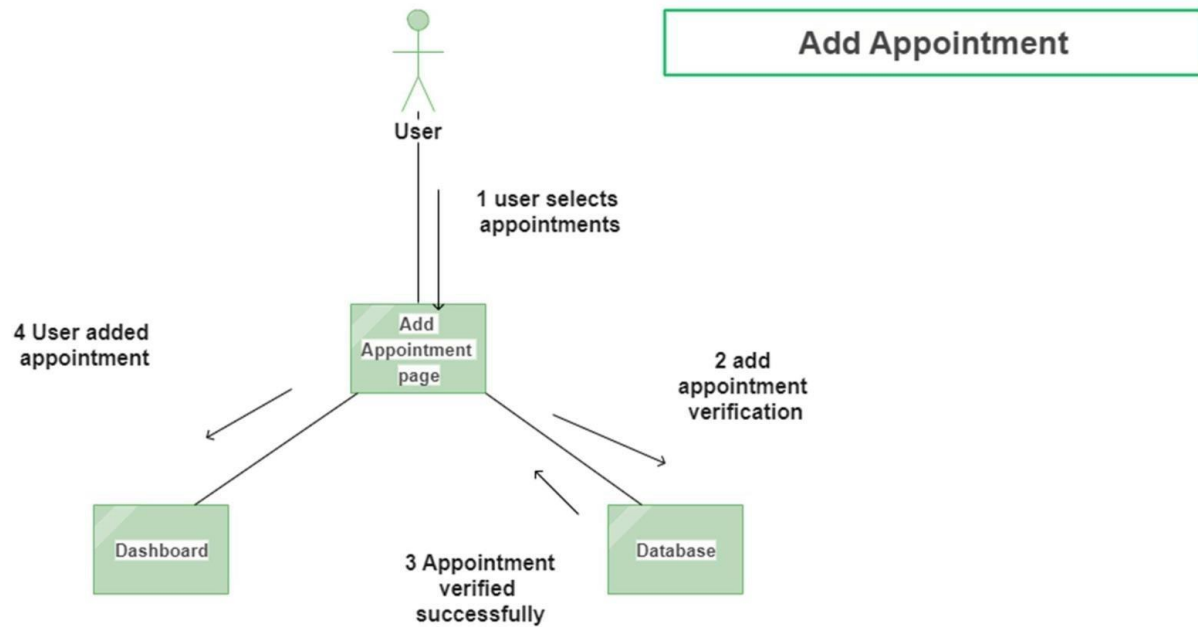


Figure 62 CD- Add Appointment

3.8.9 Delete Appointment:

The user will first login to portal and from the dashboard my appointments option user will select delete appointment. The request will be sent to server and after that the appointment will be deleted from the database. After deletion of appointment from the database the updated list will be displayed.

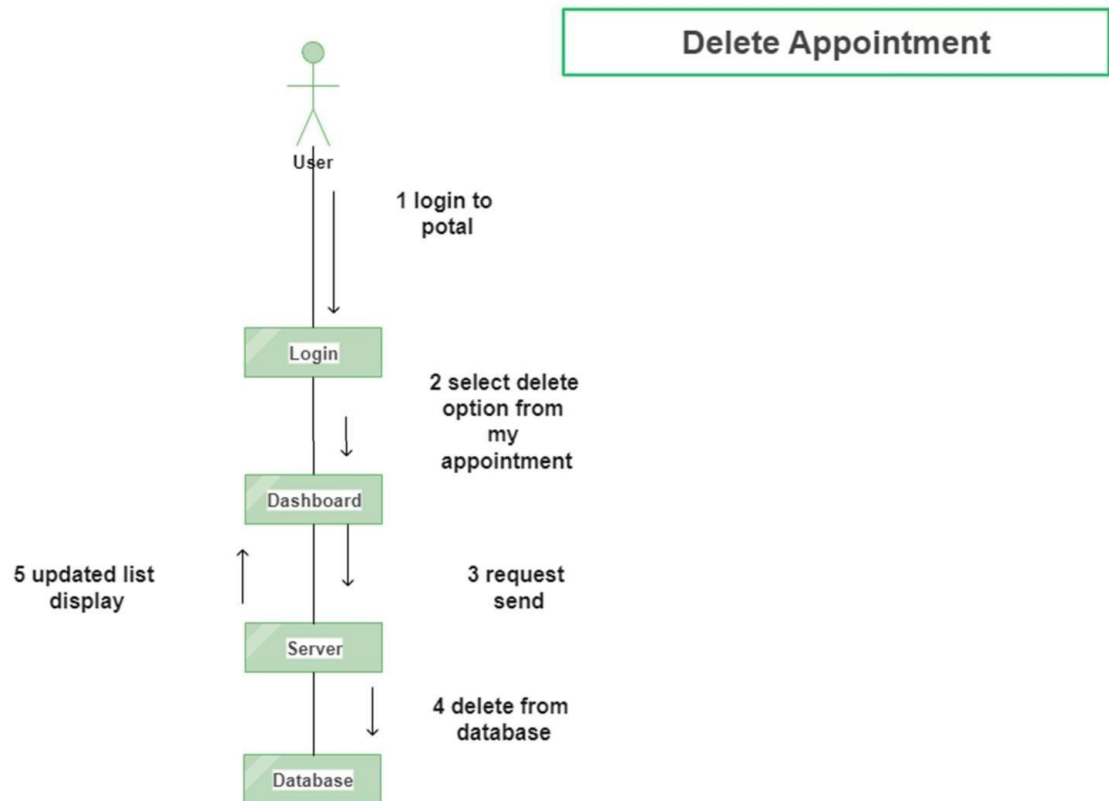


Figure 63 CD- Cancel Appointment

3.8.10 View Appointment:

The user will login first and from the dashboard select my appointment option. The request to view appointment will be sent to server and after the data is fetched from the server the appointment will be displayed.

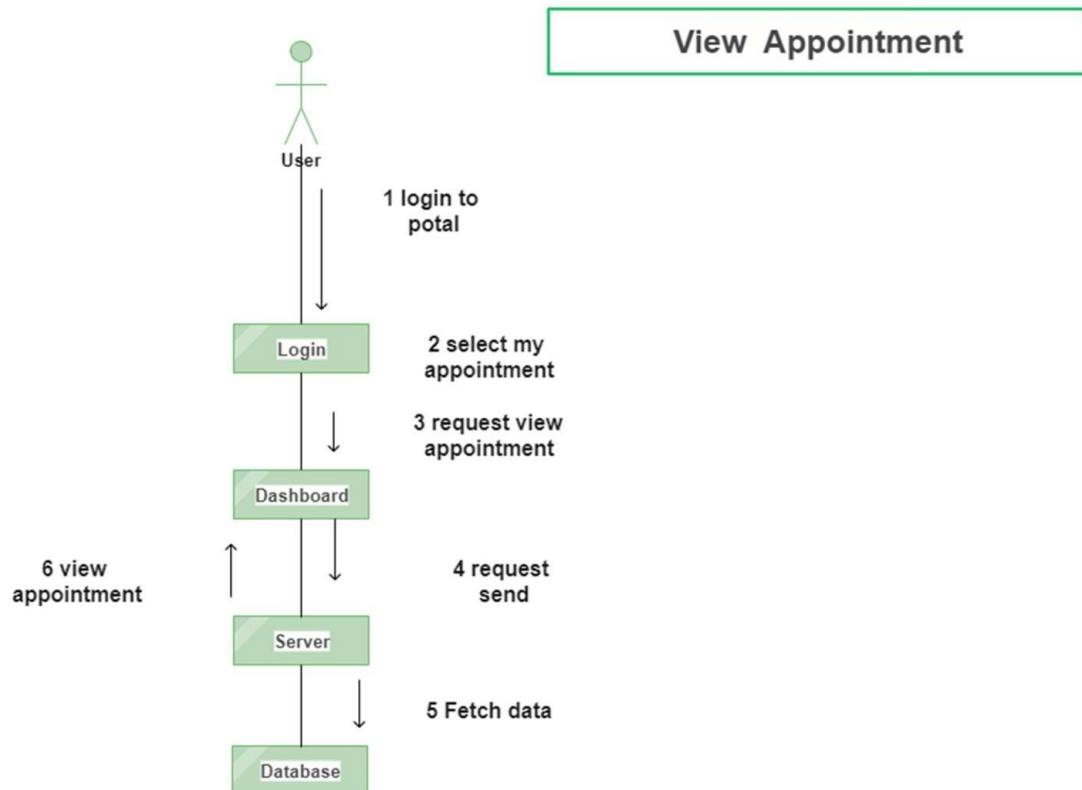


Figure 64 CD- View Appointment

3.8.11 View Profile:

The user will select open profile option from the homepage and then the system will fetch profile data from database. After this the profile will be displayed to user.

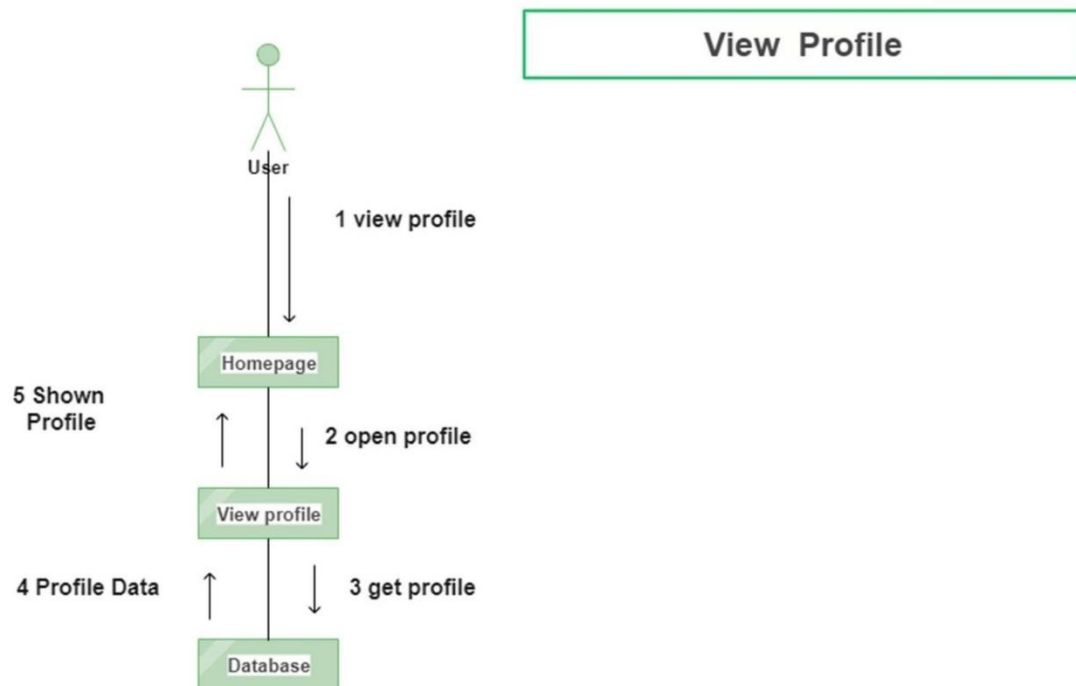


Figure 65 CD- View Profile

3.8.12 Add doctor list:

The admin will first login to the portal and from the dashboard admin will select list of doctors from service option. The request will be sent to server and the doctor will be added to the database.

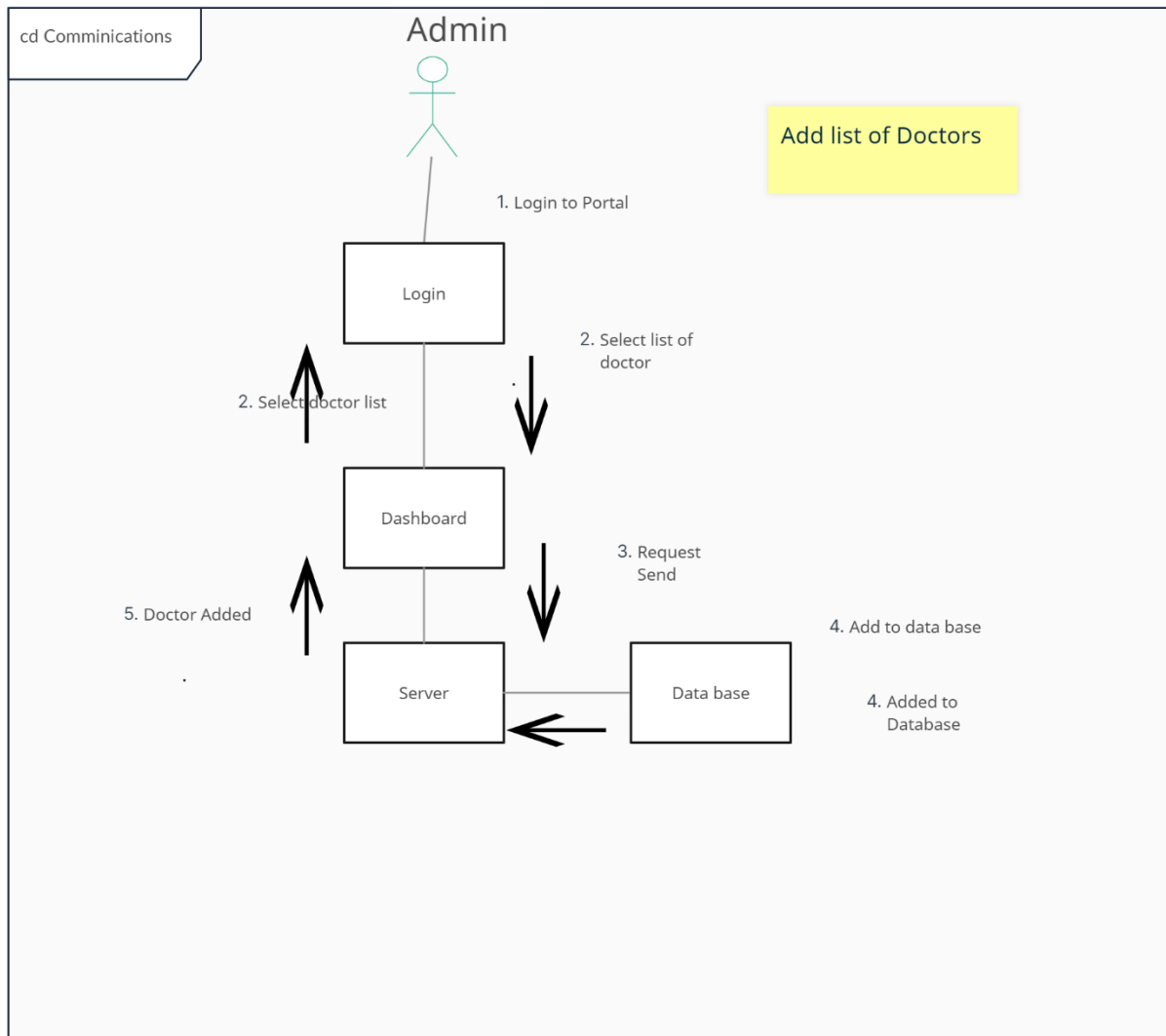


Figure 66 CD- Add Doctor List

3.8.13 Update doctor List:

The admin will first login to the portal and will be directed to the dashboard from where the admin will select list of doctors from service option and make the update. The request will be sent server and the database will be updated. After this the updated list will be displayed.

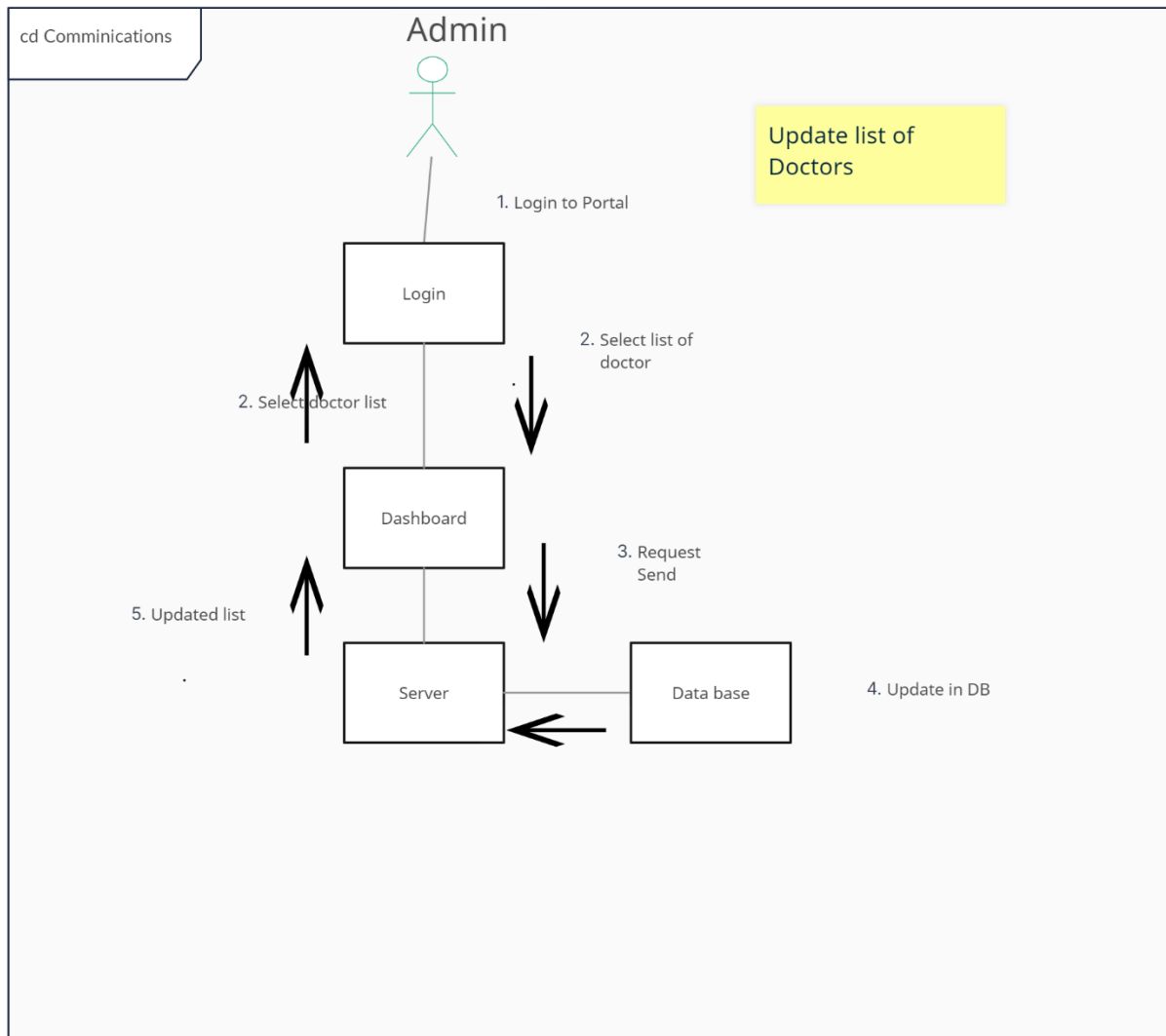


Figure 67 CD- Update List of Doctor

3.8.14 Delete List of Doctor:

The admin will first login to the portal and will be directed to the dashboard from where admin will select list of doctors from service option which is to be deleted. After this the request will be sent to server and the doctor will be deleted from the database. The updated list will be displayed.

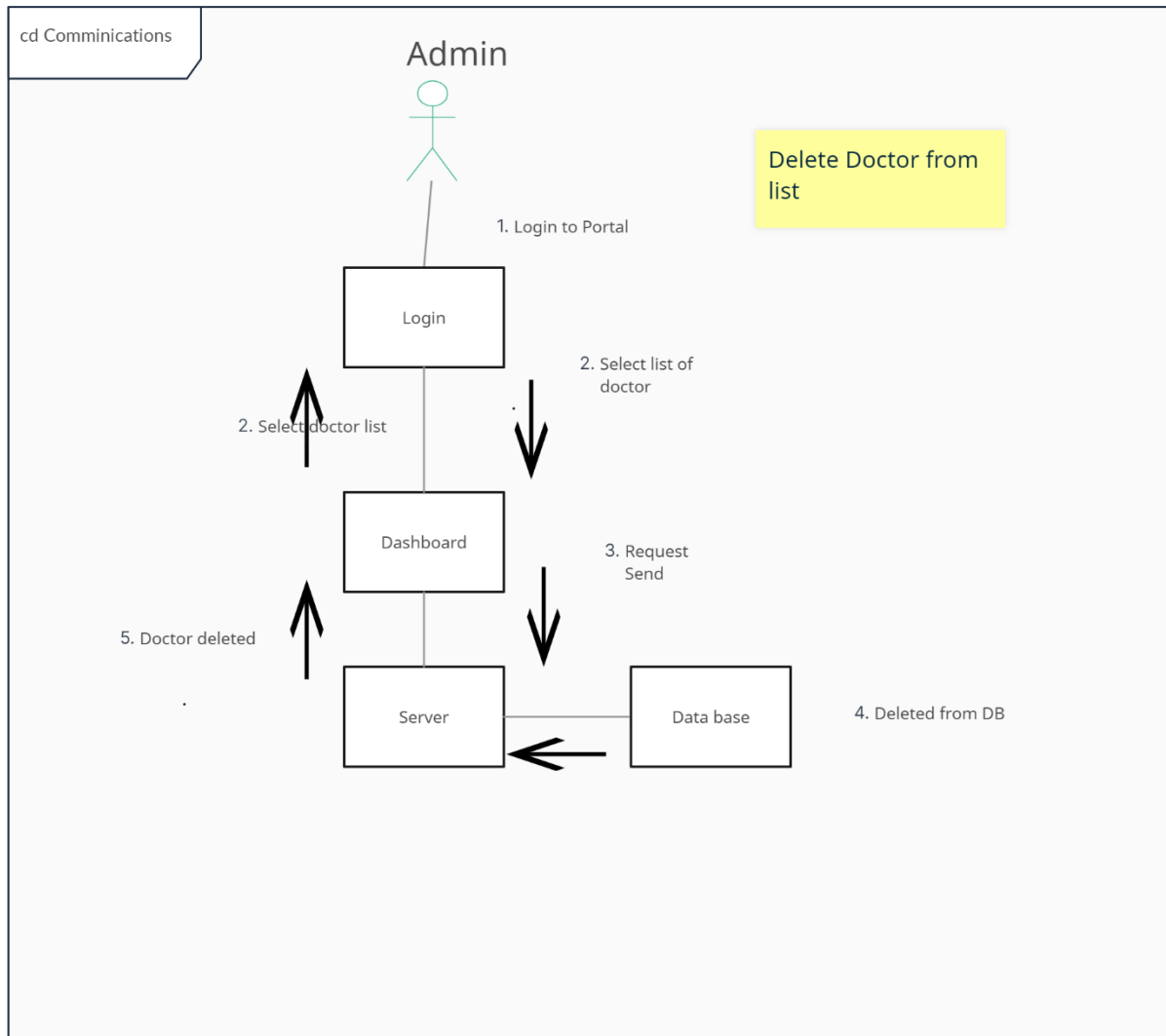


Figure 68 CD- Delete List of Doctor

3.8.15 Logout:

The user will login to the portal and from the dashboard user will select the logout option. The logout request will be sent to server and then the session will be expired and user will return to login page.

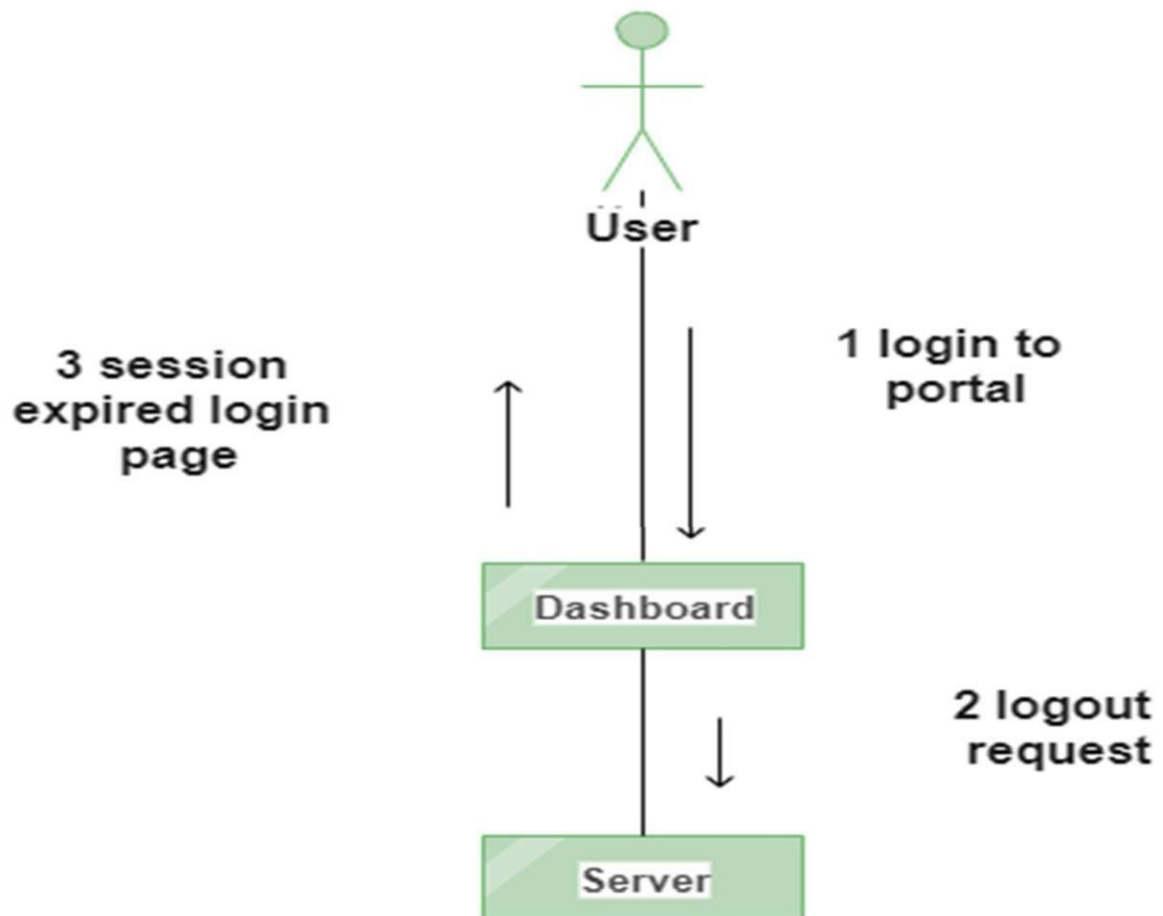


Figure 69 CD- Logout

CHAPTER 4

SYSTEM TESTING

4 System Testing

4.1 Test Cases

TC01: Sign Up

Test case#1	Test Case Name: Sign Up
Designed By: Rameez Ashraf	Design date: 01/06/2023
Executed By:	Execution date:
Short description: User can create account by submit the credentials	

Pre-Condition:

User must not be already registered

Step	Action	Expected system response	Pass/Fail	Comment
1.	The User visit sign up page	The system displays form.		
2.	The User enter details Name, CNIC, Contact Number, Email, Address and Password.	The system validates the information entered.		
3.	The User submit account information.	The system send verification Email.		
Post condition: User received Verification Email.				

Table 62: TC-01: Sign Up

TC02: Email Verification

Test case#2	Test Case Name: Email Verification
Designed By: Rameez Ashraf	Design date: 10/06/2023
Executed By: Dilawar Hussain	Execution date:
Short description: Send Verification Email on submit sign Up form	

Pre-Condition:
User provides all compulsory credentials.

Step	Action	Expected system response	Pass/Fail	Comment
1.	User click on sign up link.	The system displays Sign Up form/page.		
2.	User enter credential and press sign up button	System send verification Email		
3.	User Verify the email	System create account against that email.		
Post condition: User received Verification Email.				

Table 63: TC-02: Email Verification

TC03: Sign In

Test case#3	Test Case Name: Sign In
Designed By: Rameez Ashraf	Design date: 1/06/2023
Executed By:	Execution date:
Short description: To test that User can login after submitting details.	

Pre-Condition: User should be registered.

Step	Action	Expected system response	Pass/Fail	Comment
1.	The User visit Log In page	The system display login form.		
2.	The User enter details Email Address and Password.	The system will check the entered data.		
3.	The User submit account information.	Check data in data base		
Post condition: User Log in to the application.				

Table 64: TC-03: Sign In

TC04: Password Reset

Test case#4		Test Case Name: Password Reset		
Designed By: Rameez Ashraf		Design date: 01/06/2023		
Executed By: Dilawar Hussain		Execution date:		
Short description: To test password reset functionality.				
Pre-Condition: User must have old password.				
Step	Action	Expected system response	Pass/Fail	Comment
1.	User visit login page and	The system display login form.		
2.	Click on forget password	Open new page of password reset.		
3.	Set new password	Save new password in database.		
Post condition: Password reset successfully.				

Table 65: TC-04: Password Reset

TC05: View Doctor List

Test case#5	Test Case Name: View Doctor List
Designed By: Dilawar Hussain	Design date: 1/06/2023
Executed By:	Execution date:
Short description: To test that User can login after submitting details.	

Pre-Condition: Number of doctors should be exits.
--

Step	Action	Expected system response	Pass/Fail	Comment
1.	User click on the doctor list.	The system must show list of doctors.		
2.	The User click on a doctor account.	The system will open the doctor profile.		
3.	The User can view the doctors details and schedule.	System will show doctor profile details.		
Post condition: User viewed the doctor.				

Table 66: TC-05: View Doctor List

TC 06: Apply for Doctor

Test case#6	Test Case Name: Apply for Doctor
Designed By: Rameez Ashraf	Design date: 01/06/2023
Executed By:	Execution date:
Short description: Test case for apply doctor profile	

Pre-Condition:
User must be login and have required documents or licence.

Step	Action	Expected system response	Pass/Fail	Comment
1.	The user click on 'Apply for doctor'.	System will open Apply form.		
2.	User fill all the required filed and press "Apply" button.	The system send the request to Admin.		
3.	Admin approve or disprove the request.	System send result status back to user.		
4.	User role changed to 'Doctor'.	System change role in database,		
Post condition: User role changed to doctor in database.				

Table 67: TC-06: Apply for Doctor

TC 08: View Profile

Test case#8	Test Case Name: View Profile
Designed By: Rameez Ashraf	Design date: 01/06/2023
Executed By:	Execution date:
Short description: User and Admin can view the profile.	

Pre-Condition:
User must be registered

Step	Action	Expected system response	Pass/Fail	Comment
1.	The user click on Profile button.	System must display profile.		
2.	The User can see all his/her information in the profile like name, city, address etc.	The system displays the information		
Post condition: User will done with profile view.				

Table 68: TC-08: View Profile

TC 09: Set Schedule

Test case#9	Test Case Name: Set Schedule
Designed By: Rameez Ashraf	Design date: 01/06/2023
Executed By:	Execution date:
Short description: Test Doctor set the appointment schedule .	

Pre-Condition: Must have role of Doctor

Step	Action	Expected system response	Pass/Fail	Comment
1.	Doctor Login and click on Schedule	System must display Schedule.		
2.	Doctor can set the schedule according own time table.	The system must set the schedule.		
Post condition: Doctor will set the schedule.				

Table 69: TC-09: Set Schedule

TC 11: Appointment Booking

Test case# 11	Test Case Name: Appointment Booking
Designed By: Rameez Ashraf	Design date: 01/06/2023
Executed By:	Execution date:
Short description: Test to book appointment.	

Pre-Condition: Slot should be free.

Step	Action	Expected system response	Pass/Fail	Comment
1.	User click on book appointment.	System should send the notification to doctor.		
2.	Doctor view the appointment request and do response.	System should book the appointment and send response to doctor.		
Post condition: Appointment will be added and slot will become unavailable for other user.				

Table 70: TC-11: Appointment Booking

TC 12: Appointment Track

Test case#12	Test Case Name: Appointment Track
Designed By: Dilawar Hussain	Design date: 01/06/2023
Executed By:	Execution date:
Short description: Test to book appointment.	

Pre-Condition: User already book the appointment.

Step	Action	Expected system response	Pass/Fail	Comment
1.	User click on appointments.	System should show all appointments done by user.		
2.	User click on clear history.	System should clear the appointment history.		
Post condition: Appointment history will be removed.				

Table 71: TC-12: Appointment Track

TC 13: Clear Appointment

Test case#12	Test Case Name: Clear Appointment
Designed By: Dilawar Hussain	Design date: 01/06/2023
Executed By:	Execution date:
Short description: Test to Clear past appointment.	

Pre-Condition: User already book the appointment.

Step	Action	Expected system response	Pass/Fail	Comment
1.	User click on appointments.	System should show all appointments done by user.		
2.	User click on clear history.	System should clear the appointment history.		
Post condition: Appointment history will be removed.				

Table 72: TC-13: Clear Appointment History

TC 14: Cancel Appointment

Test case#14	Test Case Name: Cancel Appointment
Designed By: Dilawar Hussain	Design date: 01/06/2023
Executed By:	Execution date:
Short description: Test case to cancel the appointment successfully.	

Pre-Condition:
There should be at least one upcoming appointment.

Step	Action	Expected system response	Pass/Fail	Comment
1.	User click on appointments.	System should show all appointments done by user.		
2.	User click on cancel appointment.	Cancel the appointment and free the booking slot.		
Post condition: Appointment will be canceled.				

Table 73: TC-14: Cannel Appointment

TC 15: Update Appointment

Test case#15	Test Case Name: Update Appointment
Designed By: Rameez Ashraf	Design date: 01/06/2023
Executed By:	Execution date:
Short description: Test case to update the appointment successfully.	

Pre-Condition:
There should be at least one upcoming appointment.

Step	Action	Expected system response	Pass/Fail	Comment
1.	User click on appointments.	System should show all appointments done by user.		
2.	User click on update appointment.	Update the the appointment and free the previous booking slot and reserved mew slot.		
Post condition: Appointment will be update.				

Table 74: TC-15: Update Appointment

TC 16: Notification System

Test case#16	Test Case Name: Notification System
Designed By: Rameez Ashraf	Design date: 01/06/2023
Executed By:	Execution date:
Short description: Test case to check the notification module working.	

Pre-Condition:
User should be login to access the notification.

Step	Action	Expected system response	Pass/Fail	Comment
1.	The User presses notification link/icon.	The system must display notification page.		
2.	User can mark notification as 'read all'.	System will show zero notification.		
Post condition: Notification readied and marked as readied.				

Table 75: TC-16: Notification

TC 18: View Candidate Profile

Test case#18	Test Case Name: View Candidate Profile
Designed By: Dilawar Hussain	Design date: 01/06/2023
Executed By:	Execution date:
Short description: Test case to view the candidate profile by Admin.	

Pre-Condition: Admin must submit the application form.
--

Step	Action	Expected system response	Pass/Fail	Comment
1.	User click on 'Apply for Doctor'.	System should open the apply form		
2.	Fill and submit the form.	Form submit and send notification to Admin.		
3.	Admin prove or decline the application.	System send the result to candidate.		
4.	Successfully role changed to doctor.	System change role in DB.		
Post condition: May user role changed to doctor.				

Table 76: TC-18: View Candidate Profile

TC 20: Add Feedback

Test case#20	Test Case Name: Add Feedback
Designed By: Rameez Ashraf	Design date: 01/06/2023
Executed By:	Execution date:
Short description: Test that User can add the feedback.	

Pre-Condition:
User should be registered and may done with appointment.

Step	Action	Expected system response	Pass/Fail	Comment
1.	The User press feedback button.	System should open the apply form		
2.	Fill and submit the form.	The system displays the feedback page.		
3.	The User enter the feedback through comments or by rating through stars	The system displays the feedback.		
Post condition: User feedback will be added and viewable.				

Table 77: TC-20: Add Feedback

TC 21: View Feedback

Test case#21	Test Case Name: View Feedback
Designed By: Dilawar Hussain	Design date: 01/06/2023
Executed By:	Execution date:
Short description: Test that User can view feedback.	

Pre-Condition:
User should be Log In.

Step	Action	Expected system response	Pass/Fail	Comment
1.	The User press the view feedback button.	The system displays the feedback page.		
2.	The User enter the specific teacher to view feedback of that teacher.	The system displays the teacher.		
3.	The User enter the feedback button.	The system displays the feedback of that doctor.		
Post condition: User viewed the feedback.				

Table 78 TC-21: View Feedback

TC 22: Add List of Doctor

Test case#22	Test Case Name: Add list of doctor
Designed By: Dilawar Hussain	Design date: 01/06/2023
Executed By:	Execution date:
Short description: Test that Admin can add the new doctor.	

Pre-Condition: Admin should be login.

Step	Action	Expected system response	Pass/Fail	Comment
1.	Admin Email and password for login.	System must login the Admin.		
2.	Admin add the doctor in doctor list.	System should add new doctor account in system or database.		
Post condition: New doctor account will add by Admin.				

Table 79 TC-22: Add Doctor by Admin

TC 23: Delete List of Doctors

Test case#23	Test Case Name: Delete list of doctor
Designed By: Dilawar Hussain	Design date: 01/06/2023
Executed By:	Execution date:
Short description: Test that Admin can the delete the doctor from doctor list.	

Pre-Condition: Admin should be login.

Step	Action	Expected system response	Pass/Fail	Comment
1.	Admin enter Email and password for login.	System must login the Admin.		
2.	Admin delete the doctor.	System should delete doctor from database.		
Post condition: Doctor account deleted.				

Table 80 TC-23: Delete Doctor

TC 24: Logout

Test case#24	Test Case Name: Logout
Designed By: Rameez Ashraf	Design date: 01/06/2023
Executed By:	Execution date:
Short description: Test that User can Logout.	

Pre-Condition: User should be Login.
--

Step	Action	Expected system response	Pass/Fail	Comment
1.	User presses on the Logout button.	The system must display a Confirmation message.		
2.	User presses on the Ok option and Logout the system.	The system must redirect the user to Log-In page.		
Post condition: User is Logout.				

Table 81 TC-24: Logout

4.2 Unit Testing:

Unit testing is performed by the Developers. The Developer performs tests on all the modules and components of the System. The advantages of the unit testing are that we can detect errors or bugs at an early stage and calculate the minimal cost to fix the errors or the bugs.

The modules to be tested are:

- Patient
- Doctor
- Admin
- Login
- Appointments
- Doctor List
- Referral System
- Health Remedies(optional)
- Logout

4.3 Integration Testing:

The phase that follows unit testing is the Integration testing. The main purpose of the integration testing is to check and test the integration between all the modules and components of the System and see how they work together. In integration testing the whole functionality of the System will be tested.

The modules to be tested in Integration testing are:

- User Registration and Authentication
- Appointment Schedule
- Doctor Profile Management
- Search Functionality
- Payment gateway integration(In case of online payment)
- Notification System
- Feedback Module

4.4 Acceptance Testing:

The main purpose of Acceptance testing is to provide the features of the System at one go. After testing, the different users will be provided with the System and will be asked to use it, after using the System, the users will be asked for their feedback. Audience feedback is very important for addressing the issues of the users and make changes if necessary for good user experience and for deployment of the System. After performing acceptance testing the System will be launched if the reviews of the audience are positive.

CHAPTER 5

CONCLUSION

5 Conclusion

5.1 Problem faced and lesson learned

We faced different problems while writing this report and finalizing requirements for our project. The main problem we faced was during requirement gathering and adding functionalities in our product i.e., which functionality will be beneficial and attractive for the stakeholders and will also be practical. Also, those functionalities which will make our product different and better from our competitors.

We also faced difficulty while using different online tools like lucid chart to make use case diagrams and Figma for making prototypes for the satisfaction of stakeholders. For learning how to use these tools we had to watch different tutorials as well as to seek help from our supervisor and teachers.

At last, from this project, we learned how to work in a team and how to gather requirements which will satisfy all stakeholders and these things will also help us a lot in our future endeavour's and professional life.

5.2 Summary of Project

Our platform which is a web application, this application is an online application which simplifies the process of scheduling appointments with doctors. It offers features such as user registration, doctor search, appointment schedule, user and doctor profile management.

User get the list of doctors exist in the application. User can search the specific doctor on base of different kind of filters like (location, Speciality, Appointment fee etc.)

Doctor can handle the schedule efficiently, system not allow two appointment in the same slot. To facilitate the appointment process, we have chat box functionality that user can communicate with doctor to ask some question before appointment booking.

The application aims to provide a user-friendly interface, streamline appointment booking.

5.3 Future Work

We apply the incremental model because we can enhance our project in future as we get the more functionalities. In future may we can add some functionalities like

- 1) May add more functionalities for the ease and benefit of stakeholders.
- 2) Add Artificial Intelligence in our application to enhance the services
- 3) May expand the services of our application and launch it in different markets.

There may be more other functionalities that are not defined yet.

6 References

- [1] “OBERLO”. [Online]. Available: <https://www.oberlo.com/blog/referral-marketing>
- [2] “Practo”. [Online]. Available: <https://www.practo.com/>
- [3] “Healthgrades”. [Online]. Available: <https://www.healthgrades.com/>
- [4] “Marham”. [Online]. Available: <https://www.marham.pk/>
- [5] “MeriSehat”. [Online]. Available: <https://merisehat.pk/>
- [6] “Oladoc”. [Online]. Available: <https://oladoc.com/>

Annexure:

Turnitin Originality Report

- Processed on: 09-Jun-2023 15:20 PKT
- ID: 2112415703
- Word Count: 13988
- Submitted: 1

6.1 Online Doctor appointment By R Ashraf

Similarity Index

16%

Similarity by Source

Internet Sources:

7%

Publications:

0%

Student Papers:

15%

3% match (student papers from 08-Jan-2023)
[Submitted to Higher Education Commission Pakistan on 2023-01-08](#)
3% match (student papers from 16-Jun-2022)
[Submitted to Higher Education Commission Pakistan on 2022-06-16](#)
1% match (student papers from 08-Jun-2023)
[Submitted to Higher Education Commission Pakistan on 2023-06-08](#)
1% match (student papers from 15-May-2018)
[Submitted to Higher Education Commission Pakistan on 2018-05-15](#)
< 1% match (student papers from 10-Jun-2021)
[Submitted to Higher Education Commission Pakistan on 2021-06-10](#)
< 1% match (student papers from 08-Jun-2023)
[Submitted to Higher Education Commission Pakistan on 2023-06-08](#)
< 1% match (student papers from 15-Jun-2022)
[Submitted to Higher Education Commission Pakistan on 2022-06-15](#)
< 1% match (student papers from 16-Feb-2018)
[Submitted to Higher Education Commission Pakistan on 2018-02-16](#)
< 1% match (student papers from 29-Dec-2021)
[Submitted to Higher Education Commission Pakistan on 2021-12-29](#)
< 1% match (student papers from 18-Jun-2022)
[Submitted to Higher Education Commission Pakistan on 2022-06-18](#)
< 1% match (student papers from 05-Aug-2020)

[Submitted to Higher Education Commission Pakistan on 2020-08-05](#)
< 1% match (student papers from 28-Jun-2022)

[Submitted to Higher Education Commission Pakistan on 2022-06-28](#)
< 1% match (student papers from 09-Jan-2018)

[Submitted to Higher Education Commission Pakistan on 2018-01-09](#)
< 1% match (student papers from 19-May-2016)

[Submitted to Higher Education Commission Pakistan on 2016-05-19](#)
< 1% match (student papers from 06-Jun-2017)

[Submitted to Higher Education Commission Pakistan on 2017-06-06](#)
< 1% match (student papers from 05-Aug-2020)

[Submitted to Higher Education Commission Pakistan on 2020-08-05](#)
< 1% match (student papers from 08-Jun-2021)

[Submitted to Higher Education Commission Pakistan on 2021-06-08](#)
< 1% match (student papers from 10-Jan-2018)

[Submitted to Higher Education Commission Pakistan on 2018-01-10](#)
< 1% match (student papers from 24-Jun-2022)

[Submitted to Higher Education Commission Pakistan on 2022-06-24](#)
< 1% match (student papers from 18-Jan-2023)

[Submitted to Higher Education Commission Pakistan on 2023-01-18](#)
< 1% match (student papers from 06-Jun-2023)

[Submitted to Higher Education Commission Pakistan on 2023-06-06](#)
< 1% match (student papers from 19-Dec-2016)

[Submitted to Higher Education Commission Pakistan on 2016-12-19](#)
< 1% match (student papers from 24-Dec-2020)

[Submitted to Higher Education Commission Pakistan on 2020-12-24](#)
< 1% match (student papers from 02-Jan-2023)

[Submitted to Higher Education Commission Pakistan on 2023-01-02](#)
< 1% match (student papers from 28-Dec-2015)

[Submitted to Higher Education Commission Pakistan on 2015-12-28](#)
< 1% match (student papers from 22-Dec-2016)

[Submitted to Higher Education Commission Pakistan on 2016-12-22](#)
< 1% match (Internet from 28-Mar-2023)

<https://WWW.coursehero.com/file/91561033/proposaldocx/>
< 1% match (Internet from 07-Jun-2023)

<https://www.coursehero.com/file/p3igants/If-both-is-correct-display-message-successfully-change-Test-case-for-Create-plan/>
< 1% match (Internet from 19-Dec-2021)

<https://www.coursehero.com/file/99943676/Bilaldocx/>
< 1% match (Internet from 01-Jun-2023)

<https://www.coursehero.com/file/p4ijornk/8-Admin-click-save-button-Alternative-Flow-NA-Exceptions-NA-Business-Rule-BR-01/>
< 1% match (student papers from 11-Dec-2015)

[Submitted to International University - VNUHCM on 2015-12-11](#)
< 1% match (student papers from 09-Dec-2015)

[Submitted to International University - VNUHCM on 2015-12-09](#)
< 1% match (student papers from 11-Dec-2015)

[Submitted to International University - VNUHCM on 2015-12-11](#)
< 1% match (Internet from 27-Sep-2021)

<https://www.guru99.com/dbms-architecture.html/>
< 1% match (student papers from 11-Dec-2020)

[Submitted to NCC Education on 2020-12-11](#)
< 1% match (Internet from 12-Dec-2022)

<http://repository.cuilahore.edu.pk/xmlui/bitstream/handle/123456789/3275/7503..pdf?isAllowed=y&sequence=1>
 < 1% match (Internet from 30-Dec-2022)

<http://repository.cuilahore.edu.pk/xmlui/bitstream/handle/123456789/2840/SE39%20DocPlag-%20A%20plagiarism%20Detector.pdf?sequence=1>
 < 1% match (student papers from 30-Jun-2016)

[Submitted to UNITEC Institute of Technology on 2016-06-30](#)
 < 1% match (student papers from 05-Jun-2023)

[Submitted to UNITEC Institute of Technology on 2023-06-05](#)
 < 1% match (student papers from 27-Feb-2018)

[Submitted to Asia Pacific University College of Technology and Innovation \(UCTI\) on 2018-02-27](#)
 < 1% match (student papers from 13-Jul-2013)

[Submitted to Asia Pacific University College of Technology and Innovation \(UCTI\) on 2013-07-13](#)
 < 1% match (student papers from 12-Jan-2016)

[Submitted to Universiti Teknologi Petronas on 2016-01-12](#)
 < 1% match (student papers from 09-Jun-2017)

[Submitted to Universiti Teknologi Petronas on 2017-06-09](#)
 < 1% match (student papers from 03-May-2019)

[Submitted to HELP UNIVERSITY on 2019-05-03](#)
 < 1% match (student papers from 26-Dec-2020)

[Submitted to HELP UNIVERSITY on 2020-12-26](#)
 < 1% match (student papers from 23-Sep-2019)

[Submitted to Universidad Carlos III de Madrid on 2019-09-23](#)
 < 1% match (Internet from 10-Dec-2020)

<https://aterasaifuddin.weebly.com/>
 < 1% match (student papers from 27-Apr-2018)

[Submitted to Study Group Australia on 2018-04-27](#)
 < 1% match (Internet from 16-Apr-2016)

http://calhoun.nps.edu/bitstream/handle/10945/9746/01Sep_McDonald.pdf?isAllowed=y&sequence=3
 < 1% match (Internet from 19-Aug-2022)

<https://crapplet8080.blogspot.com/2021/10/dbmsnotes-roadmap-by-love-babbar.html>
 < 1% match (student papers from 13-Feb-2020)

[Submitted to RDI Distance Learning on 2020-02-13](#)
 < 1% match (student papers from 22-May-2020)

[Submitted to Hodges University on 2020-05-22](#)
 < 1% match (student papers from 24-Apr-2020)

[Submitted to Laureate Education Inc. on 2020-04-24](#)
 < 1% match (student papers from 12-Jan-2013)

[Submitted to Help University College on 2013-01-12](#)
 < 1% match (student papers from 15-May-2017)

[Submitted to University of Leicester on 2017-05-15](#)
 < 1% match (student papers from 21-Apr-2023)

[Submitted to University of Bedfordshire on 2023-04-21](#)
 < 1% match (student papers from 29-Oct-2009)

[Submitted to University of Southern Queensland on 2009-10-29](#)
 < 1% match (student papers from 17-Oct-2016)

[Submitted to University of Technology, Sydney on 2016-10-17](#)
 < 1% match (Internet from 21-Mar-2023)

<https://interestingengineering.com/diy/what-exactly-is-html5-anyway>
 < 1% match (student papers from 11-Jan-2013)

[Submitted to City University on 2013-01-11](#)

< 1% match (student papers from 26-Aug-2021)

[Submitted to Universiti Tunku Abdul Rahman on 2021-08-26](#)

< 1% match (Internet from 26-Sep-2019)

<http://releasefrommalwareandspyware.blogspot.com/>

< 1% match (student papers from 20-Jan-2022)

[Submitted to Leeds Beckett University on 2022-01-20](#)

< 1% match (student papers from 05-Apr-2019)

[Submitted to University of Newcastle on 2019-04-05](#)

< 1% match (Internet from 22-Aug-2022)

<https://adoc.pub/rancang-bangun-sistem-informasi-penjualan-pada-pt-berkah-jay.html>

< 1% match (Internet from 26-Dec-2022)

<https://fdocuments.in/document/test-case-example-docnutan.html>

< 1% match (student papers from 06-Apr-2023)

[Submitted to University of Wales Institute, Cardiff on 2023-04-06](#)

< 1% match (student papers from 04-Sep-2021)

[Submitted to Victorian Institute of Technology on 2021-09-04](#)

< 1% match (Internet from 12-Jan-2023)

<http://dspace.daffodilvarsity.edu.bd:8080/bitstream/handle/123456789/6840/172-15-9921%20%2817%25%29.pdf?isAllowed=y&sequence=1>

< 1% match (Internet from 07-May-2023)

<https://fliphtml5.com/khzzn/kvgi/basic>

< 1% match (Internet from 09-Feb-2020)

<https://krex.k-state.edu/dspace/bitstream/handle/2097/12436/TejaswiBattula2011.pdf>

< 1% match (Internet from 11-Jan-2023)

<https://pdfcoffee.com/c-l-i-n-i-c-m-a-n-a-g-e-m-e-n-t-s-y-s-t-e-m-pdf-free.html>

< 1% match (Internet from 26-Mar-2021)

<http://repository.uinjkt.ac.id/dspace/bitstream/123456789/1607/1/MIA%20RAMADHINI-FDK.PDF>

< 1% match (Internet from 02-Aug-2015)

<http://spod.tarc.edu.my/sas/fulltext/201112/aia2/kohweyhongaia201112f.pdf>

< 1% match (Internet from 02-Apr-2023)

<http://umpir.ump.edu.my/id/eprint/26700/1/Mobile%20based%20for%20inventory%20system%20%28MBIS%29.pdf>