The Government Sadiq College
Women University Bahawalpur
Pakistan

‘Consult a Doctor’
Online Medical Consultation System

Session: BSCS Fall 2017-2021

Project Supervisor: Ma’am Muniba Saleem

Submitted by:

Haleema Sadia BSCS/FA17/506

STATEMENT OF SUBMISSION

This is certified that Haleema Sadia successfully completed the final project named as: **'Consult a Doctor'**, at the Department of Computer Science & Information Technology, The Govt. Sadiq College Women University Bahawalpur, to fulfill the requirement of the degree of **BSCS**.

_____	_____
Project Supervisor Ma'am Muniba Saleem Lecturer DOCS&IT -GSCWU	Head of Department Dr. Muhammad Saeed Ahmad

_____
External Examiner

ABSTRACT

With each passing day, people are getting more and more busy in their life, they even don't have time for their health. They find it difficult, to take out time from their busy schedule, to consult with the doctor. So, we have designed an application 'Consult a Doctor' through which they can consult with the doctor while staying at home. Basically, there are three modules in our system, i.e., Admin, Patient and Doctor. In admin module admin verify doctors who want to do their medical practice on our website, manage doctors as well as patients. In Patient module patient can view doctors, request appointment from doctor and consult with the doctor. In Doctor module doctor can respond to patient's appointment request and attend patients.

ACKNOWLEDGMENT

I would like to thank my supervisor **Ma'am Muniba Saleem** as well as our HOD **Dr. Muhammad Saeed Ahmad** who gave us the precious opportunity to do this project on the topic '*Consult a Doctor'*, which helped us doing a lot of research and we came to know about a lot of new things. We are thankful to them.

Secondly, I would also like to thank my Family specially my sister who helped me a lot in Completing this project within the limited time frame.

TABLE OF CONTENTS

Chapter 1: Final Project Proposal

1.1 Introduction

Usually a medical consultation is considered as a two way interaction between patient and a doctor. Patients have to visit the clinic or hospital to seek appointment from the doctor then again they have to visit the doctor for check-up at a time appointed by the doctor. Doctor asked about patient's issues in this interaction. And diagnose their problem then offer necessary treatments.

Medical consultation has an indispensable role in someone's health to protect and manage usual heath issues. In medical consultation doctor suggest treatments on the basis of patient's health issues.

In spite of all that there are some problems in medical consultation system. Such as patient have to visit doctor again and again for single routine check-up, sometime patient feel hesitant to discuss some health issues face to face. Also it is difficult for doctor to keep detail record of patients.

1.2. Project Title:

According to my project's subject online medical consultation system, best suitable title is **"Consult a Doctor"**.

1.3. Project Overview statement:

Through online medication system person will be able to consult a doctor while staying at home. Patients and doctors will have their separate sign up or login pages. Patients and doctor can update their information anytime. It will be easy for doctors to keep track of patient's health.

There will be a two-way interaction between Patients and doctors. They can communicate through text messages and voice calls.

Patients can seek doctor's help in their minor issues. It will be easy for patients to seek for appointments on regular basis; they don't have to go to clinic again and again.

1.4. Project Goals & Objectives:

Goals:

Our main goals are to facilitate patients having common health issues in consulting a doctor while staying at home and to help doctors in keeping health records of their patients regularly, and to check their patient's health update on regular basis.

Objectives;

My objective is to provide

1. An interface which is easy to use for patients and doctors.
2. An interface through which patients and doctors can interact, such as text messages, voice calls or video calls.
3. An interface through which patients can pay check-up dues easily.

1.5. High-level system components:

Major functional units used in an application are:

- ✓ Administration Mode
- ✓ Doctor Mode
- ✓ Patient Mode

Administrative Mode: Involves team having full access to the application, they can modify anything in this mode.

Doctor Mode*: In this mode doctors will have complete access; they can use this mode freely.*

Patient Mode*: It involves patients, who can use this mode according to their need.*

1.6. List of optional functional units:

Functional Requirements:

Functional requirements of the system are:

Welcome Page: page will contain name of the application which will give description about the system.

Login Page: it'll be used for login. This page will have two login options one is for doctors and other for patients.

Home Page: this page will contain appointments or other regular things of patients or doctors.

Profile Page: it will contain detailed information about patients and doctors.

Searching Page: Person will be able to search for specific doctor using this page.

Non-Functional Requirements:

Non-Functional requirements of the system are:

Availability: Application should be available for use, anytime by anyone.

Security: User's information should be secured.

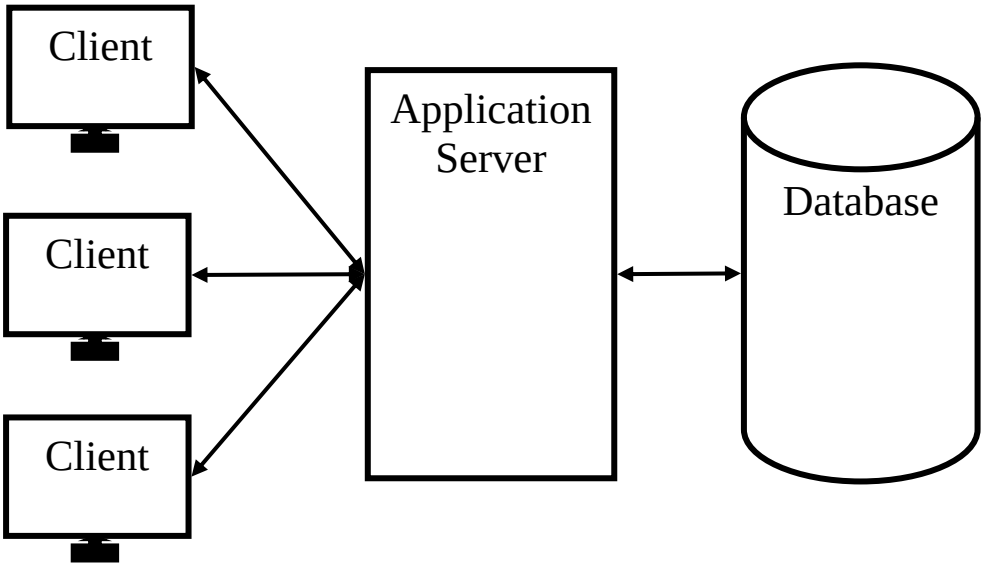
Performance: to keep good performance, application should be enhanced regularly.

Efficiency: application's efficiency matters a lot in an application, that's why application should have good response time.

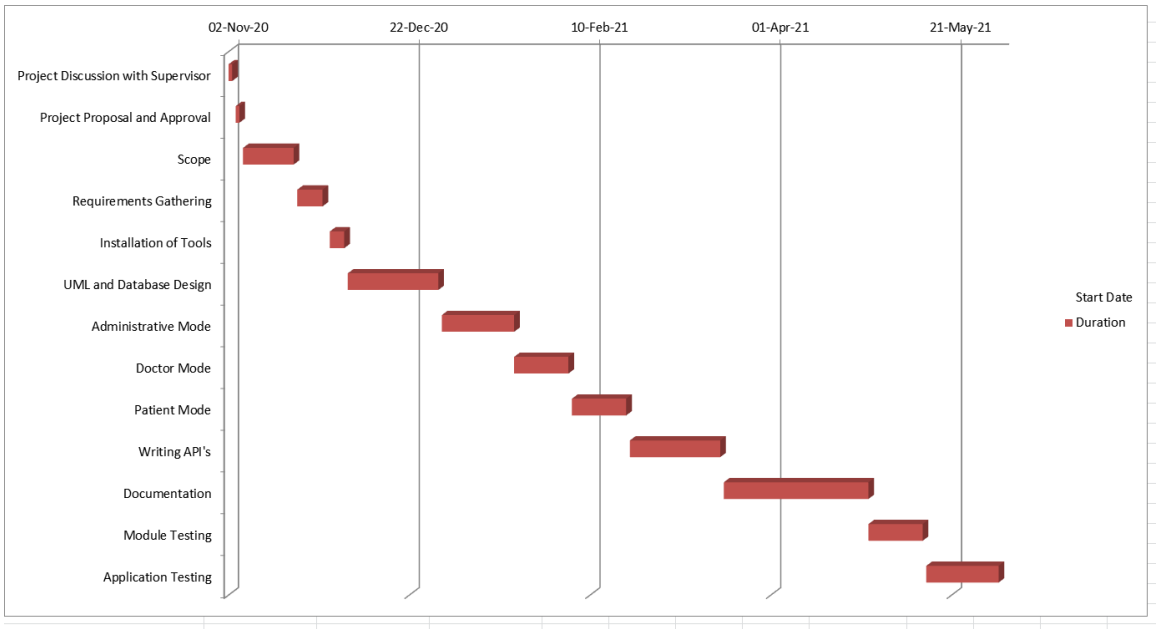
Usability: application should be made in such a way that it can be used by anyone without any specific training.

1.8. Application Architecture:

My application Architecture is Three-tier architecture:



1.9. Gantt chart:



1.10. Hardware and Software Specification:

Hardware Specifications:

Processor Intel® Core™ i7-8565U CPU @ 1.80GHz 1.99GHz

RAM 4.00 GB

System Type 64-bit Operating System, x64-based processor

Software Specifications:

Android Studio

Android(Kotlin)

1.11. Tools and technologies used:

The software that will be used to make an application is:

Android Studio

Frontend Language

C++

Backend Language

Android (Kotlin)

Chapter 2: First Deliverable

2.1 Introduction

Primary deliverable is first deliverable which includes planning and scheduling of project. It is the initial document made in process of planning. This deliverable should have below artifacts:

1. Project Feasibility
2. Project Scope
3. Project Costing
4. Task Dependency Table
5. Critical Path Method Analysis (CPM Analysis)
6. Gantt Chart
7. Introduction to team members
8. Tasks and member assignment table
9. Tools and Technologies
10. Vision Document
11. Risk List
12. Product Features

2.2 Project/Product Feasibility Report

Feasibility report is an important task in project initialization. It is actually a documentation of feasibility study that sums up the analysis and evaluation of proposed idea/solution. We check whether the project is feasible within the available resources and tools, to what extent we can apply our solution. It investigates project purposes for the aim of finding if the project is really feasible, gainful and beneficial. It describes the most feasible solution that can be applied to the project.

There are many types of feasibilities:

- Technical
- Operational
- Economic
- Schedule
- Specification

- Information
- Motivational
- Legal and Ethical

2.2.1. Technical Feasibility

There is basis of our idea. Hardware that is used in our system has the ability to hold the required structure. Our system can respond properly to user's requests without paying attention to the number of users. We have used particular certifications of accuracy, persisting quality and details security.

2.2.2. Operational Feasibility

Operational feasibility means that everyone can operate the system. End user can easily use that system. The system you are designing is operationally feasible or not, that will work and provide the solution which was expected. Our system is addressing all those problems which existing users were facing. Our system will provide solutions of those problem areas that are identified in an existing system. Now the system is operationally feasible it will provide the same expected performance.

2.2.3. Economic Feasibility

In computer science anything is economically expensive in terms of cost and time taken. We will discuss these two factors in economic feasibility. System that we are designing is economically feasible and there is no use of extra resources. We don't need any extra hardware, extra man power, or expertise to develop the system. System is economically feasible it can be developed within available resources and second thing is effectiveness, it means time based performance. Our system will give required output within the given time; it will be developed within given deadline.

2.2.4. Schedule Feasibility

In schedule feasibility most important factor is time. Our project is developed according to the given schedule, within given time, with the use of available resources.

2.2.5. Specification Feasibility

In specification feasibility it is mentioned that we have followed the schedule, and develop the system within given requirements. Features which are required for the system are enough to develop a system. The performance which was expected is achieved.

2.2.6. Information Feasibility

Information which is required to build a system is feasible. We have two type of information, first is the information which a system is providing as an output and second information which a system is using to produce an output. System is capable of producing an output easily and output is reliable which means output is trustworthy and clear. It can be used further if required by the system.

2.2.7. Motivational Feasibility

A flaw of previous system is our motivation. Whatever the output or requirements of the system supposed to be done is accomplished. Our system is removing all flaws and producing better performance.

2.2.8. Legal & Ethical Feasibility

Our software is open source and is easily available. We haven't used pirated software. Plagiarism rate is also less. Our system is free from all legal and ethical issues.

2.3. Project/Product Scope

This is very important part. In project's scope we talk about the edges of the system. System through which we can consult doctors online is referred to as Online Medical Consultation System. It is easy to consult doctor online for some minor issues because it save time and cost. It is also very helpful in situations such as Covid-19. In project's scope we also talk about the attributes of the system.

Our system is providing the facility of interacting with your doctor for minor health issues through online medical consultation system. We have defined payment attribute, appointment attribute, and record attribute in our system.

We have easily added attributes like html, CSS, PHP, JavaScript because these attributes are easily available we don't have to purchase them. Processing speed of our system is also good.

2.10. Tools and Technology with reasoning

Tools and technologies are those which are used for the development of system. We have used the following tools and technologies for front and back end of the system;

HTML: Hypertext Markup Language is used for developing front end or user side of the system. Through html we can give structure to our website and can manage what to content to be viewed on user side.

CSS: Cascading Style Sheet is also used for the front end. Through CSS we can design our website or give style to our website.

JavaScript: JavaScript is used for frontend as well as backend of the system. Through JavaScript we can make our webpage interactive.

PHP: Hypertext Preprocessor is a scripting language. It is used for the back end. Through PHP we can connect our website to database.

Asp.NET: Asp.Net is a server side framework. It is open source framework used to create web applications.

C#: C# is a programming language which is used to create websites, web applications, and web services.

2.11. Vision Document

As by word **vision**, vision document gives the vision of the system. The purpose of designing this system is to make it easy for patients to consult doctors for minor health issues and help doctors to maintain record of their patients. Stakeholders of this system are admin, patients, and doctors.

Interface of our system is user friendly. User can easily understand the system. They don't need to have any specific training to use the system. Admin is allowed to do changes in the system. Admin can modify the system, can update the system. Admin can check user's records and can provide help to the user. User have to login to the system. User can search for doctors, take appointment, and consult doctors.

Tools and technologies used in our system are; html, CSS, JavaScript, PHP, Asp.NET, C#. This system is very helpful for the users. Processing speed is also good.

2.12. Risk List

Our system may face reliability risk.

2.13. Project Decomposition

In project decomposition we decompose our project. Our project is decomposed as followed:

Admin module

Patient module

Doctor module

In admin module we discuss about the authorities admin have. Admin have full control over the system. He or She can make changes to the system and can update the system. Admin also have access to patient and doctor module. In Patient module

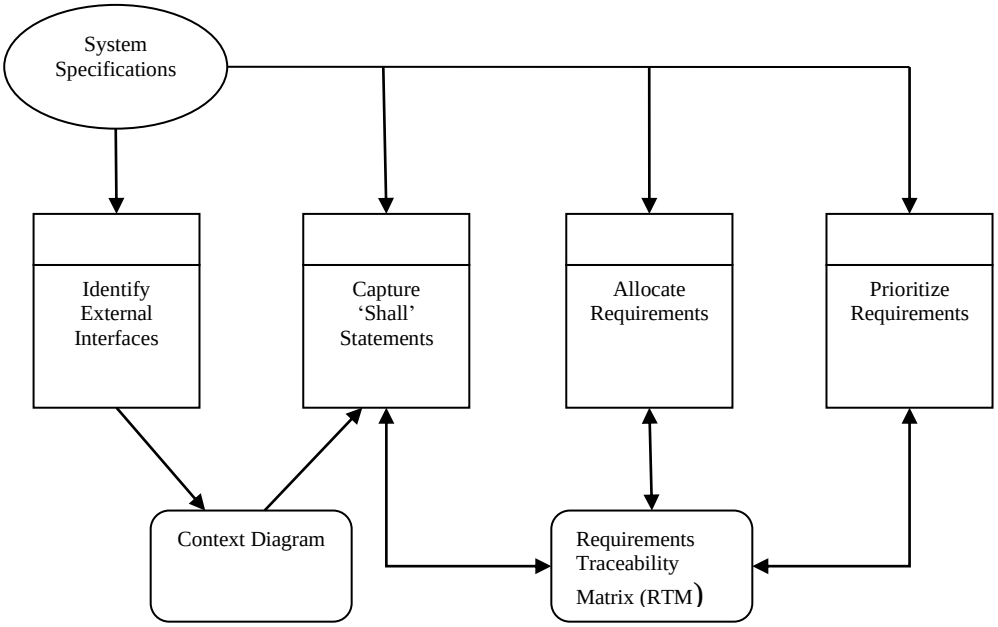
we talk about the authorities' patient have. Patient can search for doctors, take appointment, and can consult doctor. Patients have access to only patient module. In doctor module we elaborate the authorities doctor have. Doctor can check patient's health report and keep patient record. Doctors have access to only doctor module.

Chapter 3: Second Deliverable For Object Oriented Approach

3.1 Introduction:

This chapter is about requirements engineering. In requirement engineering, we are doing system specification and elicitation. We are analyzing our system and capturing ‘shall statements (for example, user shall enter login id and password to enter into the system). In this chapter we will be analyzing what user wants and what are the needs of the user. We will be doing following things:

- Requirements elicitation
- Requirements analysis and negotiation
- Requirements specification
- System modeling
- Requirements validation
- Requirements management



For the specification of the system we have performed four steps:

- Identify external interfaces

  First of all we have identified our system interface which means after development we will identify the interfaces that user will have. Such that user will have separate login interface for patients and doctors, interface for registration, interface for home page, interface for search page and many more.

- Capture "shall statements

  By identifying interfaces we have got to know that what are the 'shall' statements means we have identified that what user have to do, what admin have to do and what system have to do.

- Allocate requirements

  After knowing about the 'shall' statements we will allocate requirements.

- Prioritize requirements

  When we have allocated requirements we prioritize those requirements. We give high priority (those requirements which must be

fulfilled at any cost for the system), medium priority (those which needs to be fulfilled for the improvement of the system) or low priority (those which may be fulfilled but doesn't harm system if not fulfilled).

- Development of requirements traceability matrix

3.1.1 Systems Specifications

While giving system specifications following clauses must be included.

Introduction

'Consult a Doctor' is an online medical consultation system. Mostly patients felt difficult to go to the doctor to discuss minor issues. Doctors also find it difficult to keep track of their patients' health record. We have designed this application to help patients in discussing minor health issues with the doctor while staying at home. Through this application patient can easily consult doctor, doctor will be able to record their patients' health details. Patients and doctors can have text conversation or voice calls. User don't have to worry about clinical dues, our application provides the facility of paying dues through any account.

Existing System

Basically there are 3 parts in our system i.e. Patient, Doctor and Admin. Each of the modes has their separate specifications, interfaces and many more.

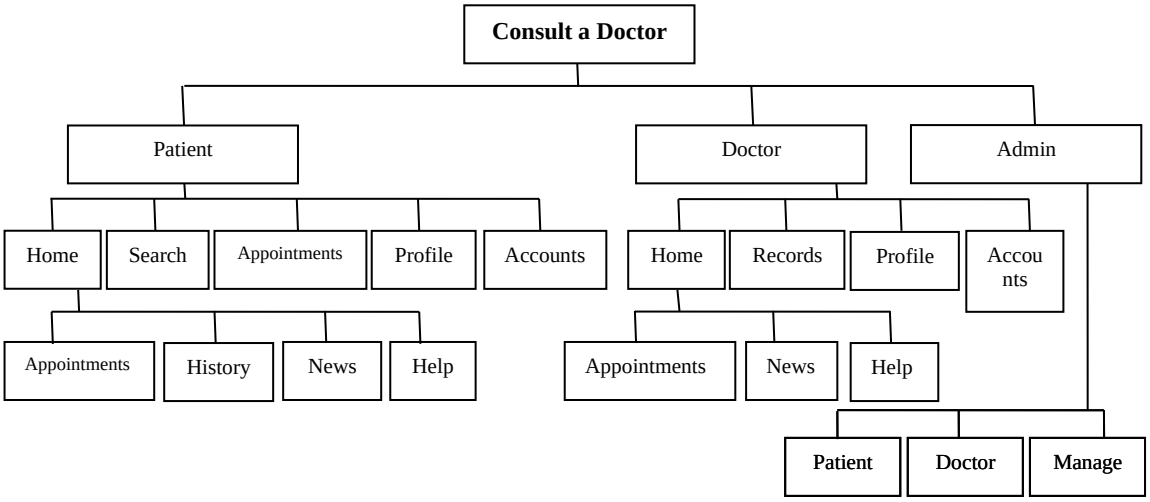
Patient: Patient have separate login interface. They have a profile page which contains their personal information. They will be able to search for doctors through search interface. They can save their appointments. There will be a home page which contains patients' appointments, recent history, latest news, and help. They can pay dues.

Doctor: As patients, doctors also have login interface. They have a profile page. They can save patients' health records. Doctor's home page contains appointment details, latest news, and help. Doctor can mention their schedule.

Admin:

Admin can login to the system. They have complete access to the system. They can modify the system, and can update the system with time.

Organizational Chart



3.2 Scope of the System

Usually system has three phases but our project is limited to Phase I, in which we have designed and developed our system. Next two phases are for working of the system which is not included in our document.

Phase I

Phase I involve

- System Management
- Registration

3.2.1. Summary of Requirements: (Initial Requirements)

Usually patients visit doctor for every minor or major health issues. They have to go to clinic, bare convince expenses. My system ‘Consult a Doctor’ is about online medical consultation. Main purpose of my system is that patients can consult doctor for minor issues by staying at home because it is the easy way. It will save time and cost.

System allows both user and admin to enter into the system. Admin is allowed to modify the system; he/she can make changes in the system according to the

customer's needs. Admin can also view records, profiles of patients and doctors. She/he can also provide help to the user. Admin can update the system with time.

User can login to the system by entering username and password. Patient can search for doctors, can take appointments, and can save their medical details. Patient can also save their mostly viewed doctors. Doctor can ask issues of patient through text or voice calls, can ask patient to pay through any account, and can save patients' health record. Doctor can also review previous records of the patient, and can mention their schedule.

3.2.2. Identifying External Entities:

External entities identification will depend on information present in the summary of the system.

External entities identification is performed in two phase:

Over Specify Entities from Abstract:

On the basis of the summary, there are following entities in our system:

- Patient
- Doctor
- Appointment
- Record
- Account
- Schedule

Perform Refinement:

After specifying the entities, we have refined them according to our system. We found following two entities most relevant to the system:

- Doctor
- Patient

3.2.3. Capture "shall" Statements:

Para #	Initial Requirements
1.0	A doctor and patient “shall” registered himself to the system
1.0	Admin “shall” accept, reject, and temporarily put the requests on pending based on provided details.
1.0	A doctor and patient “shall” login to the system and can change his password
1.0	Admin “shall” view doctor’s and patient’s Requests.
2.0	Patient “shall” request for appointment for consultation.
2.0	Admin “shall” accept or reject appointment based on provided details.
3.0	Patient “shall” define symptoms or provide medical report.
3.0	Patient “shall” pay for appointment and provide invoice, confirmation receipt. (pay for appointment)
3.0	Admin “shall” manage patient’s clinical payments.
4.0	Doctor “shall” check patient’s appointment before check-up.
4.0	Doctor “shall” check previous record of patient before check-up.
4.0	Doctor “shall” attend patients on the basis of the symptoms of patients.
4.0	Doctor “shall” prescribe test or medicine if needed.

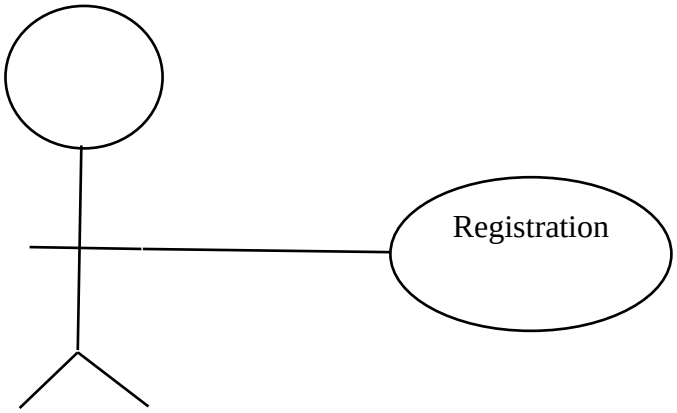
3.2.4. Allocate Requirements:

Para #	Initial Requirements	Use Case Name
1.0	A doctor and patient “shall” registered himself to the system.	UC_Registration
1.0	Admin “shall” accept, reject and temporarily put the request on pending based on provided details.	UC_Process_Registration_Request
1.0	A doctor and patient “shall” sign in to the system.	UC_Login
1.0	Admin “shall” view doctor’s and patient’s Requests.	UC_View_User_Details
2.0	Patient “shall” request appointment for consultation.	UC_Request_Appointment
2.0	Admin “shall” accept or reject appointment on the basis of credentials provided.	UC_Accept_Appointment UC_Reject_Appointment
3.0	Patient “shall” define symptoms or provide medical report.	UC_Define_Symptoms
3.0	Patient “shall” pay for appointment and provide invoice, confirmation receipt.	UC_Pay_For_appointment
3.0	Admin “shall” manage patient’s clinical payments.	UC_Manage_Payments
4.0	Doctor “shall” check patient’s appointment before check-up.	UC_Search_appointments
4.0	Doctor “shall” check previous record of patient before check-up.	UC_Search_Record
4.0	Doctor “shall” attend patients on the basis of the symptoms of patients.	UC_Check_Patient
4.0	Doctor “shall” prescribe test or medicine if needed.	UC_Prescribe_Test/Medicine

3.2.5. Priorities Requirements:

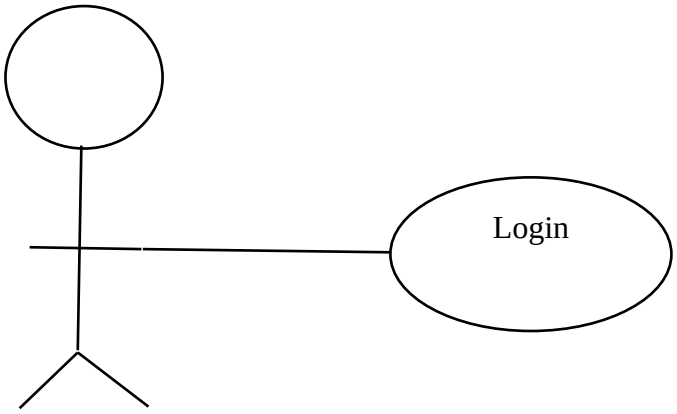
Para #	Rank	Initial Requirements	Use Case ID	Use Case Name
1.0	Highest	A doctor and patient "shall" registered himself to the system	UC_1	UC_Registration
1.0	Highest	A doctor and patient "shall" sign in to the system.	UC_2	UC_Login
2.0	Highest	Patient "shall" request appointment for consultation.	UC_3	UC_ Request_Appointment
4.0	Highest	Doctor "shall" attend patients based on the symptoms of patients.	UC_4	UC_ Check_Patient
3.0	Highest	Admin "shall" manage patient's clinical payments.	UC_5	UC_Manage_Payments
1.0	Medium	Admin "shall" accept, reject and temporarily put the requests on pending based on the provided details.	UC_6	UC_Process_Registration_Request
1.0	Medium	Admin "shall" view doctor's and patient's Requests.	UC_7	UC_ View_User_Details
2.0	Medium	Admin "shall" accept or reject appointment	UC_9 UC_10	UC_Accept_Appointment UC_Reject_Appointment

		based on provided details.		
3.0	Medium	Patient “shall” define symptoms or provide medical report.	UC_11	UC_Define_Symptoms
3.0	Medium	Patient “shall” pay for appointment and provide invoice, confirmation receipt.	UC_12	UC_Pay_For_Appointment
4.0	Medium	Doctor “shall” check previous record of patient before check-up.	UC_13	UC_Search_Records
4.0	Medium	Doctor “shall” prescribe test or medicine if needed.	UC_14	UC_Prescribe_Test/Medicine
4.0	Lowest	Doctor “shall” check patient’s appointment before check-up.	UC_16	UC_ Search_appointments



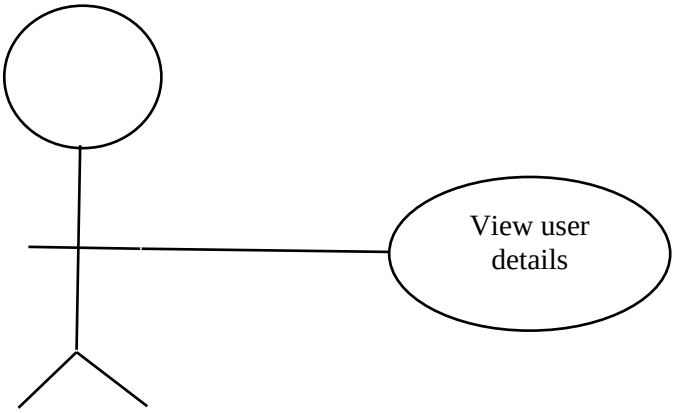
UC-1 User

<table>
<tr><td>Use case id</td><td>UC-1</td></tr>
<tr><td>Use case name</td><td>Registration</td></tr>
<tr><td>Actor</td><td>User</td></tr>
<tr><td>Precondition</td><td>User open the system.</td></tr>
<tr><td>Primary scenario:</td><td>
<table>
<tr><th>User action</th><th>System response</th></tr>
<tr><td>1 User press registration button.
2 User enter required details.</td><td>1. System display registration form
2. System verify details and display login page.</td></tr>
</table>
</td></tr>
<tr><td>Secondary scenario:</td><td>
<table>
<tr><th>User action</th><th>System response</th></tr>
<tr><td>1 User enter invalid sign up detail
2 User re-enter details.</td><td>1. System check sign up detail and display error message.
2. System verify sign up details and display home page.</td></tr>
</table>
</td></tr>
<tr><td>Exception:</td><td>User is already registered into the system.</td></tr>
<tr><td>Post condition:</td><td>User registered successfully into the system.</td></tr>
</table>



UC-2 User

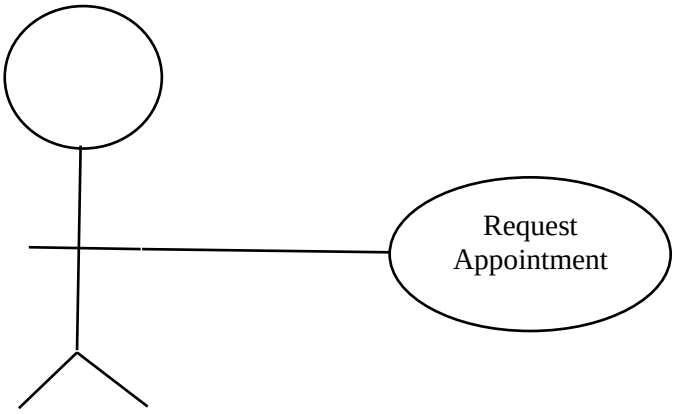
<table>
<tr><td>Use case id</td><td>UC-2</td></tr>
<tr><td>Use case name</td><td>Login</td></tr>
<tr><td>Actor</td><td>User</td></tr>
<tr><td>Precondition</td><td>User is registered into the system.</td></tr>
<tr><td>Primary scenario:</td><td>
<table>
<tr><th>User action</th><th>System response</th></tr>
<tr><td>1. User enter URL of website
2. User enter user ID and password</td><td>3. System display sign in webpage
4. System verify sign in detail and display homepage.</td></tr>
</table>
</td></tr>
<tr><td>Secondary scenario:</td><td>
<table>
<tr><th>User action</th><th>System response</th></tr>
<tr><td>1. User enter wrong sign in detail
2. User re-enter login detail</td><td>1. System verify sign in detail and display error message.
2. System verify sign in detail and display home page.</td></tr>
</table>
</td></tr>
<tr><td>Exception:</td><td>User is not registered into the system.</td></tr>
<tr><td>Post condition:</td><td>User is sign in to the system successfully.</td></tr>
</table>



UC-7 Admin

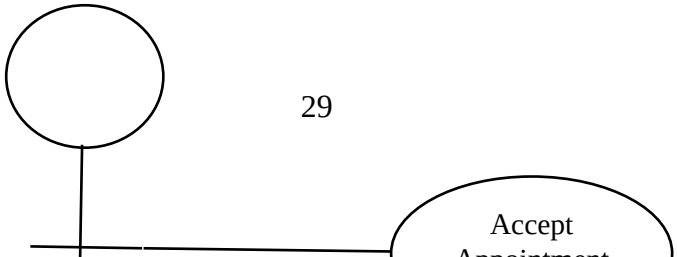
<table>
<tr><td>Use case id</td><td>UC-7</td></tr>
<tr><td>Use case name</td><td>View/Update user details</td></tr>
<tr><td>Actor</td><td>Admin</td></tr>
<tr><td>Precondition</td><td>User enter details into the system</td></tr>
<tr><td>Primary scenario:</td><td><table><tr><th>User action</th><th>System response</th></tr><tr><td>1. User enter details.
2. User enter details to update.</td><td>1. System view user details.
2. System update and view user details.</td></tr></table></td></tr>
<tr><td>Secondary scenario:</td><td><table><tr><th>User action</th><th>System response</th></tr><tr><td>1. User enter invalid updating details.

2. User re-enter updating details.</td><td>1. System check updating details and display error message.
2. System verify updating details and update user details.</td></tr></table></td></tr>
<tr><td>Exception:</td><td>Admin has already updated details into the system.</td></tr>
<tr><td>Post condition:</td><td>Admin successfully updated and view details.</td></tr>
</table>



UC-3 Patient

Use case id	UC-3
Use case name	Request Appointment
Actor	Patient
Precondition	Patient is logged into the system.
Primary scenario:	**User action** / **System response** 1. User request for appointment. / 1. System display appointment form. 2. Patient enter his detail in an appointment form. / 2. System accept and confirm appointment.
Secondary scenario:	**User action** / **System response** 1. Patient enter invalid appointment details. / 1. System display error message after checking details. 2. Patient re-enter appointment details / 2. System verify details and confirm appointment.
Exception:	
Post condition:	Patient took appointment successfully.

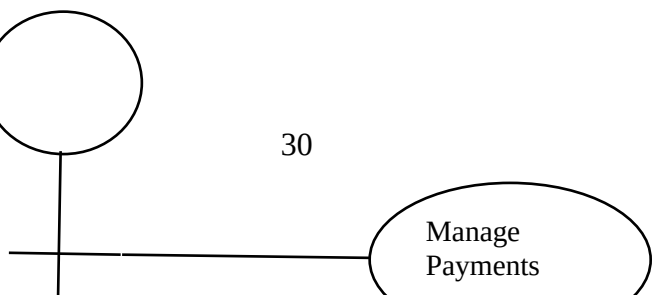


<table>
<tr><td>Use case id</td><td>UC-4</td></tr>
<tr><td>Use case name</td><td>Accept Appointment</td></tr>
<tr><td>Actor</td><td>Doctor</td></tr>
<tr><td>Precondition</td><td>Doctor is logged into the system.</td></tr>
<tr><td>Primary scenario:</td><td><table><tr><th>User action</th><th>System response</th></tr><tr><td>1. Doctor check patient has appointed or not.

2. Doctor ask for patient symptoms and medical reports.
3. Doctor prescribe medicine or test.</td><td>1. System display the appointments detail of patient.
2. System display patient details.

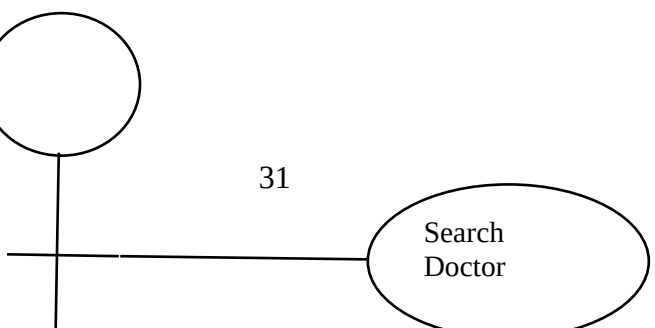
3. System display prescription to the patient.</td></tr></table></td></tr>
<tr><td>Secondary scenario:</td><td><table><tr><th>User action</th><th>System response</th></tr><tr><td>1. Doctor enter invalid patient name to check appointment.
2. Doctor re-enter patient name.</td><td>1. System check and display error message.

2. System verify and display patient’s appointment detail.</td></tr></table></td></tr>
<tr><td>Exception:</td><td>Patient is not available in doctor’s appointments record.</td></tr>
<tr><td>Post condition:</td><td>Doctor successfully check patient.</td></tr>
</table>



<table>
<tr><td>Use case id</td><td>UC-5</td></tr>
<tr><td>Use case name</td><td>Manage Payments</td></tr>
<tr><td>Actor</td><td>Admin</td></tr>
<tr><td>Precondition</td><td>Admin has access to the system.</td></tr>
<tr><td>Primary scenario:</td><td><table><tr><th>User action</th><th>System response</th></tr><tr><td>1. Admin ask for patient’s checkup details.
2. Admin generate payment receipt.</td><td>1. System check and display patient’s checkup details.
2. System display patient’s paid slip.</td></tr></table></td></tr>
<tr><td>Secondary scenario:</td><td><table><tr><th>User action</th><th>System response</th></tr><tr><td>1. Admin generate invalid payment receipt.

2. Admin regenerate receipt.</td><td>1. System check receipt and display error message.
2. System verify and display receipt to the patient.</td></tr></table></td></tr>
<tr><td>Exception:</td><td>User has already paid the dues.</td></tr>
<tr><td>Post condition:</td><td>Admin successfully receive payments.</td></tr>
</table>



<table>
<tr><td>Use case id</td><td>UC-13</td></tr>
<tr><td>Use case name</td><td>search doctor</td></tr>
<tr><td>Actor</td><td>User</td></tr>
<tr><td>Precondition</td><td>User is logged into the system.</td></tr>
<tr><td>Primary scenario:</td><td><table><tr><th>User action</th><th>System response</th></tr><tr><td>1. User ask for specific records.
2. User perform action on records.</td><td>1. System display required records.
2. System update and save records.</td></tr></table></td></tr>
<tr><td>Secondary scenario:</td><td><table><tr><th>User action</th><th>System response</th></tr><tr><td>1. User enter wrong name of required record.
2. User re-enter record name.</td><td>1. System checks and display error message.

2. System verify and display records.</td></tr></table></td></tr>
<tr><td>Exception:</td><td>Record doesn’t exist.</td></tr>
<tr><td>Post condition:</td><td>User successfully checked records.</td></tr>
</table>

Chapter 4: Third Deliverable For Object Oriented Approach

4.1. Introduction:

Third deliverable is about software design. As we have completed the analysis in previous deliverables, so we can know about problem domain. Let's look for the solution of problem domain using Object Oriented approach. Third deliverable must contain the artifacts given below:

- Sequence Diagram
- Collaboration Diagram
- Activity Diagram
- ER-Diagram
- Dataflow Diagram

Now we will describe these artifacts one after another:

4.2. Sequence Diagram

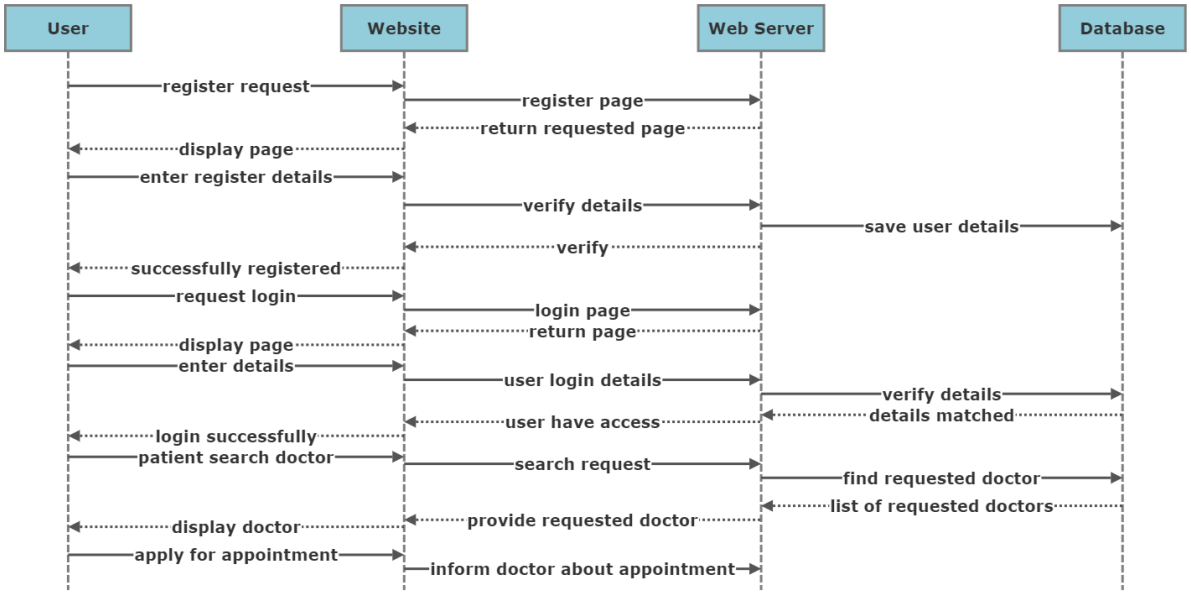


Figure 4.1: Sequence Diagram

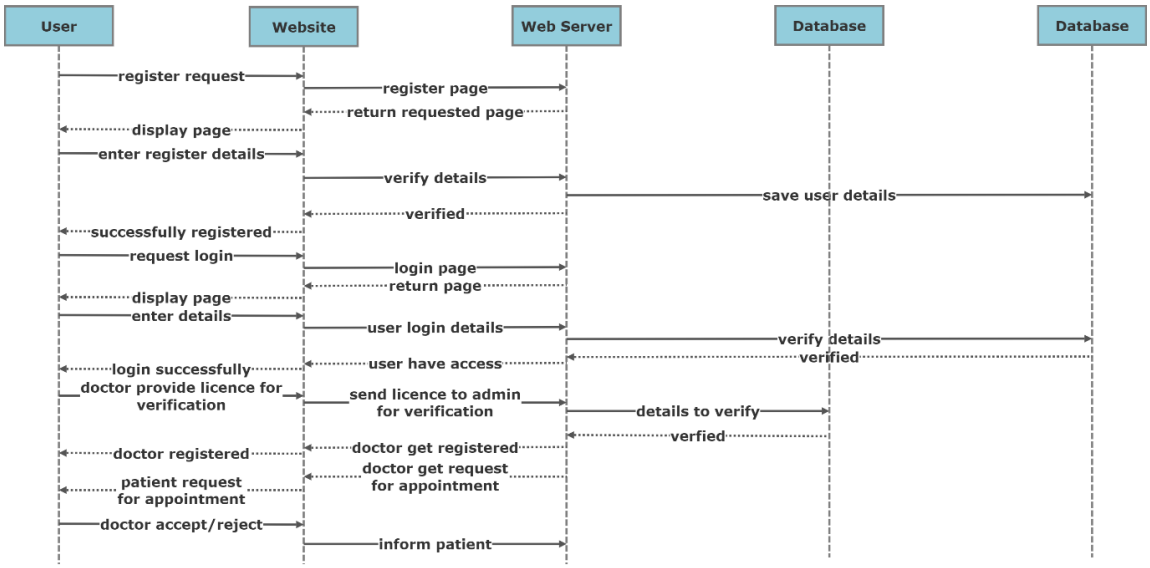


Figure 4.2: Sequence Diagram

4.3. Collaboration Diagram

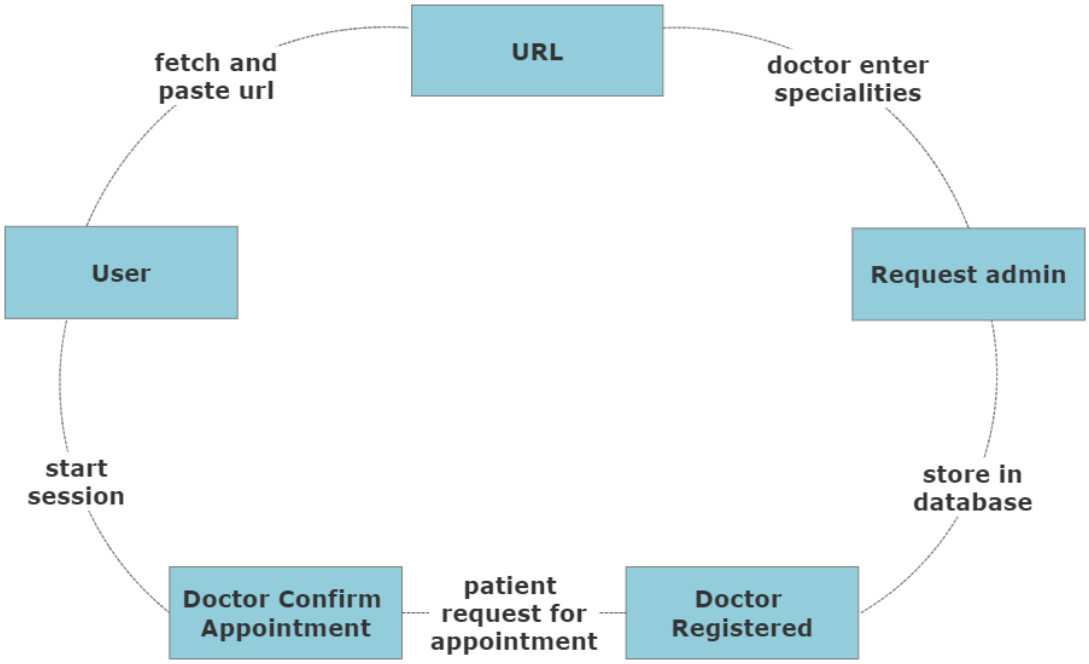


Figure 4.3: Collaboration Diagram

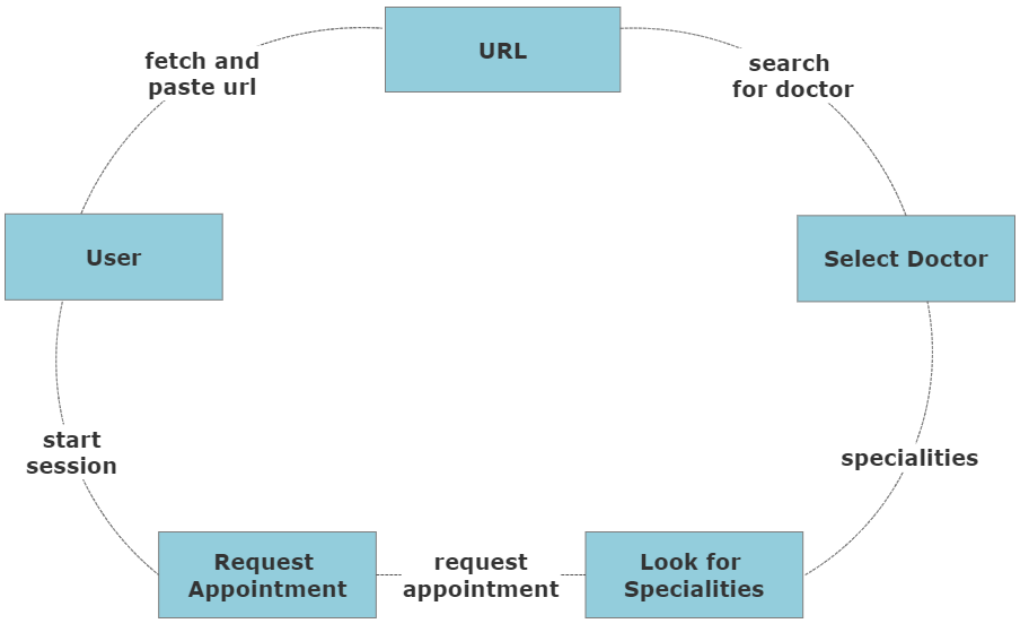


Figure 4.4: Collaboration Diagram

4.4. Activity Diagram

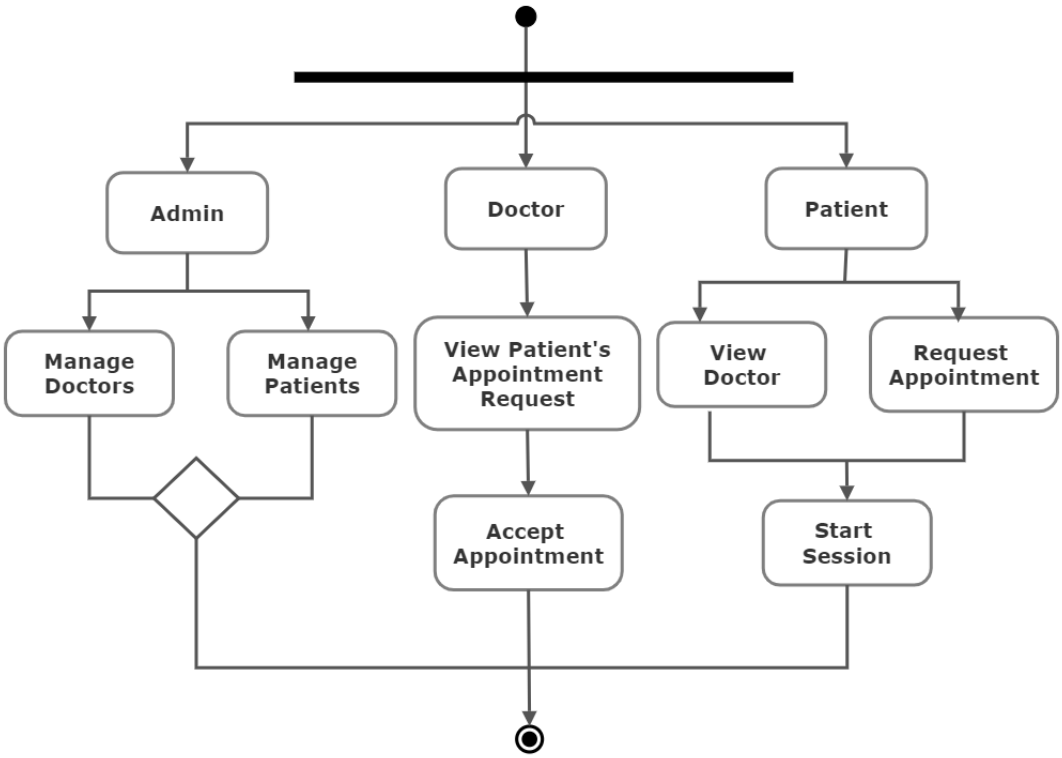


Figure 4.5: Activity Diagram

4.5. Entity Relationship Diagram

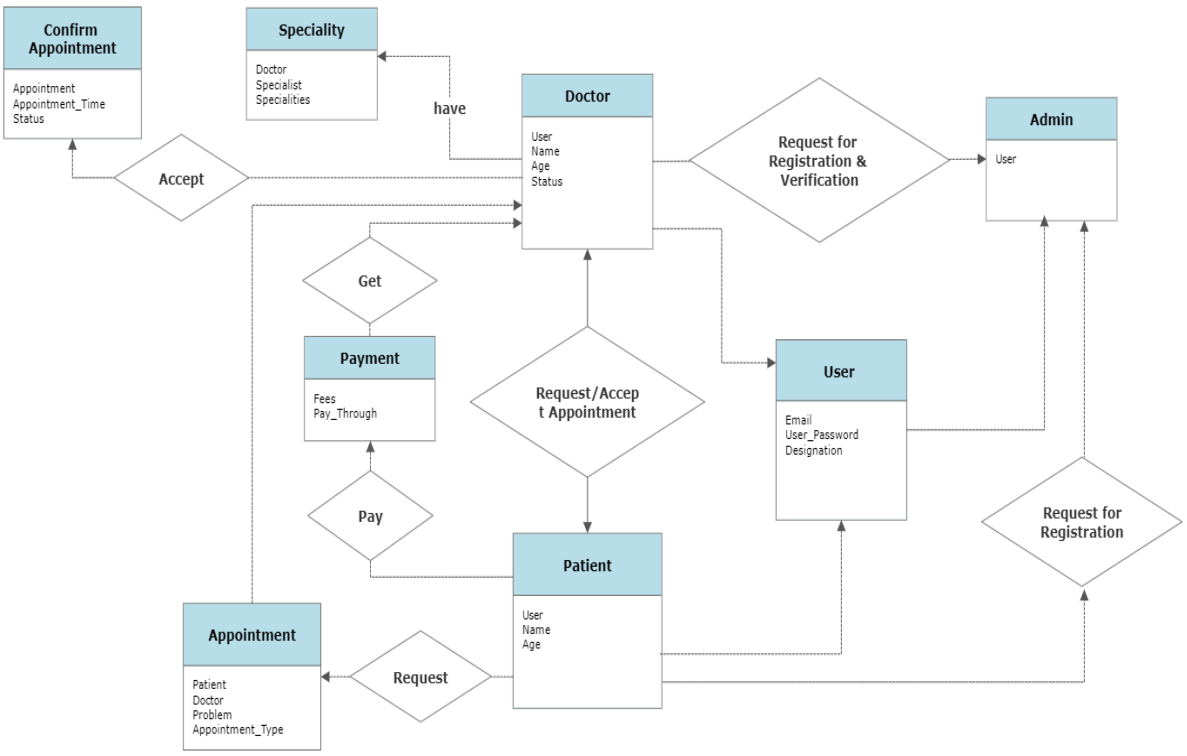


Figure 4.6: ER-Diagram

4.7. Class Diagram

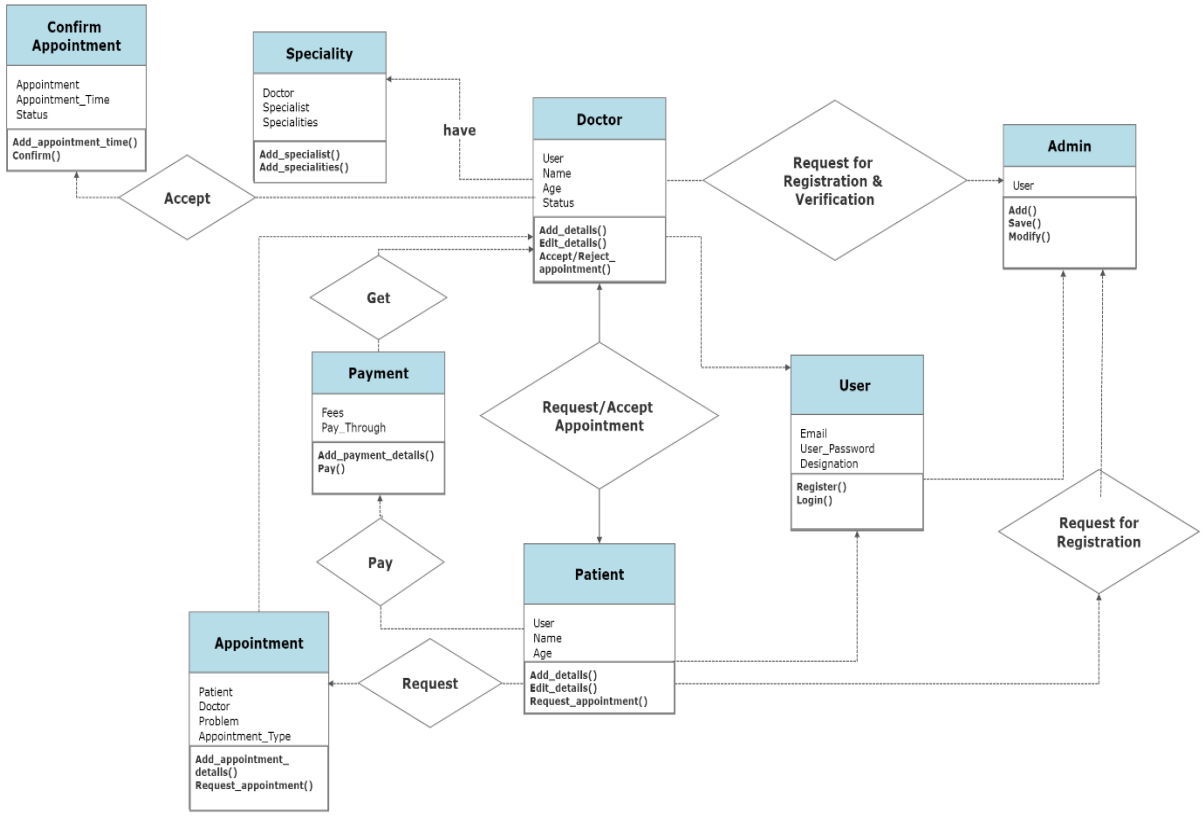


Figure 4.7: Class Diagram

4.7. Data flow Diagram (DFD)

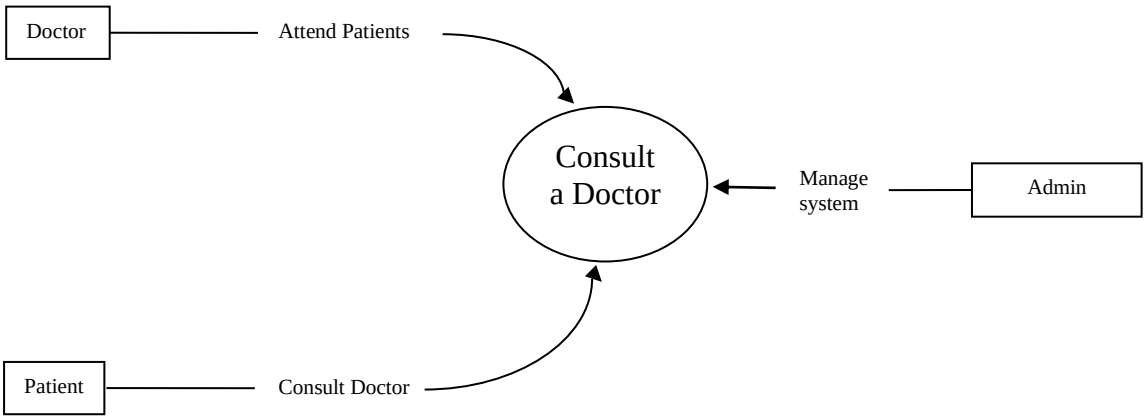


Figure 4.8: Data Flow Diagram Level 1

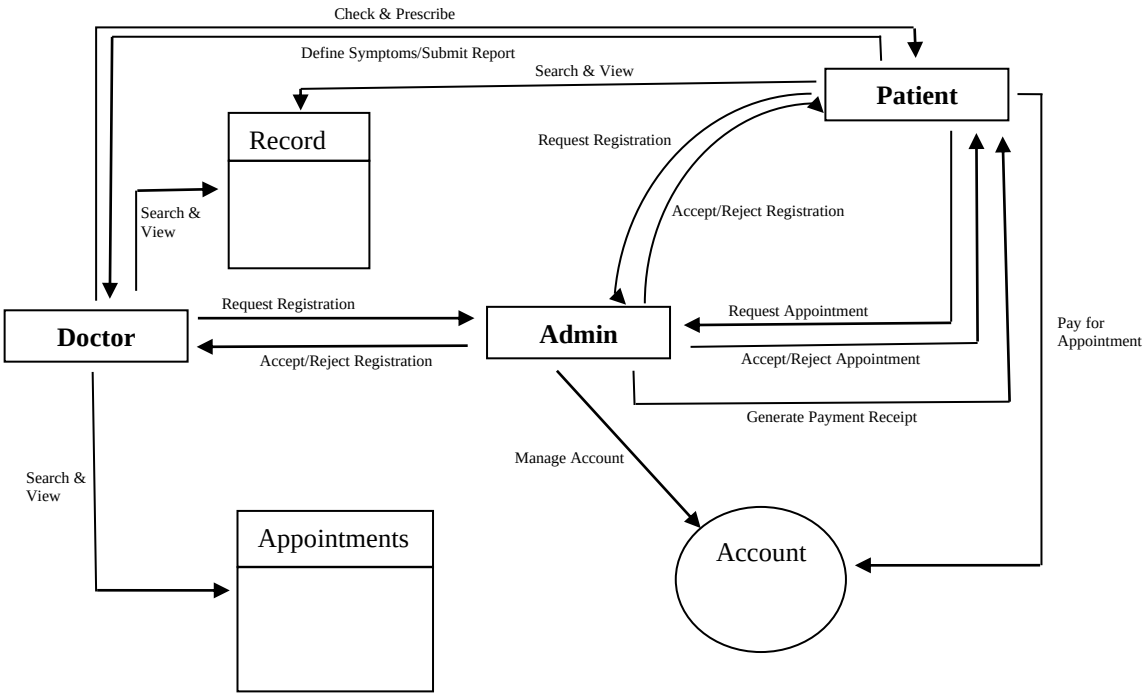


Figure 4.9: Data Flow Diagram Level 0

Chapter 5: Software Testing

5.1. Introduction:

In previous chapters we have completed designing phase, now we will be doing software testing.

My website named 'Consult a Doctor' is for patients to consult with certified specialists online. Basically, we aim to build a website that can manage activities of online medical consultation, and all these activities are handled under admin supervision. With time it is becoming a need to consult a doctor online.

5.2. Test Plan

Purpose	To ensure the successful development of the system.
Approach	
Features to be Tested	Following need to be tested: ▪ Login Profile Verification ▪ Add Doctors ▪ Delete Doctors ▪ Modify Doctors ▪ Search Doctor ▪ Request Appointment

5.2. Test Cases

We have tested the features of our project which are given below:

Login Profile Verification:

Test Case name	Login Profile Verification
Requirement	User must have valid details.
Pre-Condition	Good internet connection is required.
Steps	1- Users click on website. 2- Users enter valid details. 3- User press login button.
Expected Result	Users logged in successfully. Users have permission to access website.

Add Doctors:

Test Case name	Add Doctors
Requirement	Admin must have valid details about the added doctor.
Pre-Condition	Admin must verify the doctor.
Steps	1- Admin check request from doctor for verification. 2- Admin check the required details, i.e., doctor license. 3- Click on submit button.
Expected Result	Hostels in suitable facilities are added.

Delete Doctor:

Test Case name	Delete Doctor
Requirement	Doctor must exist in the system.
Pre-Condition	Doctor must be added in the system.
Steps	1- Admin select the doctor which is to be deleted. 2- Press delete button.
Expected Result	Doctor is removed from the system permanently.

Modify Doctor:

Test Case name	Modify Doctor
Requirement	Doctor must exist in the system.
Pre-Condition	Admin can remove doctors from the system.
Steps	1- Admin can modify existing doctor. 2- Hostel Admin can change required details. 3- Add on modify button.
Expected Result	Doctor is modified in the system.

Search Doctor:

Test Case name	Search Doctor
Requirement	User must know the correct details of the doctor who is being searched.
Pre-Condition	Users know which doctor is to search.
Steps	1- User click on the search bar. 2- User looks for the doctor. 3- Users press the search button. 4- Matching doctors list is displayed.
Expected Result	User successfully search the required doctor.

Request Appointment:

Test Case name	Request Appointment
Requirement	User must have found the required doctor.
Pre-Condition	User must have logged in to the system.
Steps	1- User find the doctor. 2- User press request appointment button. 3- Mention their problem. 4- Press submit button to submit appointment.
Expected Result	User successfully requested an appointment.

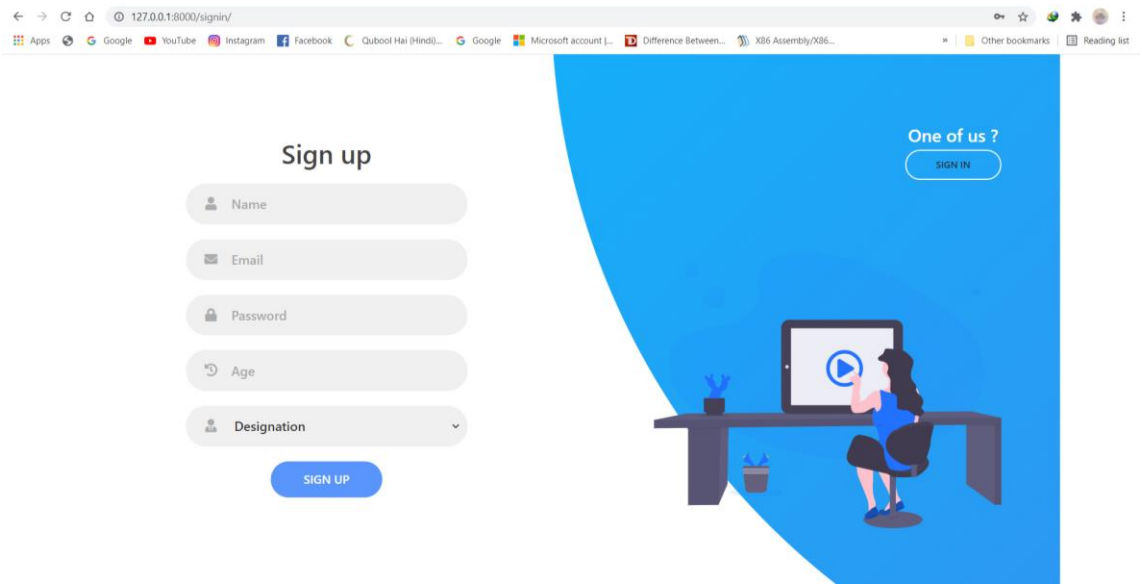
Chapter 6: User Interface Design

6.1. Introduction:

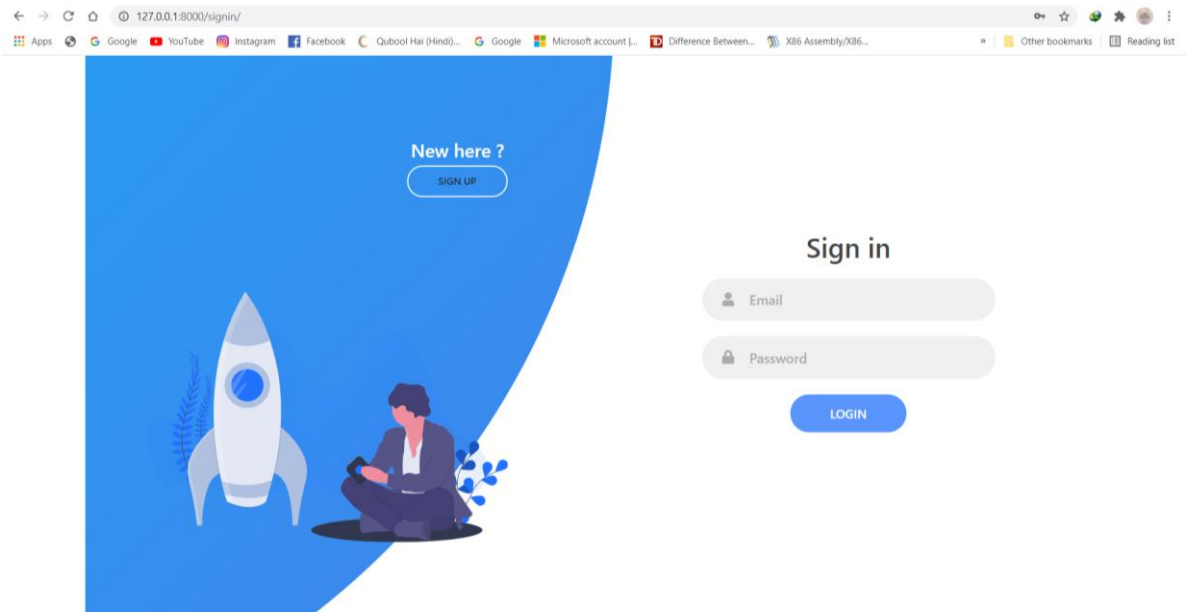
Previously, all the phases of project development, designing, and testing have been discussed. Now we will be discussing user interface design.

My website named ‘Consult a Doctor’ is for patients to consult with certified specialists online. Basically, we aim to build a website that can manage activities of online medical consultation, and all these activities are handled under admin supervision. With time it is becoming a need to consult a doctor online.

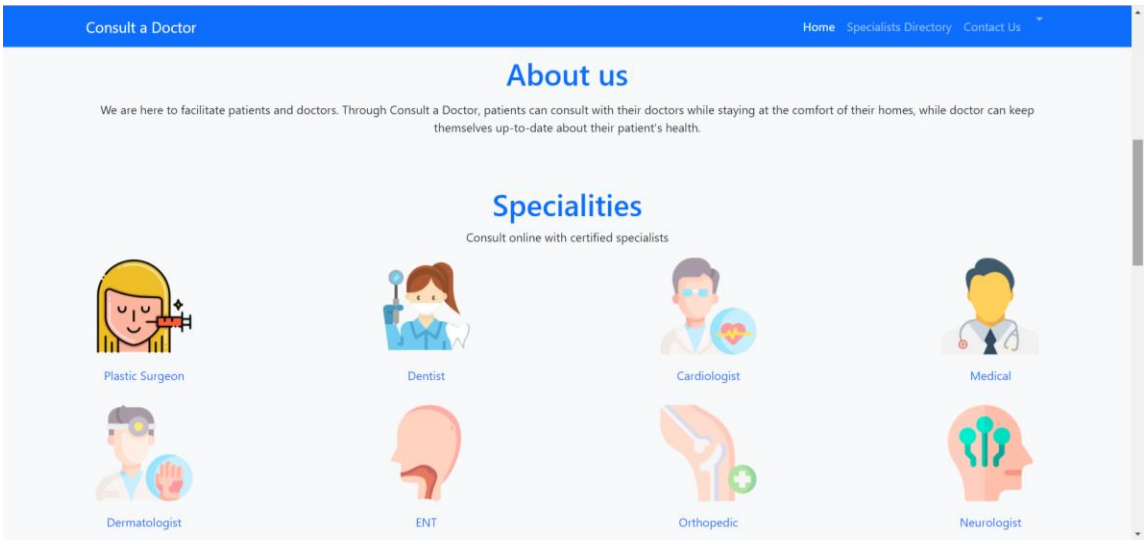
6.2. Signup:



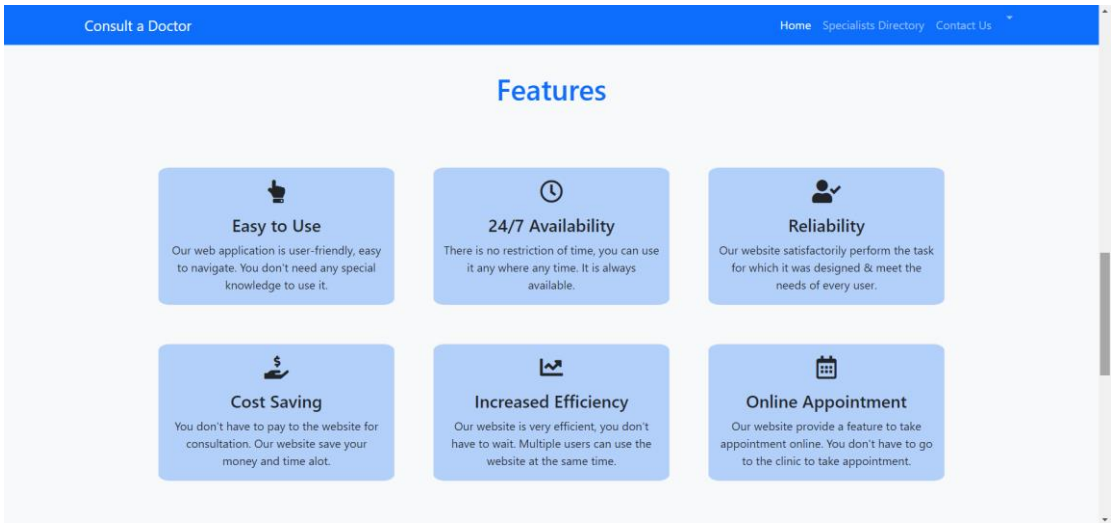
6.3. Sign in:



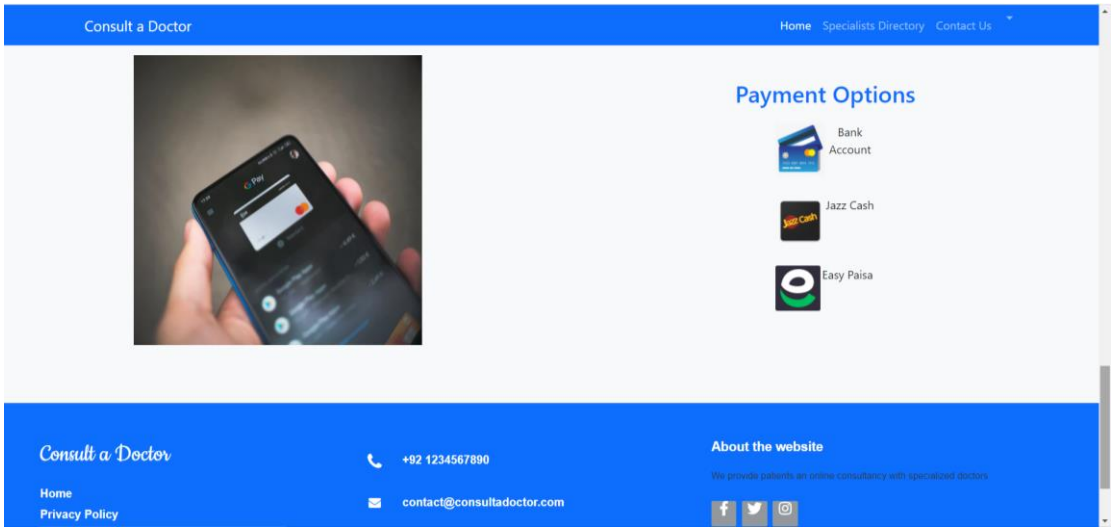
6.4.1. Home page:



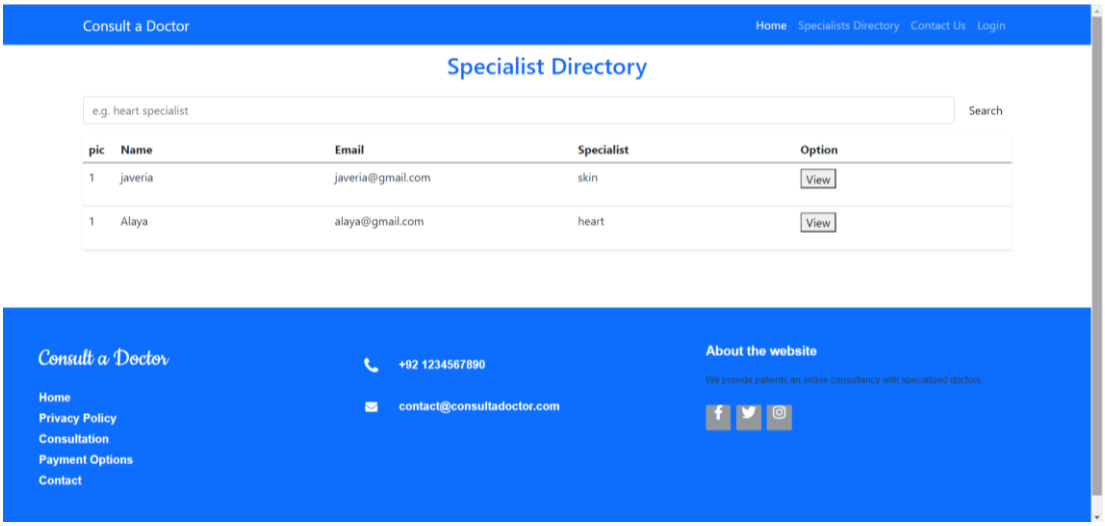
6.4.2. Home page:



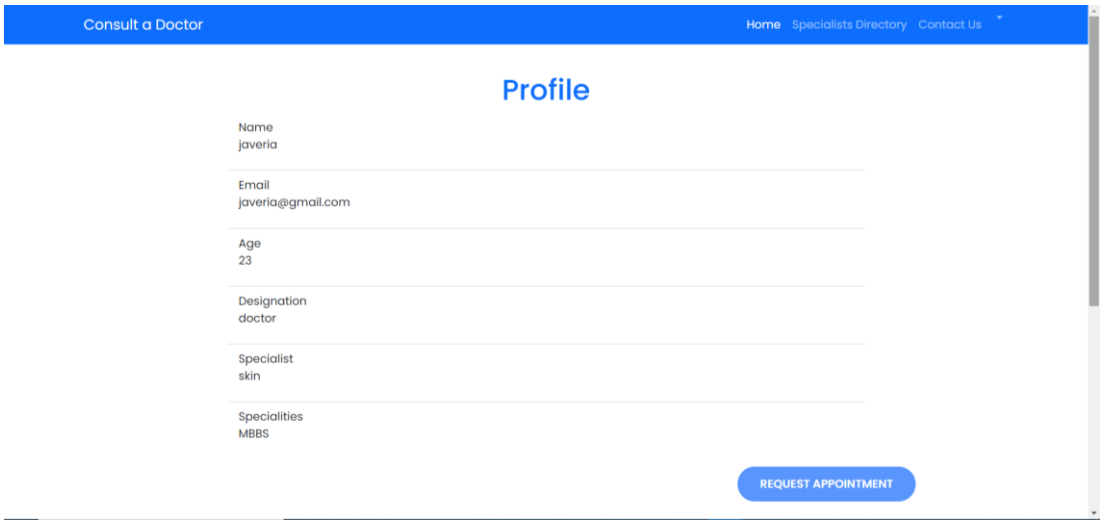
6.4.3. Home page:



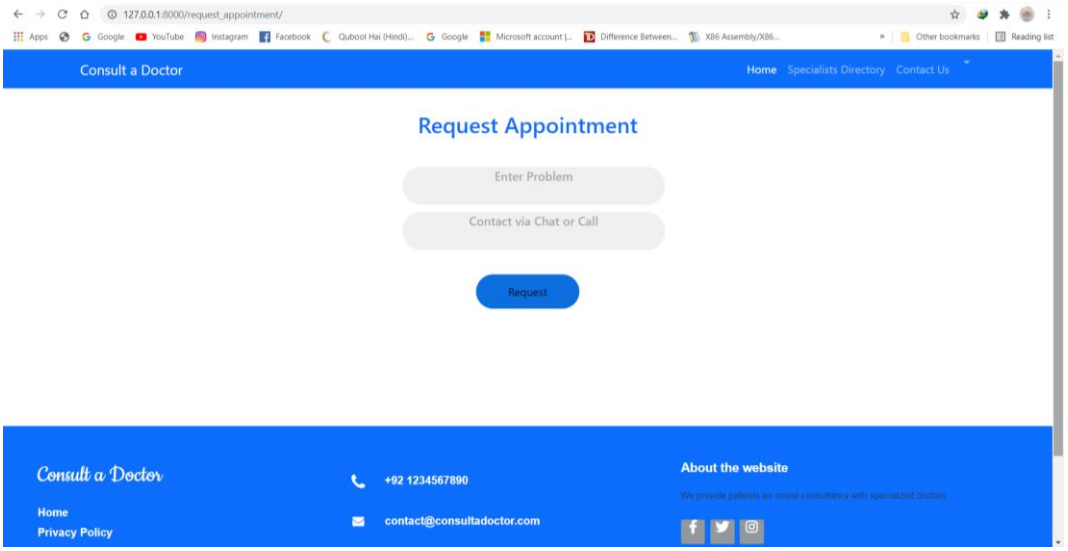
6.5. View of Doctors list to User:



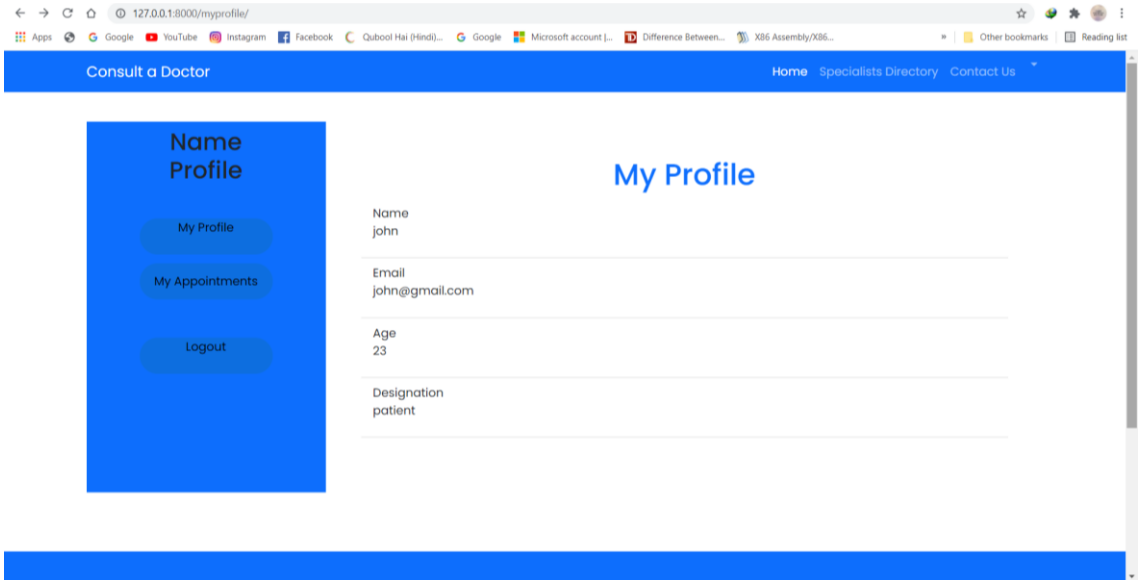
6.6. Doctor's Profile View to Patient:



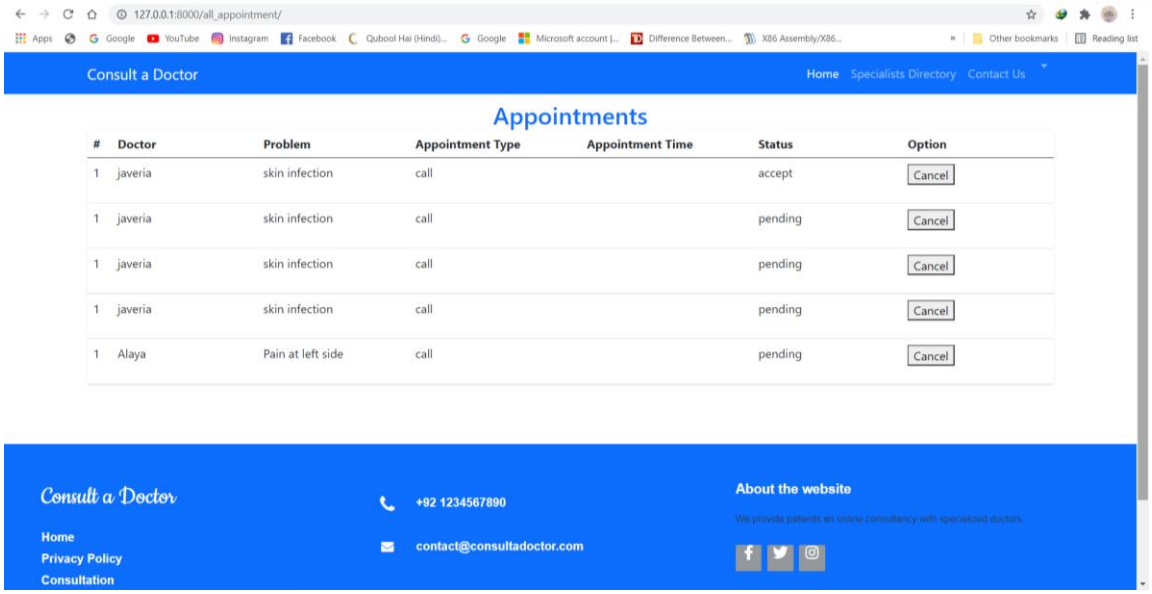
6.7. Request Appointment:



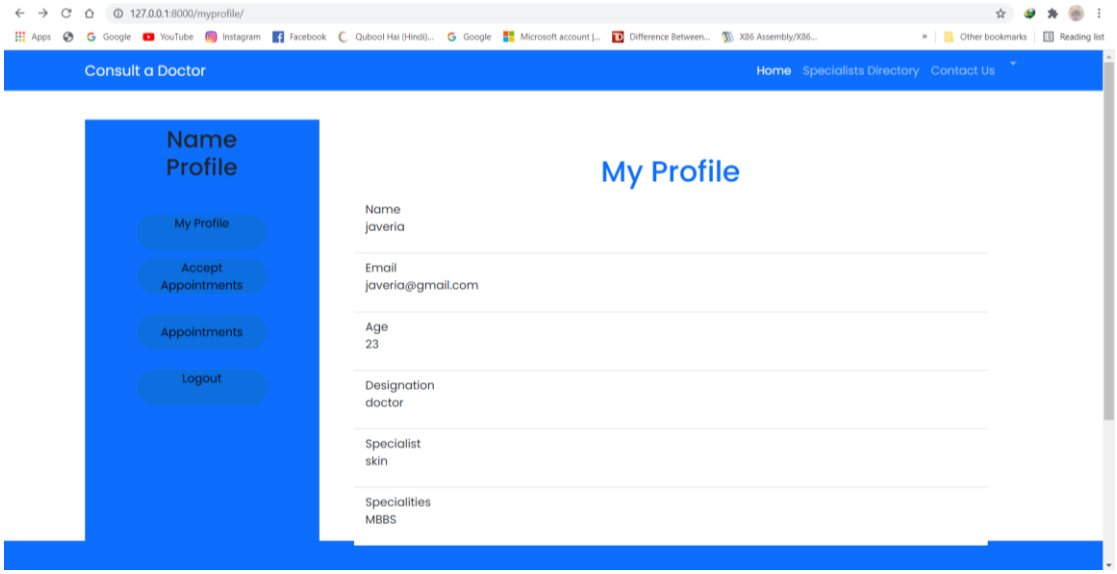
6.8. Patient's Profile:



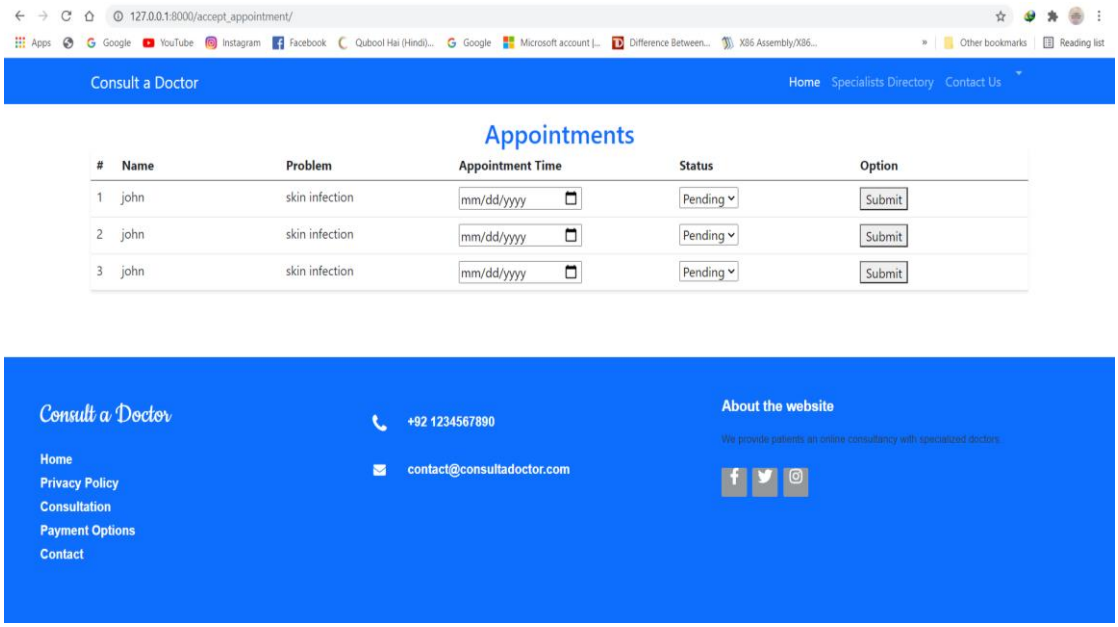
6.9. List of requested appointment by Patient:



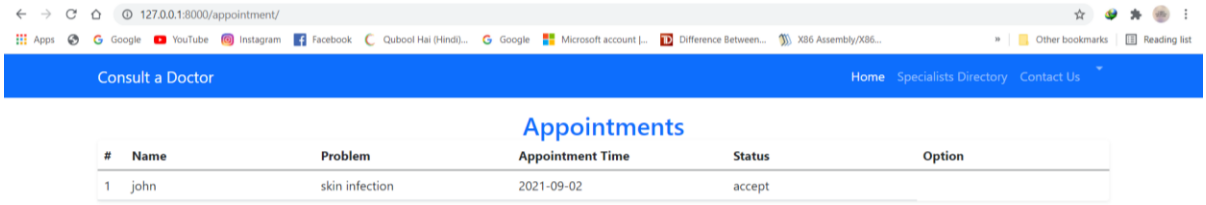
6.10. Doctor's Profile:



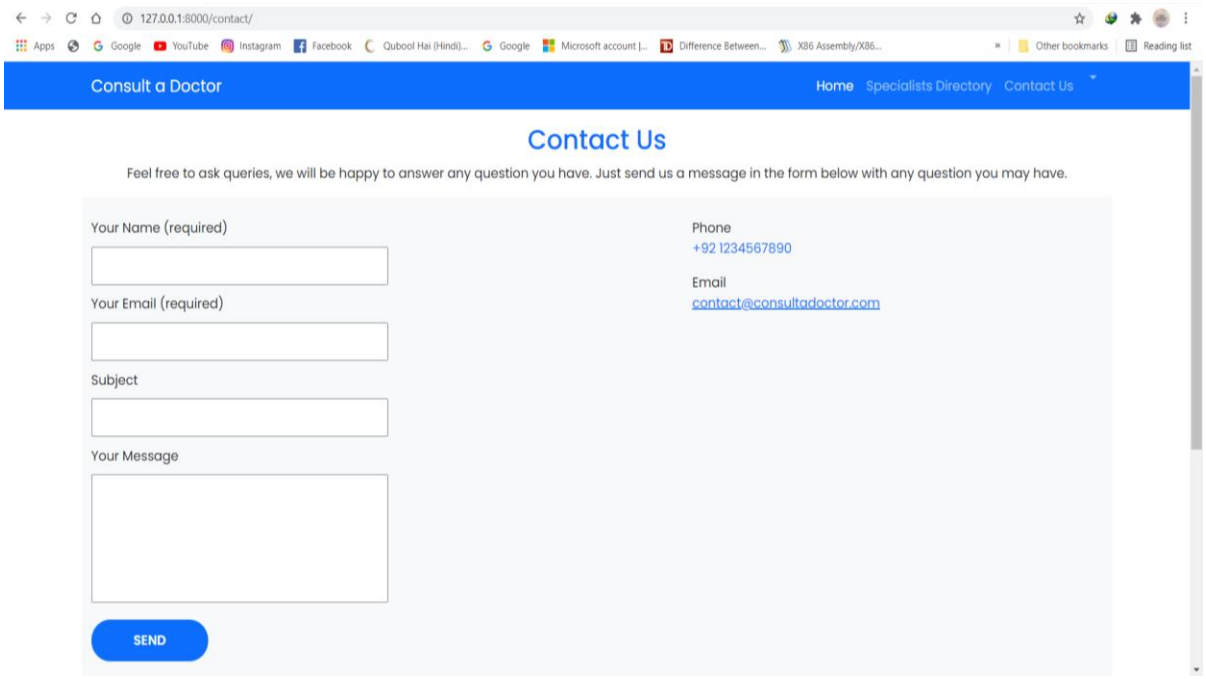
6.11. Doctor side view of appointments requested by patients:



6.12. List of appointment which are responded by doctor:

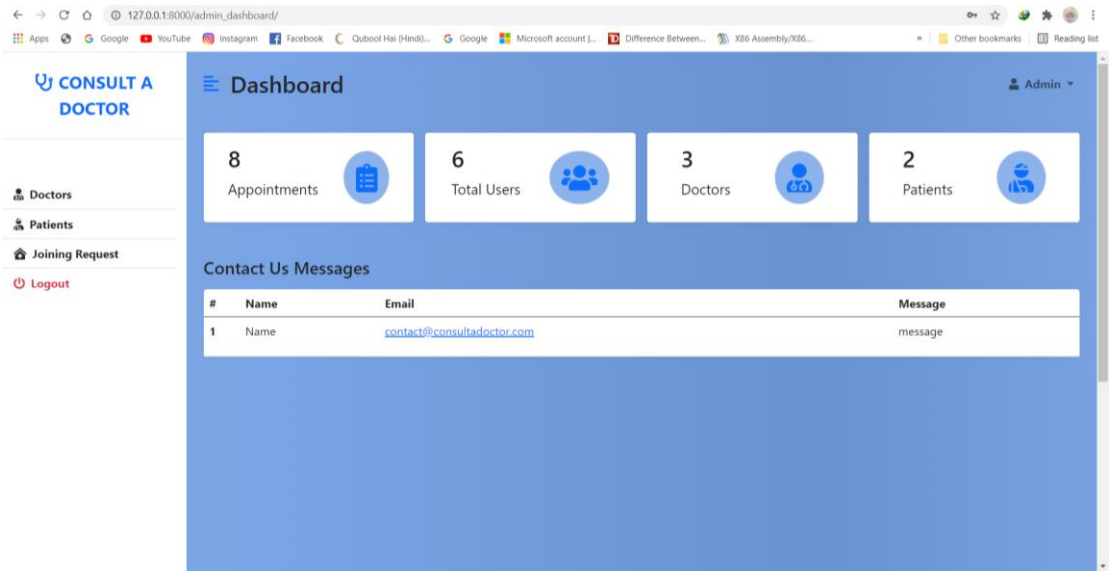


6.13. Contact Us:

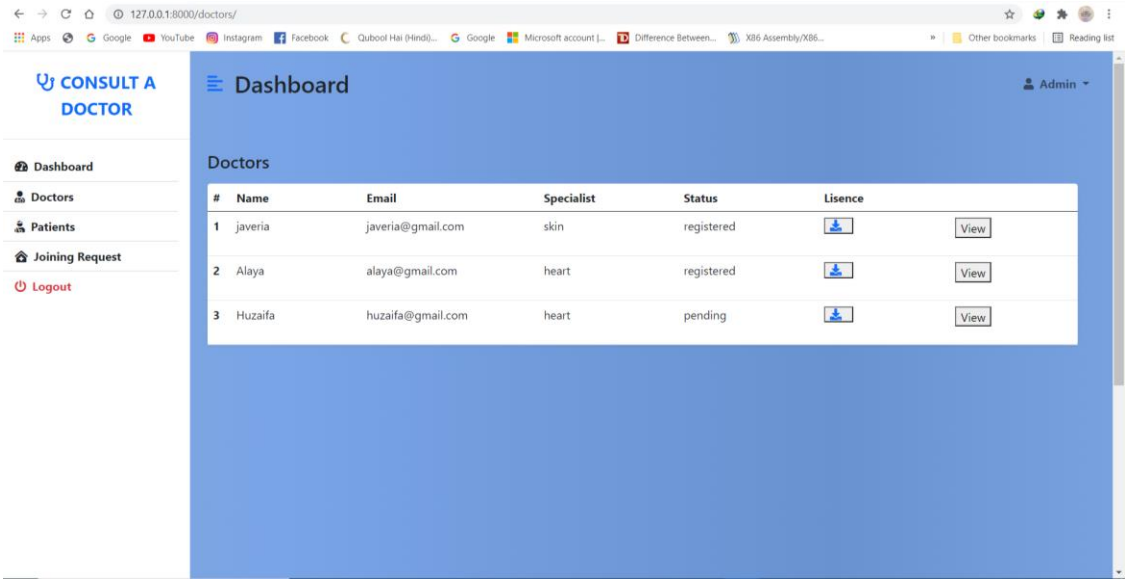


6.14. Admin Dashboard

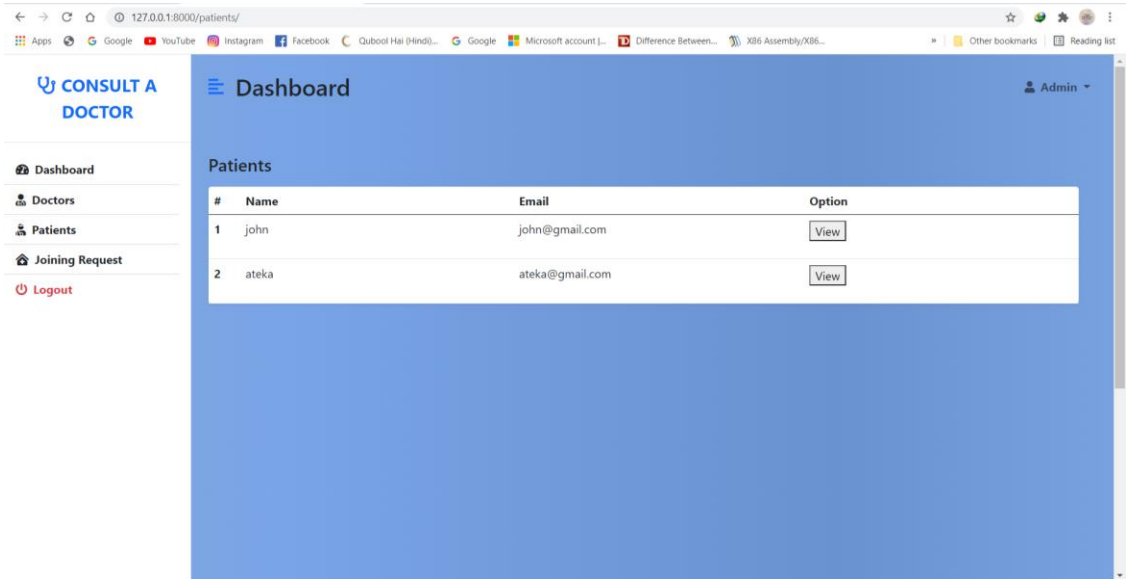
6.14.1. Dashboard:



6.14.2 Registered Doctors:



6.14.3. Registered Patients:



6.14.4 Registration Request by Doctor: