FINAL SEMESTER ASSESSMENT (FSA)
B.TECH. (CSE)
VI SEMESTER

UE18CS355 – OBJECT ORIENTED ANALYSIS AND DESIGN WITH SOFTWARE ENGINEERING LABORATORY

PROJECT REPORT

ON

Online Consultation and Appointment system

SUBMITTED BY

NAME	SRN
1) Pavan Kumar Desai	PES1201801539
2) Ashay G	PES1201801767
3) Girish G N	PES1201802101

JANUARY – MAY 2021

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
RR CAMPUS,
BENGALURU – 560100, KARNATAKA, INDIA

TABLE OF CONTENTS

ABSTRACT

The Purpose of Online Consultation and appointment system is to automate the existing manual system by the computerized equipment and full-fledged computer software, fulfilling their requirements, so that their data can be stored for a longer period with easy accessing and manipulation of the same. The main objective of the project is to build an application program to reduce the manual work of managing the details of Doctor, Appointment, Patient.

The Online Consultation and appointment system provides a platform for various citizens/patients for getting appointments with the selected doctor. Here we propose a system that connects patients to available doctors for online consultation. Our proposed system aims to build an environment where various patients needing doctor help consult doctors, chat with doctors, tell them their issues and discuss remedies. It also consists of a doctors login panel where doctors may login to the system and then see patient requests for consultations. The system then schedules those requests and serves them to doctor one after another. Overall, the system is designed for both patients and doctors, where a patient can book,modify and cancel appointments according to his/her schedule and a doctor can view all the patients allocated and treat them accordingly.

Software Requirements Specification

for

ONLINE CONSULTATIONS AND APPOINTMENTS(PRACTO)

Version 1.0 approved

Prepared by

S Vinamra - PES1201801463

Nanditha - PES1201801536

Rallapalli Aishwarya - PES1201800309

PES University

6th February 2021

Table of Contents

Revision History

Name	***Date***	***Reason For Changes***	***Version***

1. Introduction

1.1 Purpose

The purpose of this document is to provide a correct and complete description of the requirements for the software of an online consultation appointments application similar to PRACTO.
The requirements will be shown in the written description to explain various concepts and different types of functionalities with relevant information.
A patient, his/her representatives or affiliates, searching for practitioners through the Website

1.2 Intended Audience

A growing demographic for online consultations are young people – teenagers and young adults. There are several reasons for this, but an important factor is that young people are met in their own arena – the digital one
The targeted users range from the population having internet access and are familiar with various online services available. This group of targeted users are eager to ease out their daily life through online facilities provided by various online organizations.
The developer has targeted 4 types of users:

- *General Users who are seeking online service to avoid the chaos of traditional appointment booking systems.*
- *Physicians working with large scale medical organizations who attend a large number of patients every day.*
- *Physicians running independent small scale clinics in suburban areas.*
- *Large scale healthcare institutions who are looking for simple cost-effective appointment scheduling.*

1.3 Product Scope

This software is responsible for making the people to make better health care decisions and health related issues .This also helps people to find better suggestions regarding their health and medications by consulting doctors online and also they can choose their practitioner on their own in a website they can also consult them at any time if they appointment. People start finding the help from the best doctors which can be managed by a single health account for the entire family .Tt also secures our information and all the details that are entered by the people. This simple looking software allows the people to take their health care decisions.with search we provide you to find and consult the best doctors across the website Everyday millions and billions of cases were registered on health issues due to this the people were struggling to get treated in hospitals so we are here to help them our main motto is to help mankind to live healthier and longer

1.4 References

I. *Practo Online Consultation application,*
II. *https://www.c-sharpcorner.com/article/software-requirements-specification/,*
III. *http://www.se.rit.edu/~foryourhealth/srs.pdf*

IV. Kumar, Amit & Kumar, Amit. (2014). Practo Technologies: the online way of life!. Emerald Emerging Markets Case Studies. 3. 1-19. 10.1108/EEMCS-06-2013-0127.

2. Overall Description

2.1 Product Perspective

The software will help the users in viewing the details of nearest clinics. The location issue, this shows the exact location of the clinics so that the user can reach without any panic. The user can solve their health related issues by questioning the doctors and getting the best solution.It helps you connect to a doctor in just 60 seconds. All you have to do is tell us your symptoms or health problems, choose the speciality, make a payment. Once payment is made, we alert our panel of verified, high-quality doctors and allocate a doctor to your consultation.
Practo is a healthcare app that enables users to find medical practices and book an instant appointment and also do an online consultation with doctors. Also, users can post medical questions and get answers from the experts available. The application also allows users to take private online consultation sessions with the doctor of their choice.
Other than that, the application allows users to view previously booked appointments and save their appointments and their favorite doctors. Furthermore, the application has a navigation feature which will help users to find the practices' locations. The application also allow user to choose a convenient time slot for them to make an appointment

2.2 Product Functions

This product supports the following functions -

- ❏ *Login and signup.*
- ❏ *Searching for doctors,clinics,hospitals.*
- ❏ *Booking an appointment.*
- ❏ *Consultation with doctors.*
- ❏ *Pharmacy (medicines and health products).*
- ❏ *Diagnostics(tests and checkup).*
- ❏ *Showing patient reviews of the doctors*
- ❏ *Users can upload their medical documents for future reference.*

2.3 User Classes and Characteristics

- ➔ *.IT Personnel:*
    - ❏ *Technical Skill : Has full understanding of the system, and working knowledge of SQL*
    - ❏ *Frequency of Use : Low*
    - ❏ *Education Level : At least formal training or some college*
    - ❏ *Privileges : Administrator privileges has access to all parts of the system and can directly access the database.*
    - ❏ *Experience with System : High*
    - ❏ *Product Functions: Run database queries, add new forms*

➔ *Clinician*

- ❏ *Technical Skill: Low to medium*
- ❏ *Frequency of Use: Multiple times per day*
- ❏ *Education Level: Nonspecific*
- ❏ *Privileges: Read access to all documents, write access only upon creation or special cases.*
- ❏ *Experience with System: Medium*
- ❏ *Product Functions: View and interpret data and documents, create and add new documents.*

➔ *Patient*

- ❏ *Technical Skill: Low*
- ❏ *Frequency of Use: Multiple times per day*
- ❏ *Education Level: Not*
- ❏ *Specified Privileges: Session/Daily attendance through fingerprint scanner*
- ❏ *Experience with System: Low*
- ❏ *Product Functions: Check in to session or day through fingerprint scanner.*

2.4 Operating Environment

➔ *Database*

The database will exist on a Microsoft Windows Server That runs MSSQL 2008.

➔ *Application*

The main user application will be built as a web based system. Users will access it through a browser and login window.

2.5 Design and Implementation Constraints

- ➢ *Our software product does not include multiple servers so we won't be able to handle large traffic.*
- ➢ *Language is also a constraint as it is the only language available*
- ➢ *Only HTTP will be used for communication .*
- ➢ *Only registered users will be allowed to make appointments .*

◆ *2.6 Assumptions and Dependencies*

➔ *Assumptions*

- ◆ *All users know English*
- ◆ *The documents uploaded by doctors are certified.*
- ◆ *First come first serve basis priority for booking of appointments*
- ◆ *The user is capable of using internet browsers with stable internet connection and is familiar with the usage of a computer or pc*

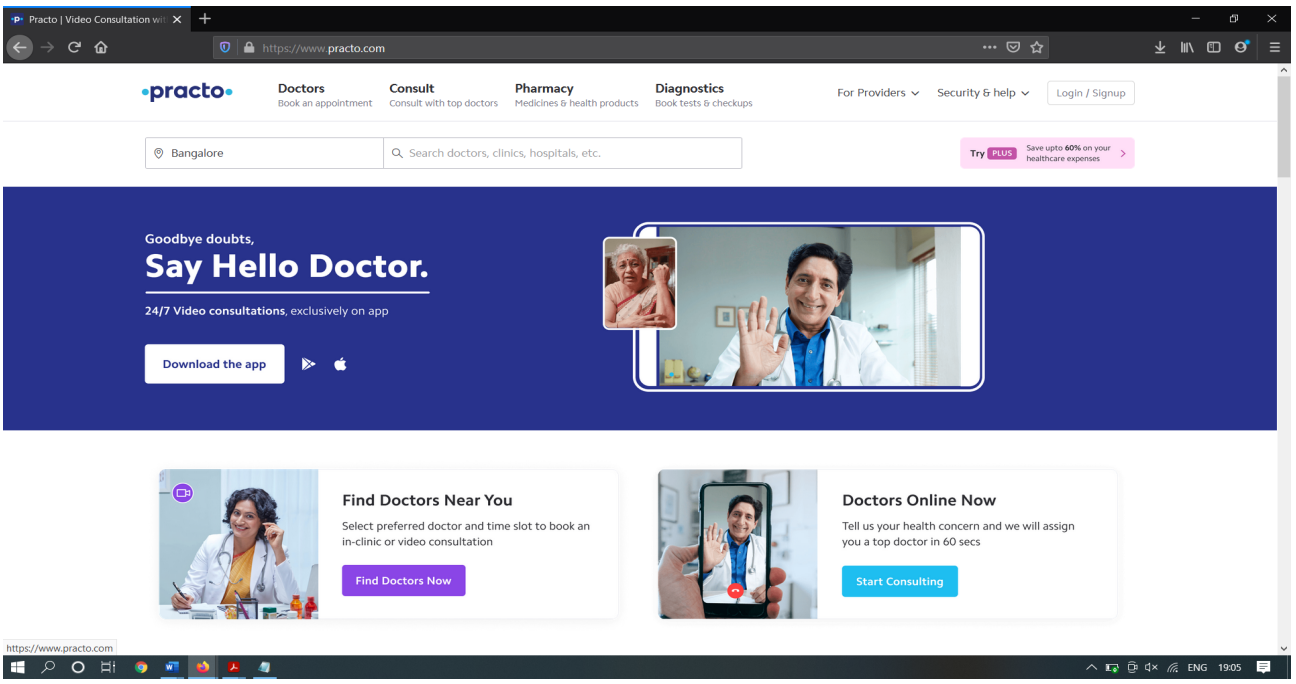
➔ *Dependencies*

- ◆ *The user is using their laptop or computer for accessing the website.*

3. External Interface Requirements

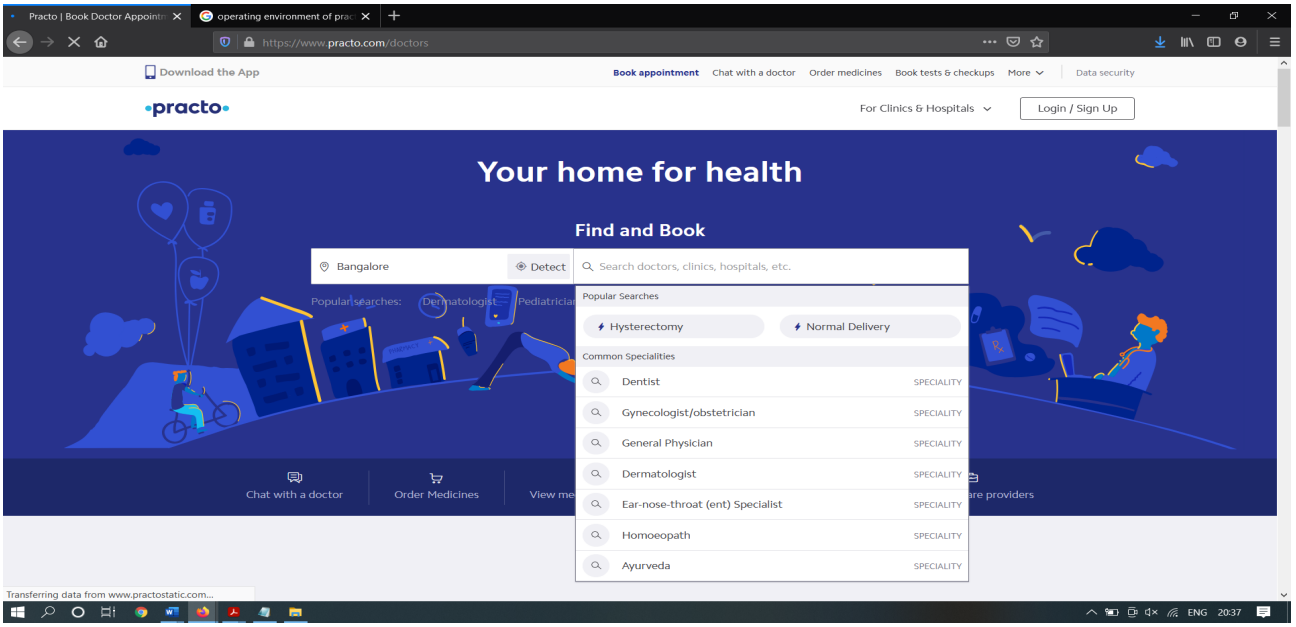
3.1 User Interfaces

- *Home page*



Here we can observe the Main menu of the website with its Logo. This is a Main Menu that Guest users (not logged in users) see when they access a webpage. On top right, we can see the login/sign up button which leads us to the login/sign up page.

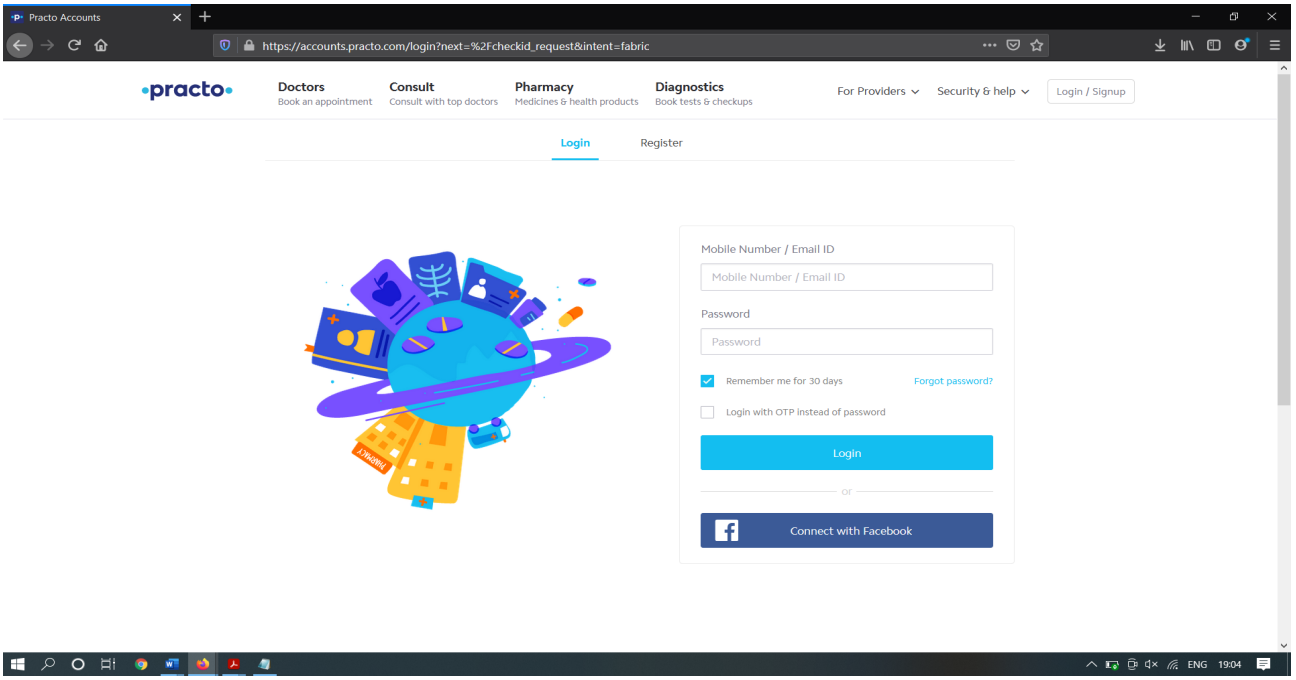
- *Search for Doctors*



Here is the screenshot of the Doctor search form with all the different criterions that can be used to search for a perfect doctor match. If a patient already knows the name of the doctor they are searching for, they could just add his first and/or last names. Otherwise, a patient could search a doctor by his/her gender (some might prefer female doctors other males and vice-versa)

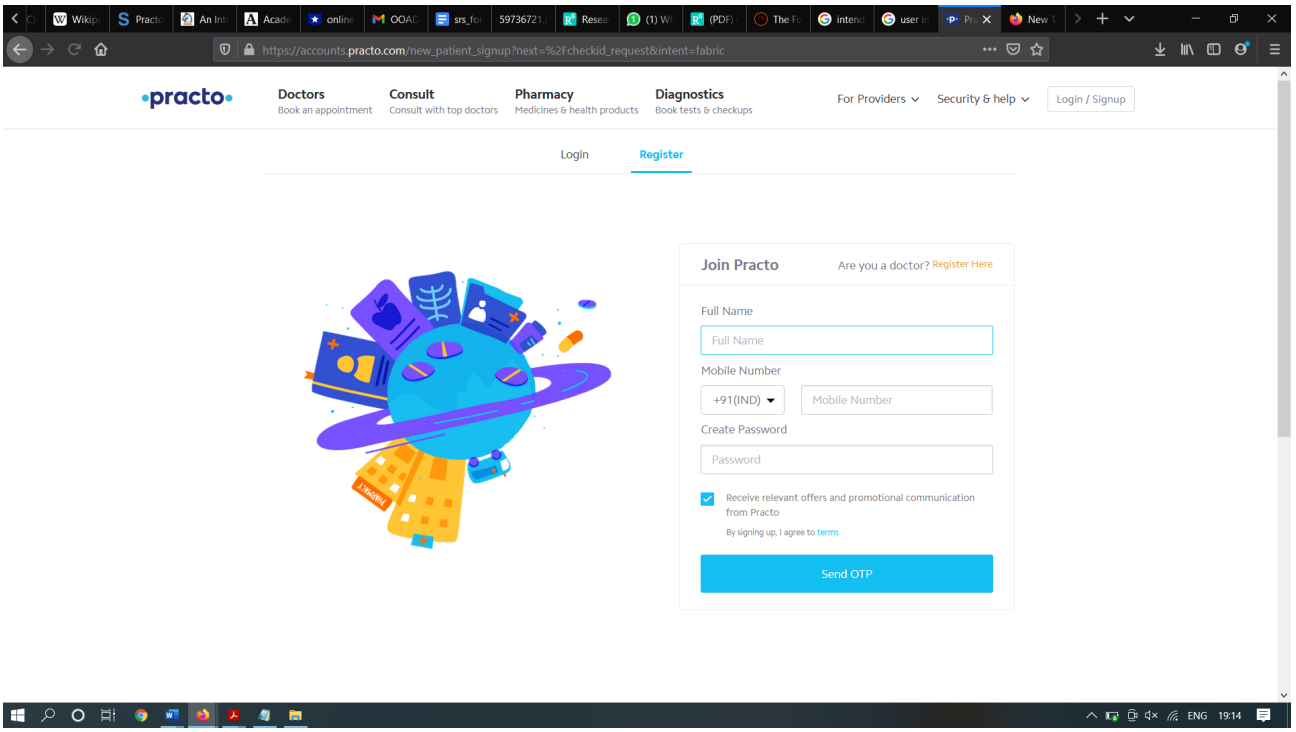
- *Register and login forms*
  - ➢ *Login form*

In order to login, a user will have to access a Login page and fill their username and password.After pressing the Login button, if everything was correct, the patient will be logged in and could start using services for registered patients.



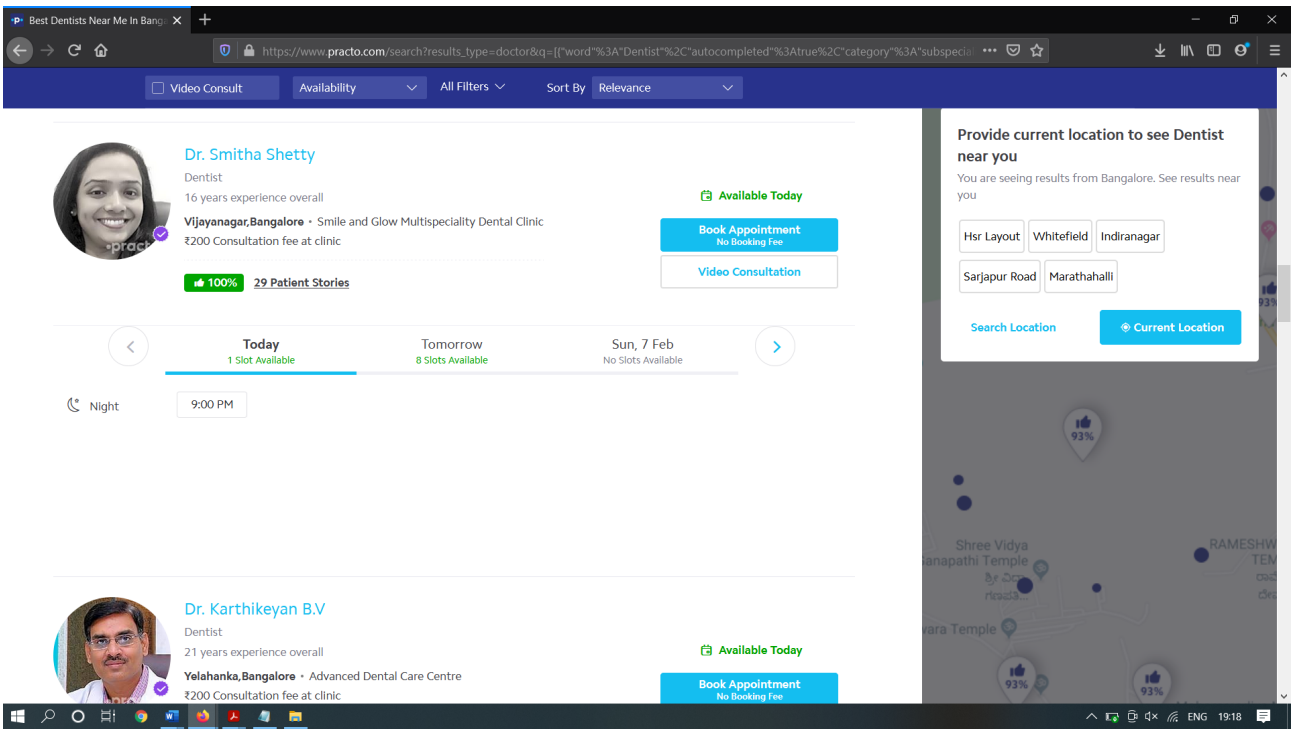
➢ *Register form*

A guest user is able to register on a website at any time. By accessing the Register page, he will be redirected to the following register form.

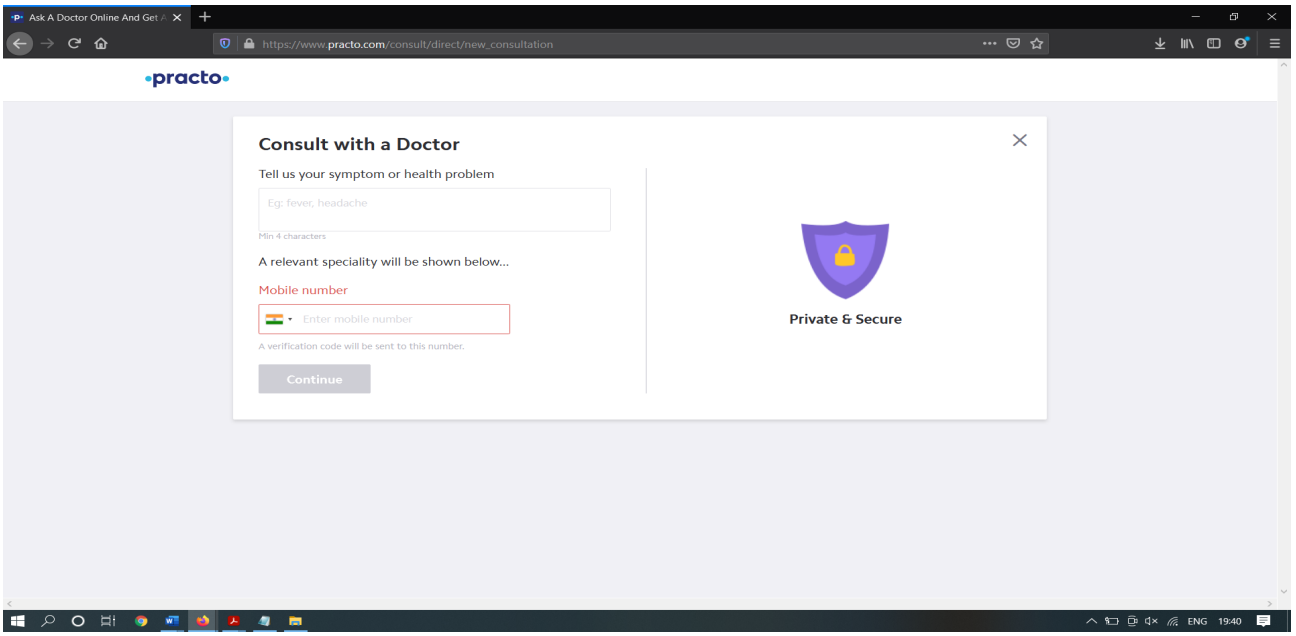


- *Make appointment*

  Choose your Appointment Date & Time and Book instantly!



- *Consult with a doctor*



3.2 *Software Interfaces*

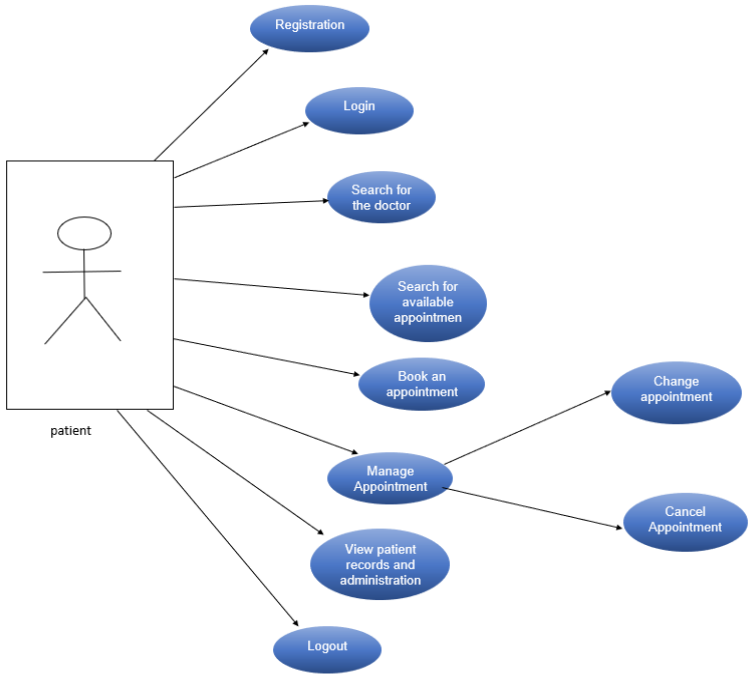
- ➔ *Database:MySQL database(distributed)*
- ➔ *Operating System:Windows*

- ➔ *Browser:The system will be intended to run on Firefox 4 and above,Google Chrome 10 and*
  above and Internet Explorer 8 and above which supports CGI, HTML & Javascript.
- ➔ *client server architecture*
- ➔ *platform: Java/PHP*

3.3 Communications Interfaces

Clients on the internet will be using HTTP/HTTPs protocols.

4. Analysis Models



use case diagram of patient

Use-Case 1: Registration
Primary-actor: *generic user/patient.*
Description: *To make an appointment, user registration is required.*
Precondition: *mobile Number and valid email address.*
Basic use-case flow: *For the registration process, the user/patient needs to give some information by filling the form. Some of the required information includes the following:*
Full Name
Email address
Mobile Number
Main scenario: *The user/patient will go onto the patient sign-up button either from the main page or from the drop-down menu. After clicking the sign-up button, a registration form will appear, where the user must give his personal information i.e., name, email address, mobile number etc. After giving the required information, the user will submit the form. If all the fields are filled including the valid email address the user will be registered onto the system. In case of any missing entry or invalid format of an email the error occurs onto the page.*

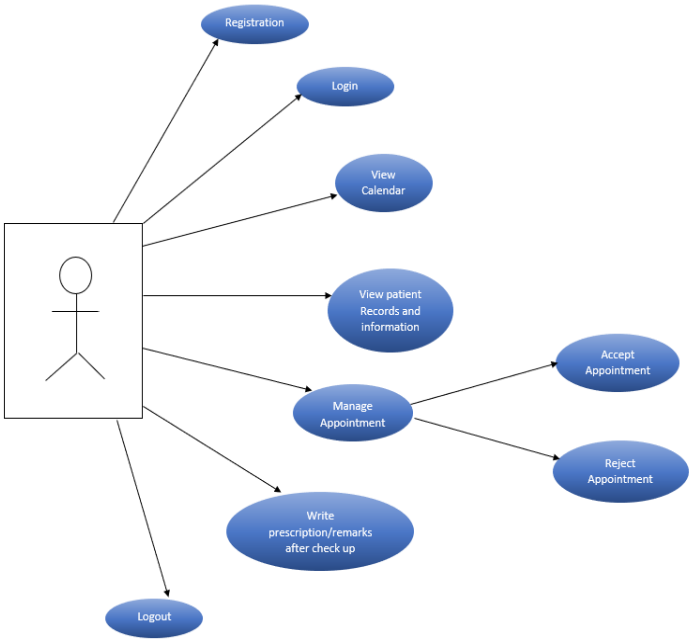
Use-case 2: Login
Primary-actor: *Patient/user.*
Description: *Before taking any appointment or get access to his medical record, the user must have to provide his mobile number or email address and password.*
Precondition: *the user must have a valid mobile number or email address and password.*
Basic use-case flow: *a valid mobile number or email address with a password must be entered by the user.*
Main scenario: *To be able to get into the system, the user needs to enter his mobile number or email address and password either from the main page or from the drop-down menu from the top of the page. After clicking onto the login button, an authentication request will be forwarded to the system.*

Use-case 3: search for a doctor
Primary-actor: *User patient/generic user*
Description: *In this use-case, any registered or unregistered user, can look and search for a doctor of any desired specialty.*
Precondition: *this use-case has no exception.*
Basic use-case flow: *By going into the category of any specific specialty, the user can view the list of all registered doctors.*
Main scenario: *The user will go to the down menu at the top of the page. A list of all specialties will appear in a drop-down list. The user will hit on the desired specialty. After that, a page will open with all doctors of that specific specialty. Now the user has the choice to choose any doctor based on qualification, experience, location etc.*

Use-case 4: Book Appointment
Primary-actor: *User/patient*
Description: *After choosing a doctor user will go further to send an appointment request from the available timings.*
Precondition: *the user must be login*
Basic use-case flow: *The patient/user hits the button for booking an appointment for the doctor. A list of available timings will appear for the chosen date. The user will select the suitable time for him and send the request for approval.*
Main scenario: *The user will hit the button for "Book appointment". List of available timings will appear for a chosen date. The user selects the suitable time. The user will hit the submit button to send the request for approval.*

Use-case 5: View medical Records
Primary-actor: *User/patient*
Description: *the user can view his medical records.*
Precondition: *the user must be signed in.*
Basic use-case flow: *The user/patient will click on the name of the patient and it opens the patient's profile. By going into it the user can view the history by clicking on the button for the patient's medical records.*
Main scenario: *The user clicks on the name of the patient and then presses the button "my medical records" to view old reports.*

Use-case 6: Log out
Primary-actor: *User/patient*
Description: *the user will log out from the system.*
Precondition: *the user must be logged in*
Basic use-case flow: *the user can sign out himself from the system.*
Main scenario: *the user clicks on the log out button. The system will bring the user to the main page for the generic user.*



user case diagram for doctor

Use-case 1: Registration
Primary-actor: *Generic user/doctor*
Description: *To get online appointment requests, the doctor must register himself as a user on the application/system.*
Precondition: *Mobile Number and active email address.*
Basic use-case flow: *To register as a doctor, the user must fill the form. Required fields include the following information:*
Full name
Email address
Qualification
Experience
Valid PMDC certificate (Practicing license issued by the PMDC authority)
Clinic address
Clinic timings
Main scenario: *The user/ doctor has to go onto the doctor's sign-up button either from the drop-down menu at the top or from the main page. After clicking the sign-up button, a registration form will appear, where the doctor/user has to give his personal as well as professional information i.e., name, clinic address, qualification, working experience etc. The user must give a valid PMDC certificate for the registration process. After giving the required information the user/ doctor will submit the form. If all the fields are filled, a request for registration will be sent to the admin. As soon as the admin receives a request for a doctor's registration, he will verify his documents and only then the user will be accepted /registered. In case of any missing entry or invalid format of email, an error occurs onto the page. A PMDC certificate is a license or a proof that an individual is allowed by the authority to practice medicine. Any user who fails to give a valid PMDC certificate issued by PMDC authority will not be allowed to register as a doctor. A PMDC certificate is important to avoid any scam.*

Use-case 2: log in
Primary-actor: *User/doctor*
Description: *For further functions, the user must have to provide his mobile number or email address and password.*
Precondition: *the user must enter the mobile number or email address and password.*
Basic use-case flow: *user/doctor should provide his username and password to log in.*

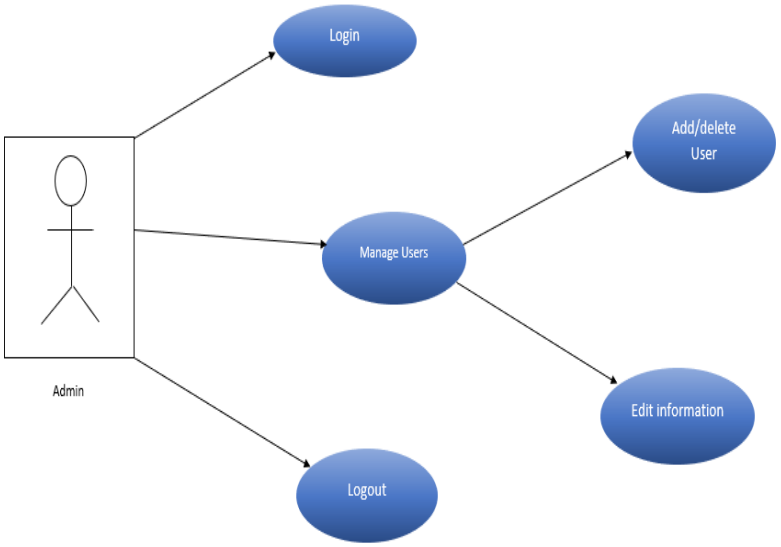
Main scenario: *To be able to get into the system, the user needs to enter his username and password either from the main page or from the drop-down menu from the top of the page. After clicking onto the login button, an authentication request is forwarded to the system.*

Use-case 3: View calendar
Primary-actor: *User/doctor*
Description: *User would be able to view his calendar.*
Precondition: *The user must be signed in.*
Basic use-case flow: *After logging in, the user/patient selects the date from the calendar to filter out the appointment.*
Main scenario: *After signing in, the user/doctor can view the page of his calendar. The user will select the month and date to see the appointments of that period or date. The user can view all the patient's appointments of any date.*

Use-case 4: Accept or reject a request
Primary-actor: *User/doctor*
Description: *User/doctor can accept or reject any patient's request.*
Precondition: *User/doctor must be signed in.*
Basic use-case flow: *User/doctor selects any date and decides to accept or reject any appointment request.*
Main scenario: *After logging in, all the patient's appointment requests will be appearing on the calendar according to the dates. User /Doctor will select the date from the calendar. User /Doctor can view all the appointment requests. User /Doctor can select the option to accept or reject the appointment request from the drop-down menu.*

Use-case 5: View patient
Primary-actor: *User/doctor*
Description: *User/doctor would be able to view the patient's detailed information including his/ her medical history.*
Precondition: *User/doctor must be signed in*
Basic use-case flow: *After selecting the date, user/doctor selects the patient. User/doctor can view the patient's details and medical history.*
Main scenario: *After selecting the date, a list of patients of that particular date will appear. User /Doctor will select the patient. User (doctor) can view his all information including the medical history by clicking on the "view patient" button*

Use-case 6: log out
Primary-actor: *User/doctor*
Description: *the user/doctor will log out from the system.*
Precondition: *the user/doctor must be signed in*
Basic use-case flow: *the user/doctor can sign out himself from the system.*
Main scenario: *the user/doctor clicks on the log out button. The system will bring the user to the main page for the generic user.*



use case for Admin

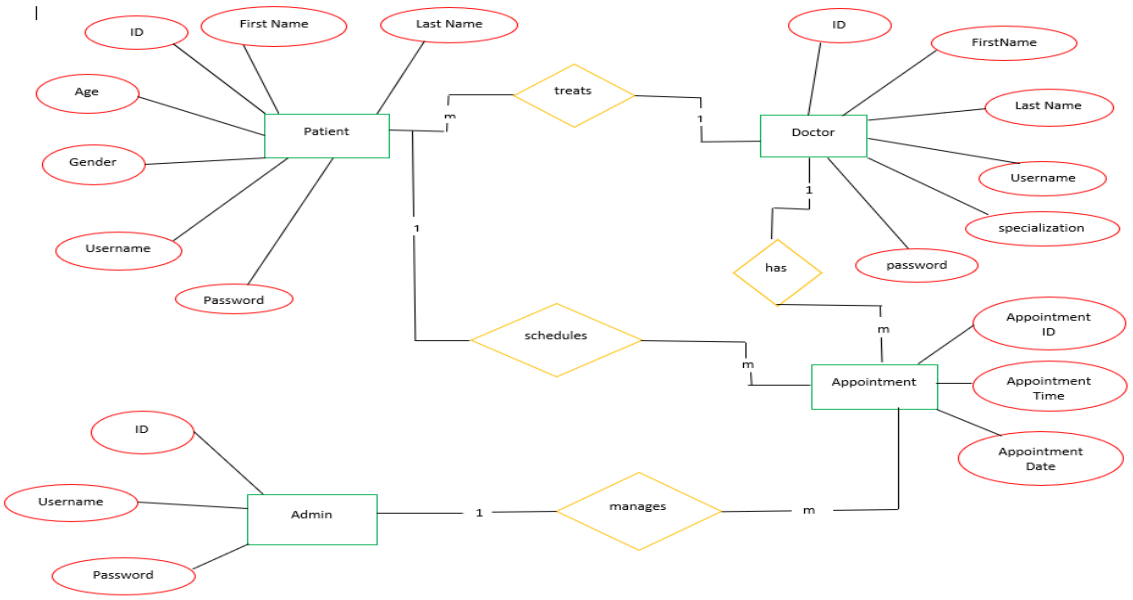
Use-case 1: Login
Primary-actor: *Admin/administrator*
Description*: A user, who possesses admin username and password, owns administrative rights.*
Precondition*: username and password.*
Basic use-case flow: *Admin/user needs to provide the username and a password.*
Main scenario: *Admin/user enters valid username and password. After verification, the user will be logged in as an admin and can use all the administrative rights.*

Use-case 2: Manage users
Primary-actor: *Admin/user*
Description: *User/admin can use his administrative rights.*
Precondition: *the user must be logged in as an admin.*
Basic use-case flow: *Admin/user can view, edit or delete any user information.*
Main scenario: *Admin can view, edit or delete any user or any information related to user doctor or user-patient.*

Use-case 3: log out
Primary-actor: *Admin/user*
Description: *Admin will log out from the system.*
Precondition: *Admin must be signed in*
Basic use-case flow: *User signs out himself “as an admin".*

Main scenario: *User clicks onto the logout button. The system will remove information from the local storage and bring the user to the homepage.*

5. System Features



ER diagram of appointment System

5.1 Login(Priority : HIGH)

By whom: Patient,Doctor
Goal: To log into account.

5.1.1 Description and Priority

Given that a user is registered, the user should be able to log into the web portal The system should keep and maintain the user login information. The system displays the menu for the doctor or patient to enter username and password. The system should redirect users to the homepage.

5.1.2 Stimulus/Response Sequences

1. *Doctor patient opens website and clicks Log in*
2. *System shows user a menu to enter username and password*
3. *User enters the information and clicks on the submit button.*
4. *System displays "login successful*

Flow of Events for Extensions (Alternate Scenarios)

1. *user enter invalid username or password*
2. *System shows a warning message that username or password entered was wrong and asked for entering information again,*

5.1.3 Functional Requirements

REQ-1: *The system should enable patients and doctors to log in .*
REQ-2: *The system should enable patients and doctors to log out .*
REQ-3: *The system should enable patients to schedule appointments.*
REQ-4: *The system should enable patients to modify and delete appointments.*

5.2 Register (Priority : HIGH)

By whom: Patient,Doctor
Goal: To Create an account that can book appointments for patients and for doctors to be able to confirm and manage appointments.
Preconditions :The system displays the menu for doctor or patient to enter personal information.
Post conditions: The system should redirect users to the homepage.

5.2.1 Description and Priority

A user, as specified in carlier section must be able to register their credentials
They should provide their basic information about themichves for eg name, address

,email,phone number.

5.2.2 Stimulus/Response Sequences

1. *Doctor/patient open website and clicks register to create an account.*
2. *System shows a user a page to get some basic information about the user such as name ,surname,user ID,date of birth ,username,password and etc.*
3. *User enters the information and clicks OK for the account to be made .*
4. *System displays "registration successful"*

Flow of Events for Extensions (Alternate Scenarios)

1. *System shows a warning message that the username or password entered was wrong and asks for entering information again.*

5.2.3 Functional Requirements

REQ-1: *The system should enable patients and doctors to register.*
REQ-2: *The system should enable patients to schedule appointments.*
REQ-3: *The system should enable patients to modify and delete appointments.*
REQ-4: *The system should allow Patients and doctors to view and modify their profile.*
REQ-5: *The system should allow Doctors to set their available time.*

5.3 Appointment Reservation(Priority : HIGH)

5.3.1 Description and Priority

In this case the initiating Actor is the Patient .The patient's Goal is to make online doctor appointment.The precondition here is that the user must create his account and logged in and he should enter the problem regarding which he is wanting to see the

doctor .The system then displays the doctor's schedule and the patient can select the most suitable time.

5.1.2 Stimulus/Response Sequences

1. *Patient requests a reservation by clicking on book appointment button*
2. *System shows patient a menu to choose department and location*
3. *Patient enters information*
4. *System shows a list of doctors that suit the criteria.*
5. *Patient selects doctor.*
6. *System display's doctor's schedule,*
7. *Patients select preferred time and day. 8 System displays appointments booked.*

Flow of Events for Extensions (Alternate Scenarios)

1. *User clicks on the "cancel reservation" button.*
2. *System shows a message and cancellation is done.*

5.3.3 Functional Requirements

REQ-I. The system should enable patients and doctors to log in.

REQ-2. The system should enable patients and doctors to register

REQ-3 The system should allow Patients to book appointments.

REQ-4 The system should allow Patients to modify or cancel their appointment

REQ-5. The system should allow administrators to delete past appointments.

5.4 View and Modify Profile (Priority : HIGH)

5.4.1 Description and Priority

By whom :Patient, Doctor
Goal :I To change personal information
Preconditions: The user must log in to the system
Post conditions The system should display a verification message to ensure information

5.4.2 Stimulus Response Sequences

1. *User click on account page*
2. *System displays user's personal account page*
3. *User selects option edit info*
4. *The system displays form to user*
5. *User enters new information on the made changes by click the Save button*
6. *System displays account updates.*

5.4.3 Functional Requirements

REQ-1 -The system should enable patients and doctors to log in .

REQ-2:The system should enable patients and doctors to register.

REQ-3: The system should enable patients and doctors to view and modify their profile

5.5 Set available Time (Priority : HIGH)

5.5.1 Description and Priority

By whom: Doctor

Goal: To set available time slot in schedule updated
Preconditions User must create an account and logged in
Post conditions: The system should display doctor's new schedule

5.5.2 Stimulus Response Sequences

1. *Doctor clicks on the account button.*
2. *System shows account page*
3. *Doctor selects a free time slot on displayed schedule*
4. *Updates schedule displays new available time slot in schedule.*

5.5.3 Functional Requirements

REQ-1- The system should enable patients and doctors to log in
REQ-2- The system should enable patients and doctors to register
REQ-3- The system should allow Doctors to set their available time.

5.6 Search Doctor(Priority : HIGH)

5.6.1 Description and Priority
By whom: Patient Doctor
Goal: To search for a particular doctor

Preconditions: User opens website
Post conditions. The system should display doctor's information

5.6.2 Stimulus Response Sequences
1. *User clicks on the search button.*
2. *System display search form.*
3. *User enters doctors name and clicks search*
4. *System duplays doctor's information*

Flow of Events for Extensions(Alternate Scenarios) Doctor not in system.
1. *System displays dat doesn't exist*

5.6.3 Functional Requirements

REQ-1. The system should allow Patients to search for available doctors.

5.7 Cancel Appointment (Priority : HIGH)

5.7.1 Description and Priority
By whom: Patient Doctor
Goal: To cancel the previously booked appointment

Preconditions: User opens website
Post conditions. The system should display a "cancelled appointment" message.

5.7.2 Stimulus Response Sequences

1. *Users click on the appointment slot on the calendar.*
2. *System displays confirmation box are you sure you want to cancel appointment*
3. *User selects YES.*
4. *System deletes appointment and displays a message appointment cancelled",*

Flow of Events for Extensions (Alternate Scenarios):User doesn't want to cancel anymore
1. *User selects NO on confirmation box*
2. *System displays initial appointment schedule.*

5.7.3 Functional Requirements
REQ-1- The system should enable patients and doctors to log in

5.8 Manage Users (Priority : HIGH)

5.8.1 *Description and Priority*
1. *By whom: database manager*
2. *Goal: to add user to system*

Preconditions: User must login to the system
Post conditions. The system should display a message to say that the user has been added..

5.8.2 Stimulus Response Sequences
1. *User clicks on the doctor patient page.*
2. *System displays a list of doctor patients.*
3. *User select option add new doctor patient*
4. *System displays form.*
5. *User enters information and clicks on add button*
6. *System displays a message to show that a user has been added.*

5.8.3 Functional Requirements

REQ-1- The system should enable patients and doctors to log in
REQ-2- The system should enable administrator to log in
REQ-3. The system should allow administrator to manage users

5.9 Manage and access database (Priority : HIGH)

5.9.1 *Description and Priority*
1. *By whom: database manager*
2. *Goal: to manage data,add,delete,edit,update data*
Preconditions: User must login to the system
Post conditions. The system should display a message to say that the modified data has been added to the database.

5.9.2 Stimulus Response Sequences

1. *User clicks on the search button.*
2. *The user then does his desired modifications and clicks apply.*
3. *System displays those doctors who follow the above constraints given by the patient.*

6. Consultation using video conference(Priority : HIGH)

6.1 Description and Priority

The appointment between the doctor and the patient is through a video conference where the doctor analyses the patient thoroughly and prescribes the best treatment. This helps the patient to take global high level and experienced advice from doctors all over the world without having to bear the expense of travelling to the place where the doctor resides.

6.2 Stimulus Response Sequences

1. *User scheduled an appointment with the doctor*
2. *The appointment happens in a very secure video conference platform.*

6.3 Functional Requirements

REQ-1 The system should allow Patients to book appointment
REQ-2 The system should allow Patients to modify or cancel the appointment

7. Other Nonfunctional Requirements

7.1 Performance Requirements

- *Static requirement*
  - ➢ *The System should be able to service multiple terminal connections simultaneously. At least 50 internet users should be able to use the system without any connection delay issues.*
  - ➢ *The system should process textual and numerical data.*
- *Dynamic requirement*
  - ➢ *The software should be able to process more that 90% of the user queries in a fraction of 1 second.*
  - ➢ *No restriction is applied on the number of queries performed by a user as long as he is authorized to perform them.*
  - ➢ *The time needed to display information of users' queries on a web client varies according to the system load and users' internet connection speed.*
  - ➢ *For a specific time on a specific date, all users have the same priority to reserve that slot. Users are served in a First-Come First-served fashion*

7.2 Safety Requirements

- *Databases should use sharding to be redundant to prevent loss of data.*
- *Backups of the databases should be done hourly and be kept for one week.*

7.3 Security Requirements

- *User authentication is needed to use the system. users should not be able to create or delete reservations for other users. Thus, a track of each user's reservations is needed in order to apply these constraints.*
- *Personal information collected upon registration should not be revealed for the sake of privacy. A user needs only to know whether a certain time slot is reserved or not, without the need to reveal the patient's name.*
- *Data has multiple encrypted backups*

7.4 Software Quality Attributes

- ***Accessibility:** Online scheduling software is accessible for all types of users. All that's required is an Internet connection and users can access a vast range of facilities.*
- ***Maintainability:***
  - ➢ *Back-Up: The system offers the efficiency for data back up.*
  - ➢ *Errors: The system will track every mistake as well as keep a log of it.*
- ***Usability:***
  - ➢ *Easy to use*
  - ➢ *Fast-loading pages*
  - ➢ *Consistent layout*
  - ➢ *Logical navigation*
  - ➢ *Cross-platform/browser compatibility*
  - ➢ *Screen Resolution*

7.5 Business Rules

The Services may change from time to time, at the sole discretion of Practo, and the Agreement will apply to your visit to and your use of the Website to avail the Service, as well as to all information provided by you on the Website at any given point in time.
The terms "personal information" and "sensitive personal data or information" are defined under the SPI Rules, and are reproduced in the Privacy Policy.
The User is responsible for maintaining the confidentiality of the User's account access information and password, if the User is registered on the Website. The User shall be responsible for all usage of the User's account and password, whether or not authorized by the User. The User shall immediately notify Practo of any actual or suspected unauthorized use of the User's account or password. Although Practo will not be liable for your losses caused by any unauthorized use of your account, you may be liable for the losses of Practo or such other parties as the case may be, due to any unauthorized use of your account.

Domain requirements -

1. *Doctors profile and contact details can be viewed by doctors and users*
2. *A medical Practitioner should have a minimum experience in the medical field for one year in a clinic or hospital.*

8. Other Requirements

Legal Requirement
You must be 18 years of age or older to register, use the Services, or visit or use the Website in any manner. By registering, visiting and using the Website or accepting this Agreement, you represent and warrant to Practo that you are 18 years of age or older, and that you have the

right, authority and capacity to use the Website and the Services available through the Website, and agree to and abide by this Agreement.

Appendix A: Glossary

- *ER-Entity relationship*
- *SQL-Structured Query LanguageSRS-Software requirement specification*

Appendix B: Field Layouts

An Excel sheet containing field layouts and properties/attributes and report requirements.

Sample sheet with information required to register a new patient and doctor

Field	***Length***	***Data Type***	***Description***	***Is Mandatory***
Full Name	***20***	*Alphabet*	*Full Name*	***Y***
Mobile Number	***10***	*Numeric*	*Mobile Number*	***Y***
Create Password	***7***	*AlphaNumeric*	*Create Password*	***Y***

Sample sheet with information required to login the patient and doctor

Field	***Length***	***Data Type***	***Description***	***Is Mandatory***
Mobile Number	***10***	*Numeric*	*Mobile Number*	***Y***
Password	***7***	*AlphaNumeric*	*Password*	***Y***

Sample Report Requirements: Include the fields to be included in the report

Registration Report	***Transaction Report***
Bank Account Number	*Transaction Reference Number*
IFSC Code	*Bank Account Number*
Bank Name	*IFSC Code*
Account Status	*Bank Name*
Account Type	*Customer Name*
Customer Name	*Card Number*
Card Number	*Debit Transaction Amount*
SI Start Date	*Transaction Date*
Status	*Status*
Remarks	*Debit Attempt Number*
	Remarks

Sample sheet with information required to register the patient

Field	*Length*	*Data Type*	*Description*	*Is Mandatory*
Patient Name	***30***	***String***		***Y***
Phone Number	***12***	***Numeric***		***Y***
Consulting Doctors name	***30***	***String***		***Y***
Medical History	***500***	***String***		***N***
Previous Consulted Doctor	***30***	***String***		***N***

Appendix C: Requirement Traceability Matrix

Sl. No	***Requirement ID***	***Brief Description of Requirement***	***Architecture Reference***	***Design Reference***	***Test Case ID***	***System Test Case ID***
1	*R1*	*The system should enable the patient and doctors to login*	*Fig2. System Architecture*	*Fig 1. Class Diagram and Fig3. State Diagram*	*UT-1,UT-2,UT-3*	*ST-1*
2	*R2*	*The system should enable patient to schedule the appointment*	*Fig2. System Architecture*	*Activity and sequence diagram of "Management of Appointment"*	*UT-4*	*ST-1*
3	*R3*	*The system should allow patients to make changes to appointment if necessary*	*Fig2. System Architecture*	*Activity and sequence diagram of "Management of Appointment"*	*UT-5*	*ST-1*
4	*R4*	*The system should be able to allow the doctors to view all appointments*	*Fig2. System Architecture*	*Activity and sequence diagram of "Management of Appointment"*	*UT-7*	*ST-1*

5	*R5*	*The system should allow patients to cancel their appointments*	*Fig2. System Architecture*	*Activity and sequence diagram of "Management of Appointment"*	*UT-6*	*ST-1*
6	*R6*	*The system should be able to allow the patients to view all appointments*	*Fig2. System Architecture*	*Activity and sequence diagram of "Management of Appointment"*	*UT-6*	*ST-1*
7	*R7*	*The system should be able send the email confirmation if patient opts for it*	*Fig2. System Architecture*	*Activity and Sequence Diagram of "Email Reminders"*	*UT-9,UT-10,UT-11*	*ST-2*
8	*R8*	*The system should be able to allow the patients to chat.*	*Fig2. System Architecture*	*Activity and sequence diagram of "Online chat system"*	*UT-15,UT-16,UT-17,UT-18*	*ST-3*

PROJECT PLAN

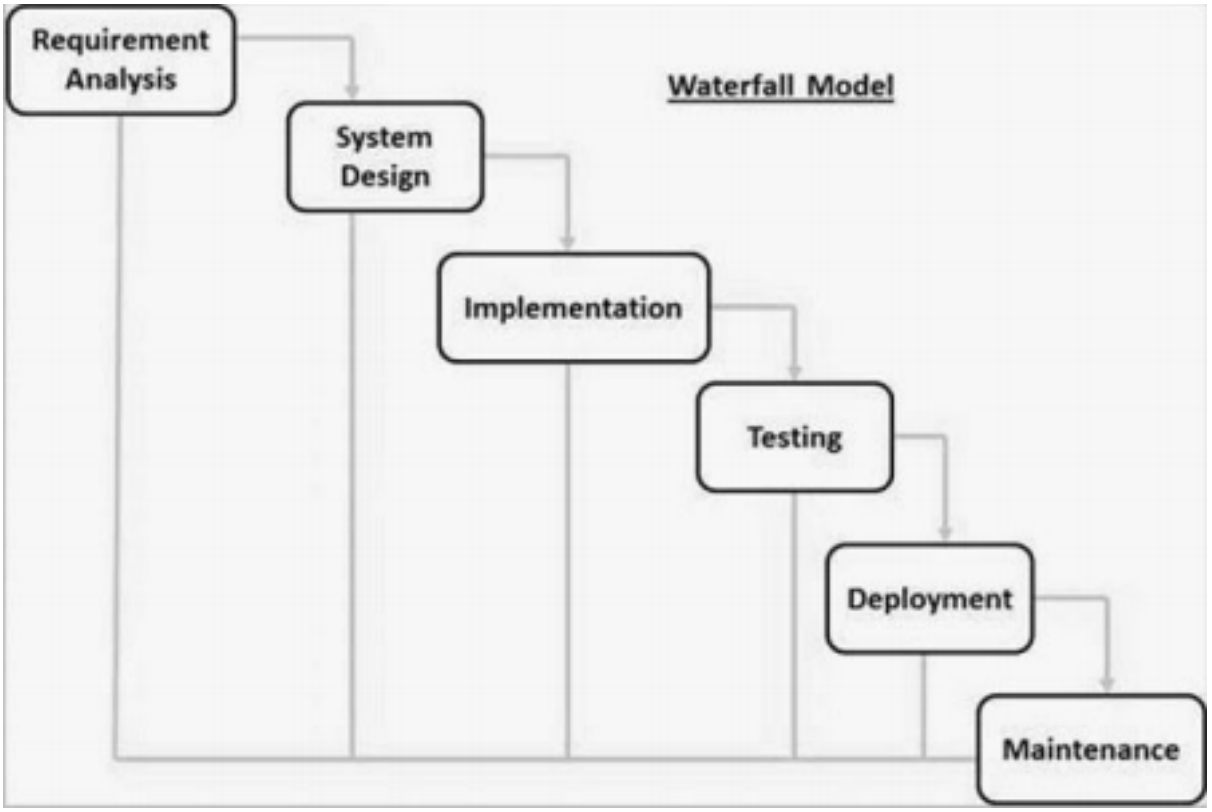
Identify the lifecycle to be followed for the execution of your project and justify why you have chosen the model.

Waterfall Model

We choose the waterfall model because all the requirements are known in advance, clear and not supposed to change in future and the problem statement is stable, technology is understood, no ambiguous requirements and also the project is short (4 months).

The Waterfall model is the earliest SDLC approach that was used for software development. The waterfall Model illustrates the software development process in a linear sequential flow. This means that any

phase in the development process begins only if the previous phase is complete.



The sequential phases in Waterfall model are −

- **Requirement Gathering and analysis** − All possible requirements of the system to be developed are captured in this phase and documented in a requirement specification document.
- **System Design** − The requirement specifications from the first phase are studied in this phase and the system design is prepared. This system design

helps in specifying hardware and system requirements and helps in defining the overall system architecture.

- **Implementation** − With inputs from the system design, the system is first developed in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality, which is referred to as Unit Testing.
- **Integration and Testing** − All the units developed in the implementation phase are integrated into a system after testing of

each unit. Post integration the entire system is tested for any faults and failures.

- **Deployment of system** − Once the functional and non-functional testing is done; the product is deployed in the customer environment or released into the market.
- **Maintenance** − There are some issues which come up in the client environment. To fix those issues, patches are released. Also to enhance the product some better versions are released. Maintenance is done to deliver these changes in the customer environment.

Also the reason why we have chosen this model is because of its advantages such as -

- Simple and easy to understand and use
- Easy to manage due to the rigidity of the model. Each phase has specific deliverables and a review process.
- Phases are processed and completed one at a time.
- Works well for smaller projects where requirements are very well understood. ● Clearly defined stages.
- Well understood milestones.
- Easy to arrange tasks.
- Process and results are well documented.

Identify the tools which you want to use throughout the lifecycle like planning tool, design tool, version control, development tool, bug tracking, testing tool.

Planning tool - Microsoft project,Lucidchart,Redmine

Design tool - Figma

Version control - Git

Development control - ● Front end
- ○ HTML
- ○ CSS(Bootstrap)
- ○ Javascript

- Backend
  - Server-Apache web server
  - Database- MySQL
    - IDE
  - Visual studio code

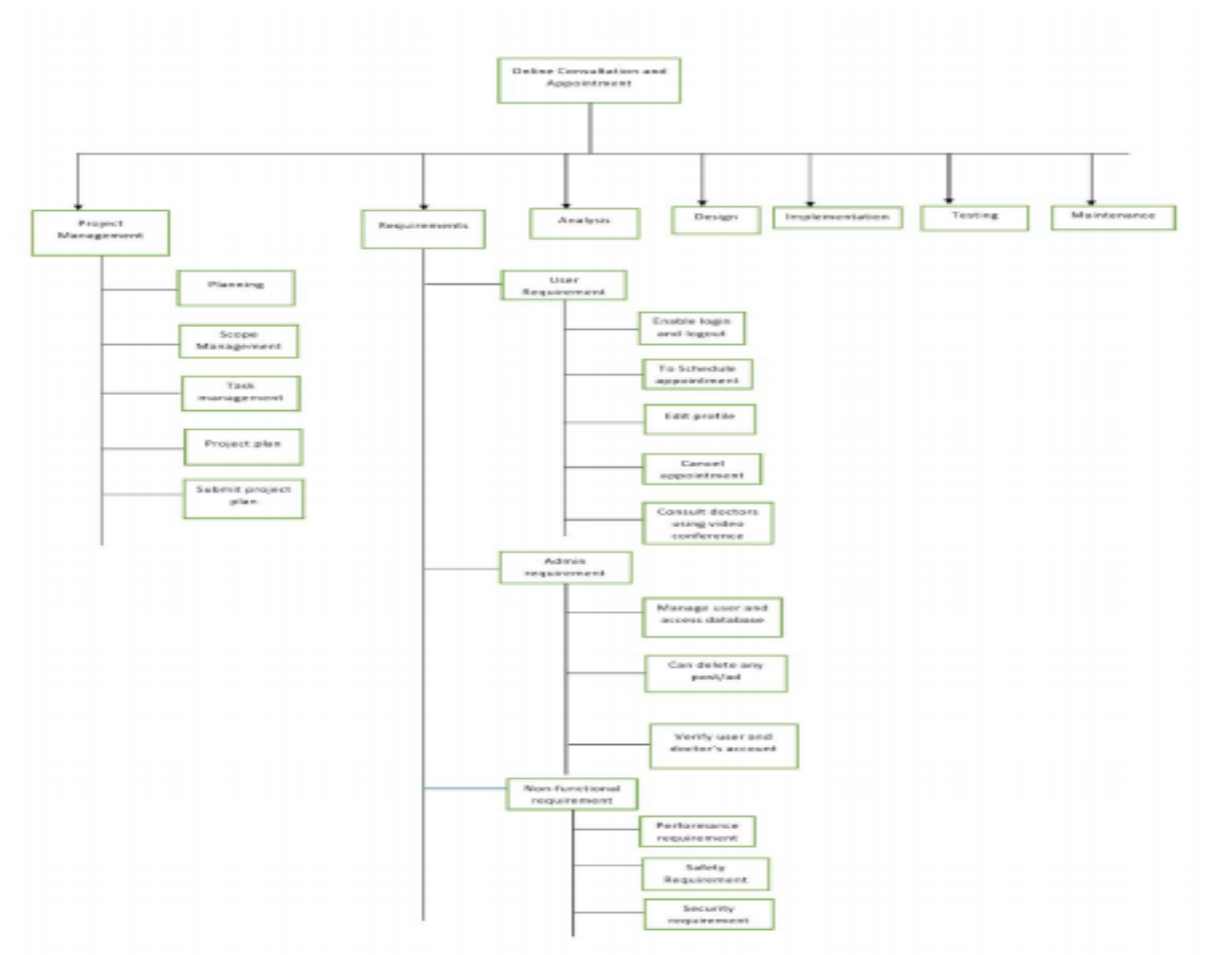
- StarUML

Use case diagram visualization tool

Bug tracking - Redmine,Postman,Bugzilla Testing tool -

Selenium

Create a Work Breakdown Structure for the entire functionalities of your project in detail.



Determine all the deliverables and categorise them as reuse/build components and justify the same.

The System will basically consist of four different users that would be interacting with the aim of providing better healthcare service through mutual utilization of self-monitoring and the consultation from a specialist. The mutual interaction exists

because users will have the option of sending the data for analysis to the doctor, getting the feedback and then acting on his advice. The four users are: - 1. Patient

2. Doctor
3. Administrator

The Patient will register himself with system initially by providing the various personal details that includes his name, age, sex, etc., along with the doctor's name with which he wants to consult, if at all. The registration part is mandatory before the user is able to use the system for his task completion. Then the user proceeds with login using his/her unique id and password.

The role of doctor is vital to any healthcare process and would provide the provision in our project to get doctors feedback. The patient should be able to send the report to the doctor, who was specified by the customer during his/her registration. But this functionality should be entirely upon the discretion of the user whether he/she chooses doctor's analysis or not. If the customer feels he/she is satisfied with his improvement and performance after monitoring the data provided by our application, then the patient can logout of the system.

The role of administrator should be limited to the validation of authenticity of the patient or the doctor. The parameters that admin would use for validating the authenticity of the doctors are external to the project and not in its scope. The designed application will be extremely user friendly since the users can be of any age. Simplicity in presenting the application to the user will be the key to success of this system.

User Interface:

● Home Page:

Actor: Any users

Flow:

Any user can browse this page.

● Admin login page

Actor: Admin users

Flow:

1. Only Admin users can browse this page.

2. Admin users can search all patient appointments and all users' activities.

● User (Patient) login page

Actor: User.

Input: User email and password

Output: User profile page.

Flow:

1. User Logs in with user email and password.
2.

Alternate Flow:

1. If the user email is wrong then it is asked to login again.
2. If the password is wrong then the user is asked to enter again.

● User account create page

Actor: User.

Input: User name, email or mobile number and password

Output: Create a new user and show a user profile page.

Flow:

1. Passwords must be more than 8 characters
2. Password and confirm password must be same

Alternate Flow:

1. If the mandatory fields are not filled up then alert is shown.
2. If password less than 8 characters or password and confirm password not match alert is shown.

● User Details page

Actor: User
Flow:
1. Users must fill up all input fields.

Alternate Flow:
1. If the mandatory fields are not filled up then alert is shown.

- Doctor Login page

Actor: Doctor

Flow:
1. User Logs in with user email and password.

Do a rough estimate of effort required to accomplish each task in terms of person months.

Cocomo (Constructive Cost Model) is a regression model based on LOC, i.e number of Lines of Code. It is a procedural cost estimate model for software projects and often used as a process of reliably predicting the various parameters associated with making a project such as size, effort, cost, time and quality.

Effort: Amount of labor that will be required to complete a task. It is measured in person-months units.

Types of Models:

1. Basic COCOMO Model
2. Intermediate COCOMO Model
3. Detailed COCOMO Model

Basic COCOMO Model

Basic COCOMO can be used for quick and slightly rough calculations of Software Costs. Its accuracy is somewhat restricted due to the absence of sufficient factor considerations.

- E= a(KLOC)^b
- time= c(Effort)^d
- Person required = Effort/ time

For our project we will assume Kilo lines of code = 1.5

For basic COCOMO model

a=2.4

b=1.05

C=2.5

d=0.38

Effort = 4.96 person months

Time = c(Effort)^d

Time = 4.59 months

Person Required = Effort/time Person Required = 1

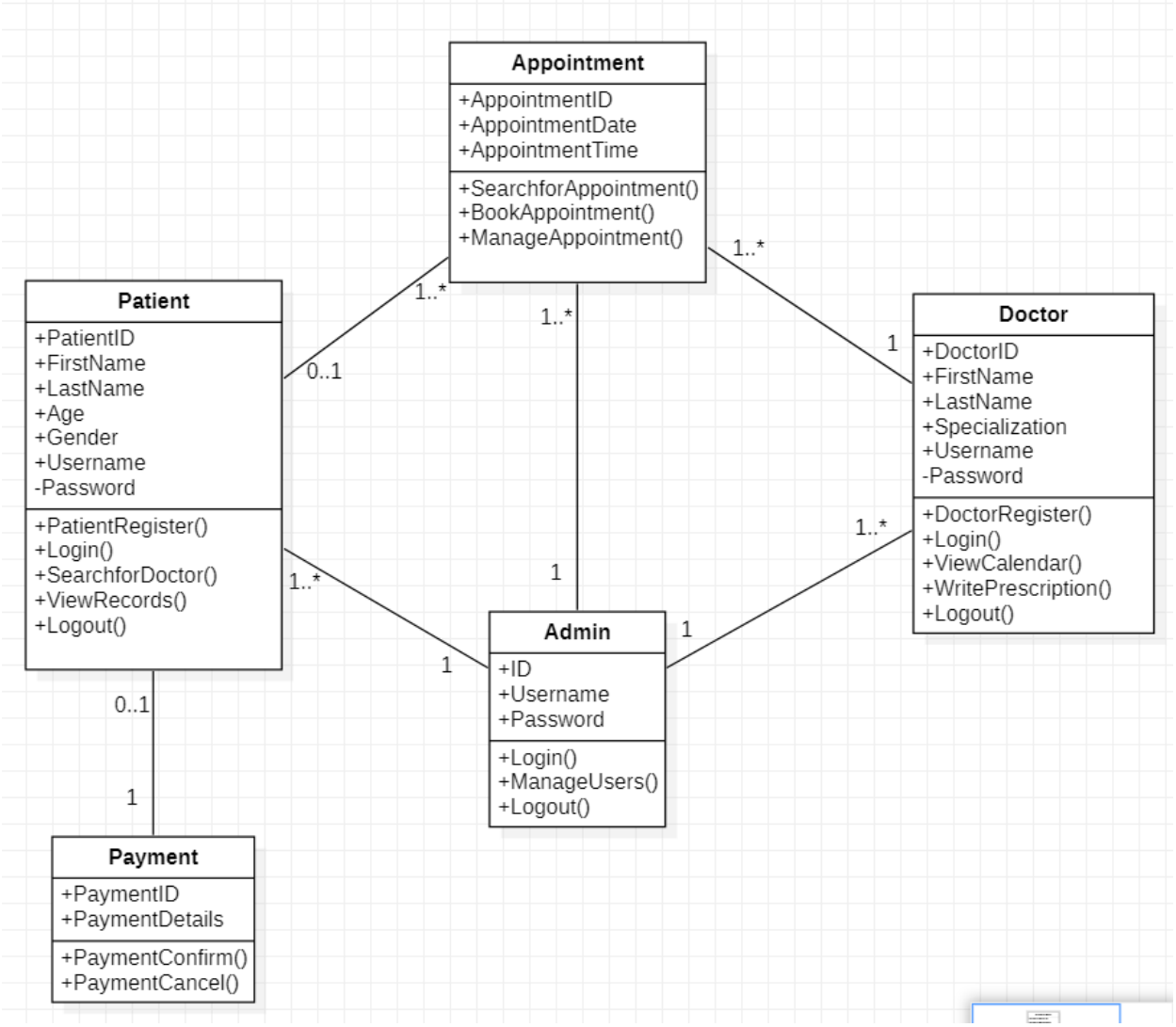
Online consultation and appointment (practo) Page 9 **Create the Gantt chart for scheduling the defined tasks.**

Task	W1	W2	W3	W4	W7	W8	W9	W10	W11	W12	W13	W14
Project planning												
Design												
Test Cases												
Implementation												
Report												

DESIGN DIAGRAMS:

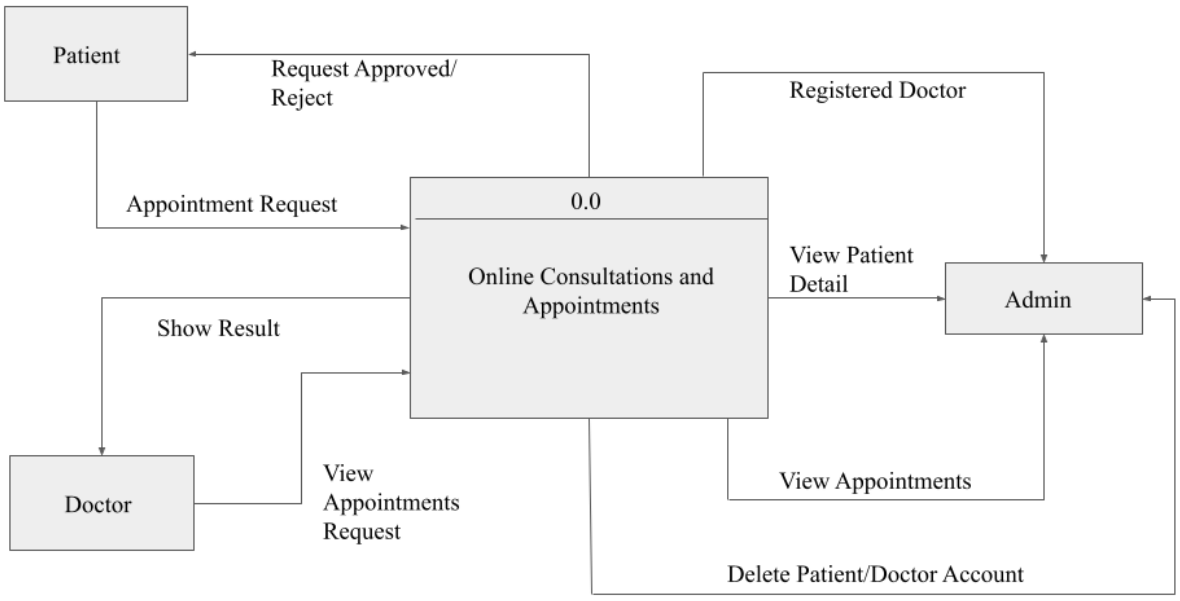
1.Class Diagram

The identified classes are Patient,Doctor,Appointment,Admin and Payment with their associated functionalities.



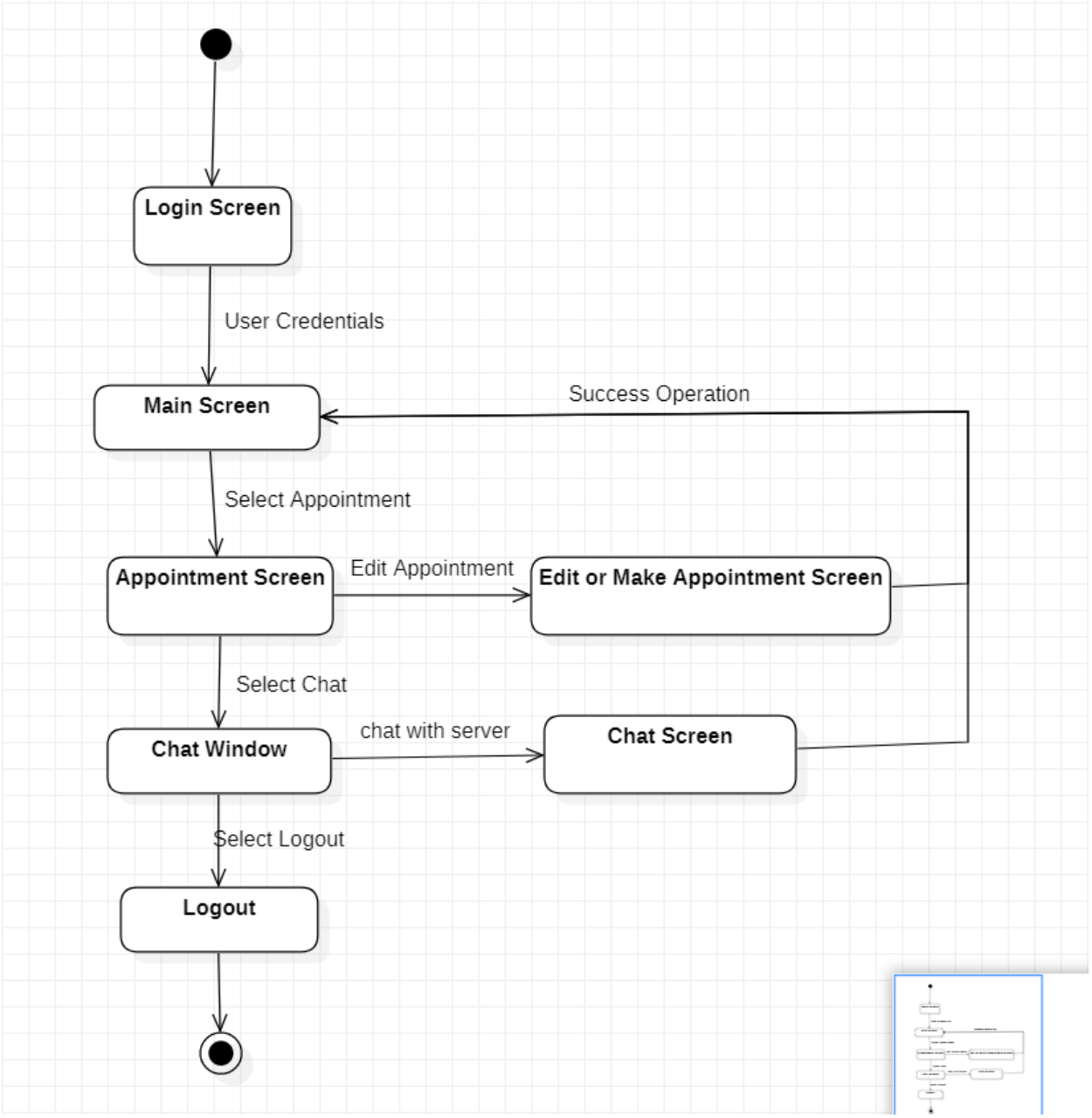
2. System Architecture:

The below figure describes the overall architecture of the system with Patient, Doctor and Admin as entities.



3. State Diagram:

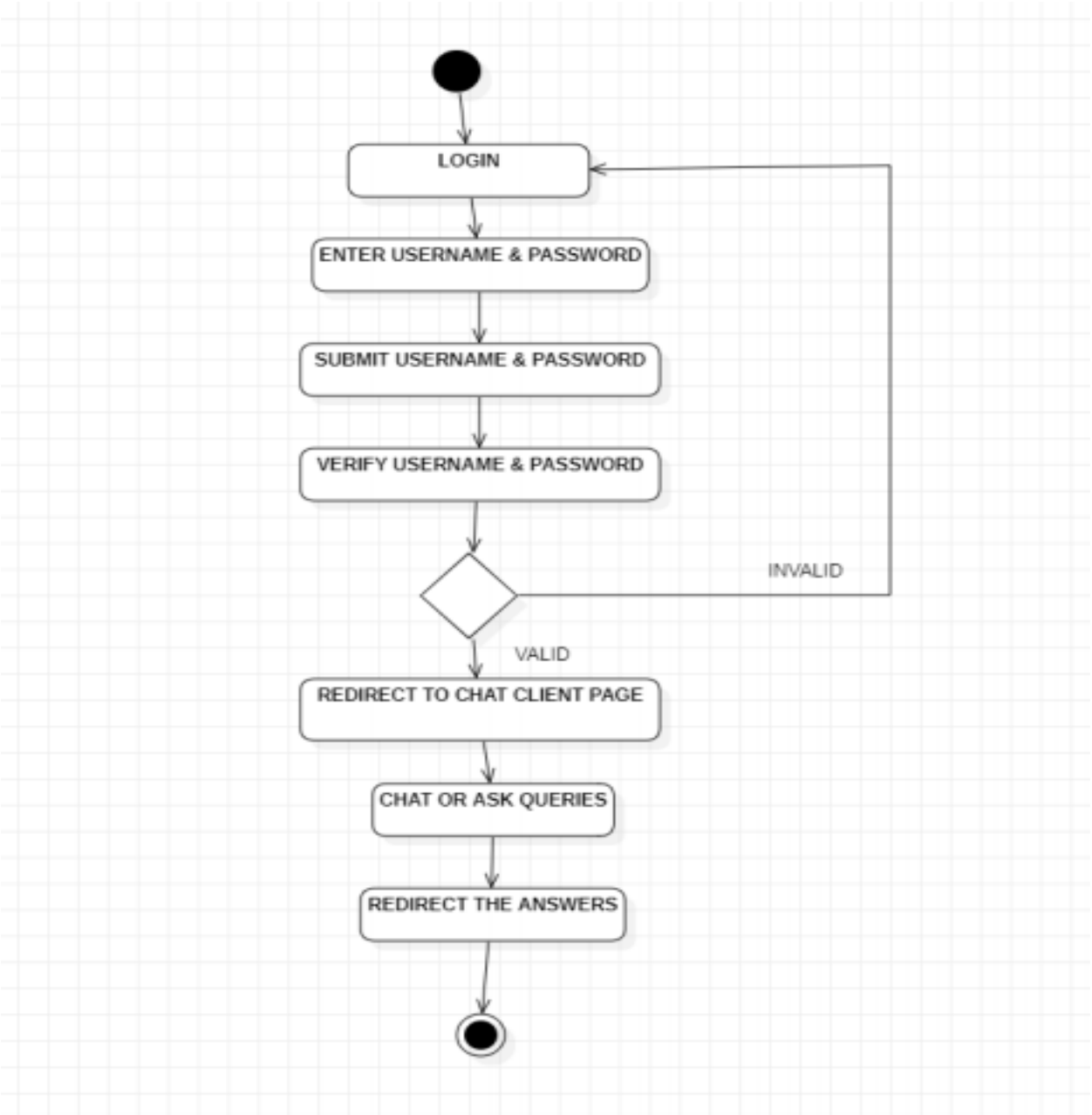
Below is a state diagram used to represent the condition of the system or part of the system at finite instances of time.



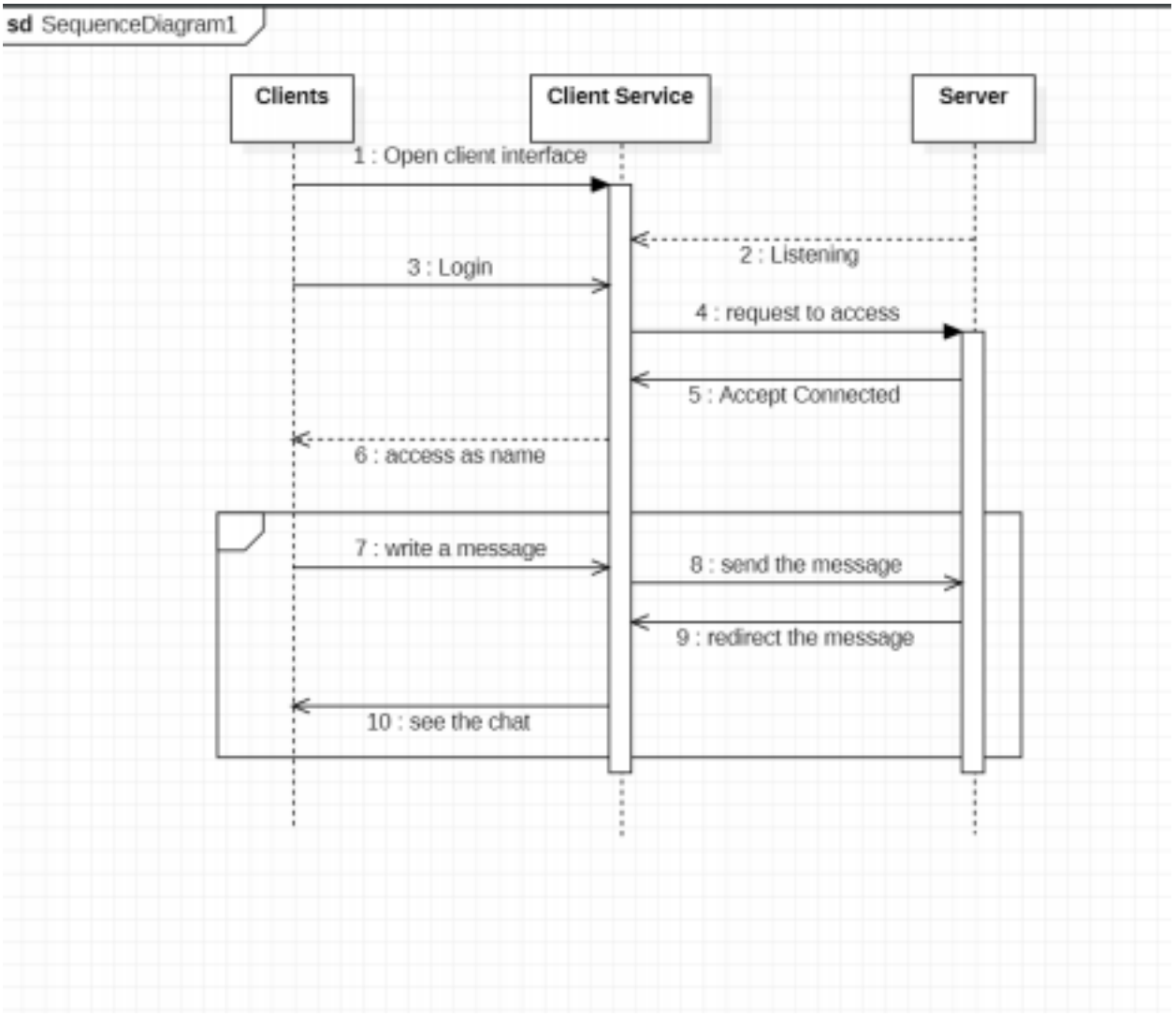
Use cases:

1.Online Chat Consultation based on appointment.

Activity Diagram:

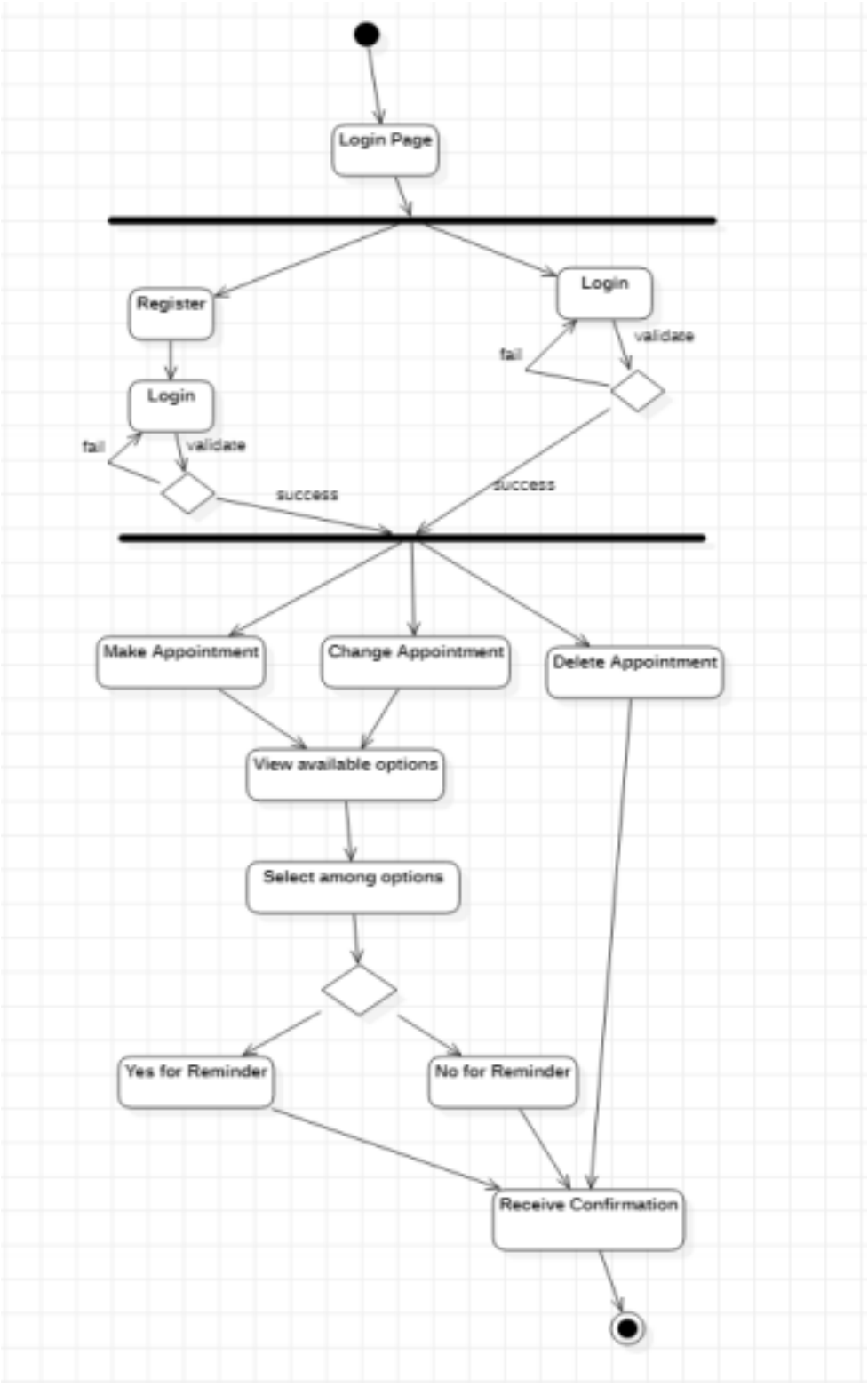


Sequence diagram:

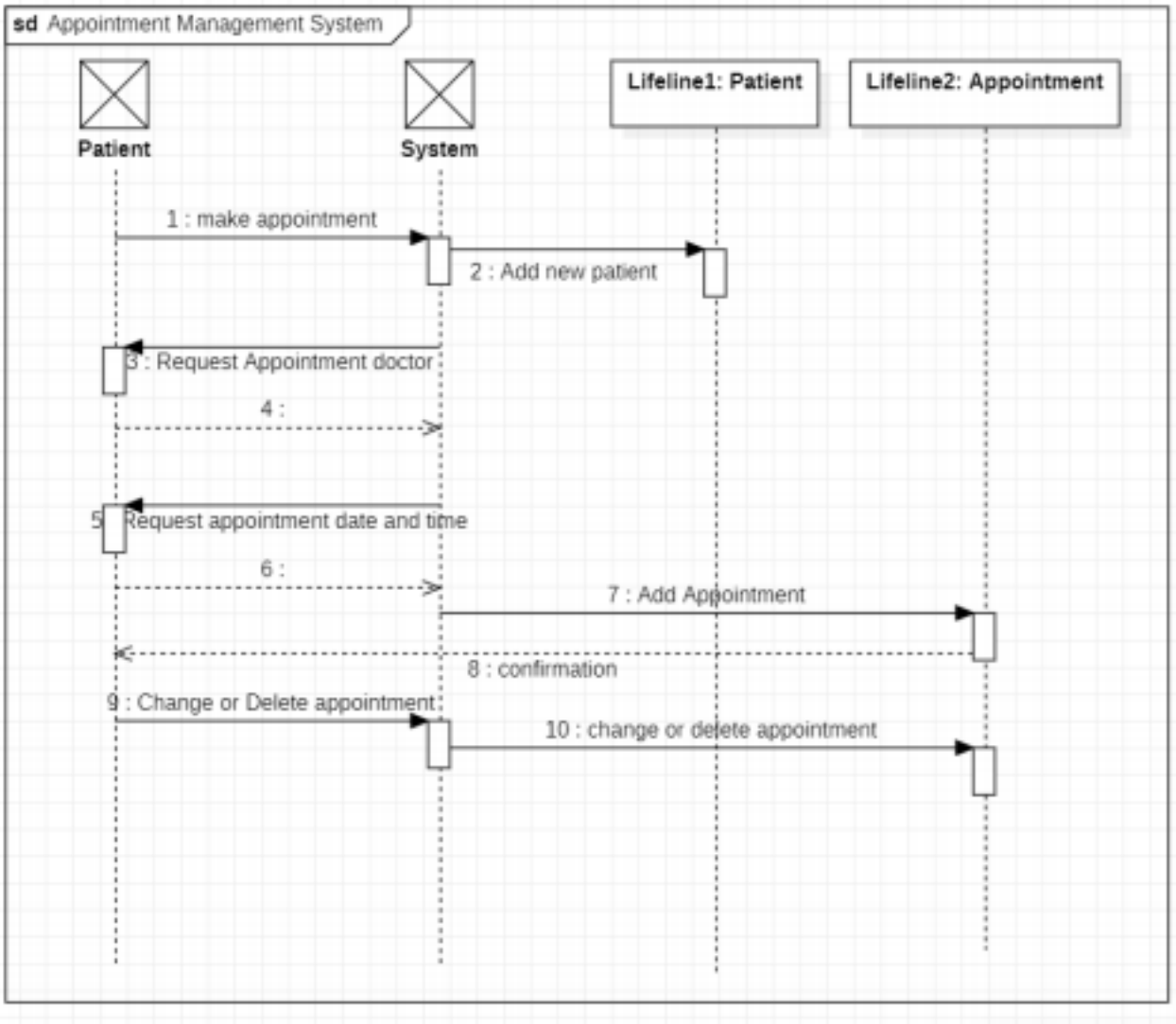


2.Management of Appointment:

Activity Diagram:

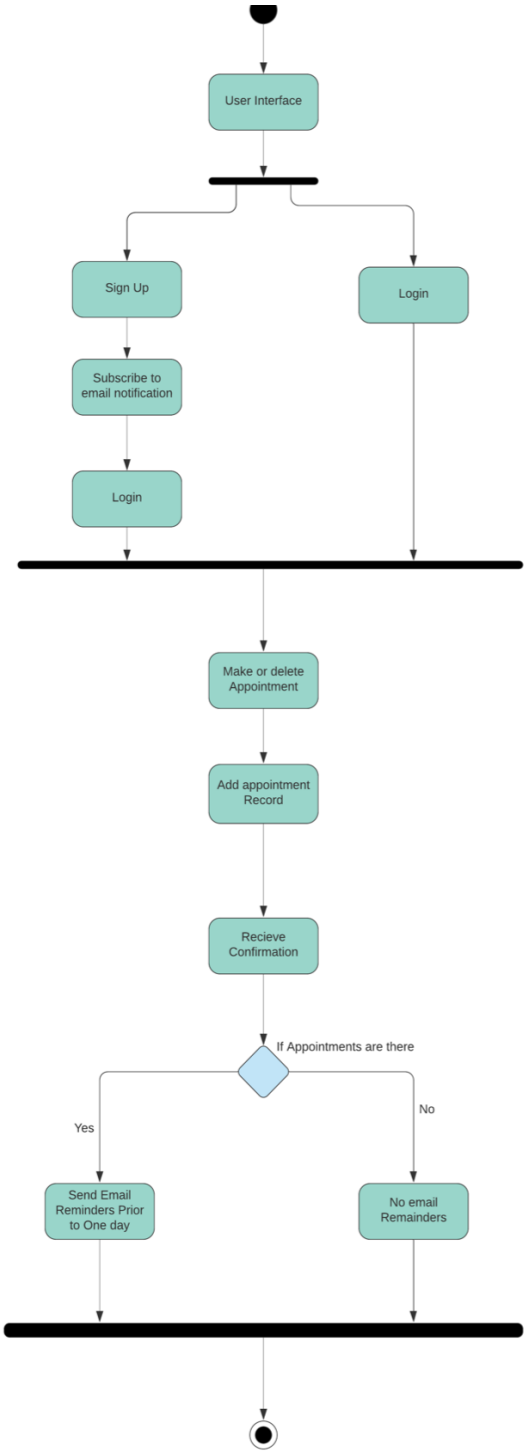


Sequence diagram:

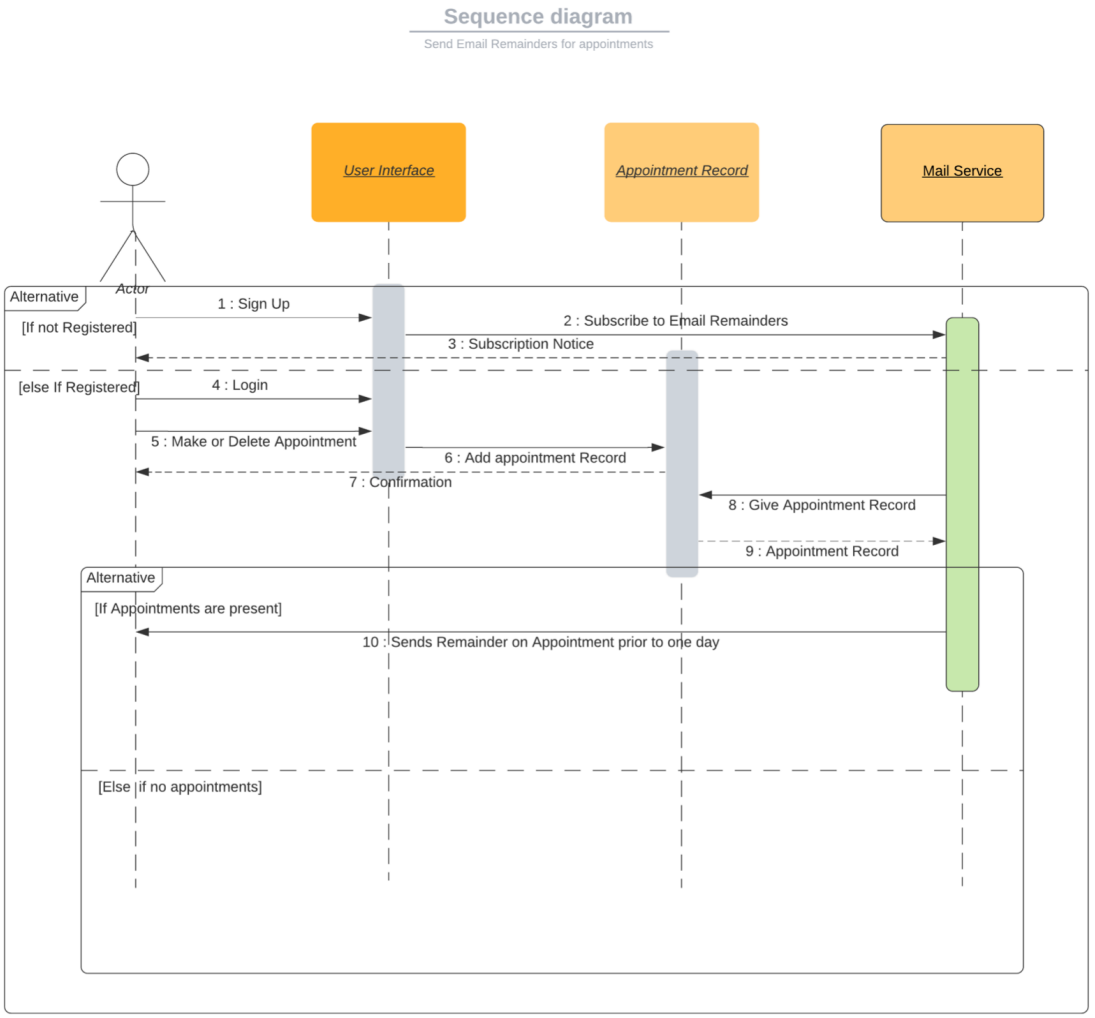


3.Send Email Reminders for appointment

Activity Diagram;



Sequence Diagram:



Module Description:

1. Management of Appointment

Management of Appointment involves managing the appointment and related changes to that appointment till the appointment is completed or cancelled. In our system, managing the appointment includes three types namely, Booking an appointment, Modifying an existing appointment and Cancelling the appointment.

Booking an appointment involves selecting a doctor and a time slot for treatment. Modifying an appointment involves rescheduling the same doctor for a different time slot.Cancelling the appointment involves deleting the appointment from both patient and doctors' calendar.We also have the option to view all appointments scheduled for a particular patient or doctor in our system.

Coming to the implementation part, we have two derived classes namely 'Patient' and 'Doctors' which are derived from Base class 'Person' with necessary functionalities.We also have the 'calendar' class for maintaining the schedule of each registered patient and doctor. We also make use of files to record each transaction and use them for our future reference.

2. Online Chat system

We have implemented chat application system to clarify any queries the user have , the chat application is accessible only to registered user and user who have appointment fixed with doctor , new user can't access the chat application , since the system admin handles the server user will get all the clarification for his/her questions
We implemented it using socket programming Because Sockets can be thought of as endpoints in a communication channel that is bi-directional, and establishes communication between a server and one or more clients. Here, we set up a socket on each end and allow a client to interact with other clients via the server.
Server side script
The socket on the server side associates itself with some hardware port on the server side. Any client that has a socket associated with the same port can communicate with the server socket. The server side script will attempt to establish a socket and bind it to an IP address and port specified by the user The script will then stay open and receive connection requests, and will append respective socket objects to a list to keep track of active connections
Client side script
The client side script will simply attempt to access the server socket created at the specified IP address and port. Once it connects, it will continuously check as to

whether the input comes from the server or from the client, and accordingly redirects output. If the input is from the server, it displays the message on the terminal. If the input is from the user, it sends the message that the user enters to the server.
We have implemented using Half duplex method.To test the chat application's server functionalities we have implemented 8 user test cases and we also performed system integration test cases also.

3. Send Email Reminders

The send email reminders for appointments module was done to send email reminders to users who have booked appointments with doctors.It is one of the means of communication between the system and users.Whenever the user books or modifies an appointment the email reminder module asks for the email for which the email notification has to be sent and it will send an email with the reminder for the appointment with the respective email if the email id is valid.This Is also checked internally in the code by regex expression.The email will go as soon as the user books the appointment.The contents of the email are the time and date of the booked appointment by the user or the modified time and date of the appointment the user has changed.The user is sent separate reminders for the appointment that he books.For each appointment there is a email sent to the users specified email.

The implementation of the module is done in python as is the whole project.Python, being a powerful language, doesn't need any external library to import and offers a native library to send emails- “SMTP lib”. “smtplib” creates a Simple Mail Transfer Protocol client session object which is used to send emails to any valid email id on the internet. Different websites use different port numbers.In this module, we are using a Gmail account to send an email. Port number used here is ‘587.Steps to send mail from Gmail account:First of all, “smtplib” library needs to be imported.After that, to create a session, we will be using its instance SMTP to encapsulate an SMTP connection.s = smtplib.SMTP('smtp.gmail.com', 587)
In this, we need to pass the first parameter of the server location and the second parameter of the port to use. For Gmail, we use port number 587.For security reasons, now we have to put the SMTP connection in the TLS mode. TLS (Transport Layer Security) encrypts all the SMTP commands. After that, for security and authentication, we need to pass your Gmail account credentials in the login instance.The compiler will show an authentication error if an invalid email id or password is entered.We store the message to sent I.e is the date and time of the appointment in a variable say, message. Using the sendmail() instance, we send the message. sendmail() uses three parameters: sender_email_id, receiver_email_id and message_to_be_sent. The parameters need to be in the same sequence.
This will send the email from the system account.After the task is completed, we terminate the SMTP session by using quit().

TEST CASES:

Use case-1:Online Chat Consultation based on Appointment

Test Case ID	Name of Module	Test case description	Pre-conditions	Test Steps	Test data	Expected Results	Actual Result	Test Result
UT-13	Online chat module	To Test The Chat Functionality	User Should Login	1.Navigate to the interface 2.Select the chatbot icon	Username: test_r Age: 20 Email: t1@gmail.com Password: 1	he must be able to chat	He is able to chat	PASS
UT-14	Online chat module	To Test The Chat Functionality	New User	1.Navigate to the interface	Username: test_r	The user must not be able to chat	The user is not able to chat	PASS
UT-15	Online chat module	To test the chat room functionality	Login as a patient and already made an appointment	1.Login as patient 2.Select the chatbot icon 3.Enter the chat room	1.Logging as patient “test_r”	He must be able to enter the chat room	He is able to enter the chat room	PASS
UT-16	Online chat module	To Test the functionality of chat room’s server	Login as a patient and already made an appointment and should send an message	1.Login as patient 2.Select the chatbot icon 3.Enter Correct Username	1.Logging as patient “test_r”	The user should be able to send message and server should receive it	the server will receive the message	PASS
UT-17	Online chat module	To Test the functiona	Login as a patient and	1.Login as patient	1.Logging as patient “test_r”	The server must	The server will send	PASS

		lity of chat room's server	should send an message	2.Navigate to chatbot in user interface		respond to that message sent by user	an message to the user	
UT-18	Online chat module	To Exit	Login as a patient and already made an appointment	1.Login as patient 2.Navigate to chatbot	1.Logging as patient "test_r".	the user must not access the chat room	the user will not be able to access chat room once exited	PASS
IT-6	Integration with chat system	Working of "Management of Appointment" module after integration with chat system	Login as Patient	1.Login as patient 2.Navigate to "Chat system"	1.Logging as patient "test_r"	User should be able to send and receive messages	User was able to send and receive messages	PASS
ST-3	System Functionality	Overall System Functionality	Working Software	1.End to End usage of software	1.Login patient as "test_r". 2. Perform all operations.	All modules should work as expected.	All modules are working as expected.	PASS

UseCase-2: Management of Appointment

Test Case ID	Name of Module	Test case description	Pre-conditions	Test Steps	Test data	Expected Results	Actual Result	Test Result
UT-1	User Registration	To Register Users as Patients .	Access to the interface	1.Navigate to the interface 2.Select the option of Register 3.Enter all the required data	Username: test_r Age: 20 Email: t1@gmail.com Password: 123	Registration should be successful and login automatically	Registration successful and logged in automatically with “Login successful” message	PASS
UT-2	User Registration	Login of Registered Users where correct data is provided	Registered already as doctor/patient	1.Navigate to the interface 2.Select the option to Log-in 3.Enter correct Username and Password	Username: test_r Password: 123	Logging successfully with “Login successful” message	Logged in successfully with “Login successful” message	PASS
UT-3	User Registration	Login of Registered Users where incorrect data is provided	Registered already as doctor/patient	1.Navigate to the interface 2.Select the option to Log-in 3.Enter incorrect Username and Password	Username: test_r Password: abc	Do not log in, displaying the message “Invalid” and asking the credentials again	Did not log in, displaying the message “Invalid” and asking the credentials again.	PASS
UT-4	Appointment Management	To Make an appointment	Login as patient	1.Login as patient 2.Navigate to “Make Appointments” 3.Pick a Doctor	1.Logging as patient “test_r” 2.select Doctor named “t2_doc” 3. select time slot	Allot appointment for patient with doctor and time slot selected	Allotted appointment for patient with doctor and time slot selected and displayed “Appointment Successful!” message	PASS

				4.Select from available slots	“9:00AM”			
UT-5	Appointment Management	To Modify an appointment	Login as a patient and already made an appointment	1.Login as patient 2.Navigate to “Modify Appointments” 3.Select the appointments to modify 4. Pick a time slot	1.Logging as patient “test_r” 2.Select the previously made appointment. 3.Pick a new slot 10:00AM	Modify the appointment to the selected slot. Display message “Appointment modified Successfully!”	Modified the appointment to the selected slot. Displayed the message “Appointment modified Successfully!”	PASS
UT-6	Appointment Management	To cancel an appointment	Login as a patient and already made an appointment	1.Login as patient 2.Navigate to “Cancel Appointments” 3.Select the appointment to cancel	1.Logging as patient “test_r” 2.Select the previously made appointment to cancel	Cancel the appointment. Display message “Appointment cancelled Successfully!”	Cancelled the appointment. Displayed the message “Appointment cancelled Successfully!”	PASS
UT-7	Appointment Management	To view all appointments	Login as a patient and already made an appointment	1.Login as patient 2.Navigate to “View Appointments”	1.Logging as patient “test_r”.	Display the appointment.	Displayed the appointment.	PASS
IT-1	Integration with chat system	Working of “Management of Appointment” module after integration with chat system	Login as Patient	1.Login as patient 2.Navigate to “Chat system”	Logged in as patient “test_r” send “hi” to server	The chat interface opens up for asking queries and the server responds back.	The chat interface opened up for asking queries and the server responds back with message “hello”	PASS
IT-2	Integration with	Working of	Login as Patient	1.Login as patient	Logged in as	To send an Email	Email was sent with	PASS

	email option	“Management of Appointment” module after integration with Email option.	and make an appointment	2.Navigate to “Make Appointments” 3.After making an appointment, decide whether to send an email reminder or not.	patient “test_r” made an appointment and opted to receive email. with given mail-id.	with necessary details to the patient's given mail id.	necessary details to the patient's given mail id.	
ST-1	System Functionality	Overall System Functionality	Working Software	1.Login as patient. 2.Make an appointment. 3.Opt to send a confirmation mail 4.Modify an appointment. 5.Cancel the appointment. 6.Use the chat system	1.Login patient as “test_r”. 2. Perform all operations.	All modules should work as expected.	All modules are working as expected.	PASS

Use Case-3: Send Email Reminders for Appointments

Test Case ID	Name of Module	Test case description	Pre-conditions	Test Steps	Test data	Expected Results	Actual Result	Test Result
UT-9	Verifying the email address	To Verifying the email address using regex	Logged in as patient and Provide the recipient email	1.Call the function with recipient email id and the message	1.recipients correct email id 2.mesaage to be sent	Email With the appropriate message to be sent if email id is right	Email With the appropriate message was sent as the email id was right	PASS
UT-10	Verifying the email address	To Verifying the email address using regex	Logged in as patient and Provide the recipient email	1.Call the function with recipient's wrong email id and the message	1.recipients wrong email id 2.mesaage to be sent	Email With the appropriate message not to be sent	Email With the appropriate message was not sent as the email id was wrong	PASS
UT-11	Sending email to the given user	To send the email to the given user	Logged in as patient and Provide the recipient email and message	1.Call the function with recipient email id and the message	1.recipient email id 2.mesaage to be sent	Email With the appropriate message to be sent	Email With the appropriate message was sent	PASS
UT-12	Frequency of email notifications	To see the frequency of email notifications	Logged in as patient and Provide the recipient email and message	1.Call the function with recipient email id and the message	1.recipient email id 2.mesaage to be sent	Email With the appropriate message to be sent once	Email With the appropriate message was sent once	PASS
IT-3	Integration with Management of	To Receive email confirmation on	Login as patient and make an appointment	1.Login as a patient with the right credentials.	1.Logging as patient "test_r"	Receive an email with necessary details	Receive an email with necessary details of	PASS

	Appointment	appointments		2. Make an appointment. 3. Opt to receive email confirmation.	2.Make an appointment with the given doctors	of the appointment.	the appointment.	
IT-4	Integration with Management of Appointment	To test the mail reminder functionality	Login as patient and make an appointment	1: Navigate and Login to the website 2: Book an appointment to preferred doctor	Booking unsuccessful, User provided valid mail ID	He must not receive mail of appointment details	Appointment details not received via mail	PASS
IT-5	Integration with Management of Appointment	If the User modifies the appointment then the email reminder has to be changed	Login as patient and modify an appointment	1.Login as a patient with the right credentials. 2. Modify an appointment. 3.Select reminders for the modified appointment	1.Logging as patient “test_r” 2.Modify an appointment with the given doctors	Receive an email with necessary details of the appointment.	Received an email with necessary details of the appointment.	PASS
ST-2	System Functionality	Overall System Functionality	Working Software	1.End to End usage of software	1.Login patient as “test_r”. 2. Perform all operations.	All modules should work as expected.	All modules are working as expected.	PASS

SCREENSHOTS OF OUTPUT:

1.Register

```
------Welcome to Online Consultation-------

Are you:
1.Doctor
2.Patient
3.To Quit
2
Did you already Register?
1.Yes
2.No
3.Go back
2
Enter Your Name: demo
Enter Your Age: 24
Enter Your e-mail-id: demo@gmail.com
Enter Your Password: 1234
Re-enter Your Password: 1234
Password match, you are logging in
login successful!
```

2.Login

```
------Welcome to Online Consultation-------

Are you:
1.Doctor
2.Patient
3.To Quit
2
Did you already Register?
1.Yes
2.No
3.Go back
1
Enter Your Username:demo
Enter Your Password:1234
login successful!

Select a service you need:
1.Make an appointment
2.Modify an appointment
3.Cancel an appointment
4.View all appointments
5.Chat with our officials
6.Go back
```

3.Management of Appointment

```
Select a service you need:
1.Make an appointment
2.Modify an appointment
3.Cancel an appointment
4.View all appointments
5.Chat with our officials
6.Go back
1
Select Doctor for appointment:

Doctor          Speciality
-----------------------
t1_doc          Dentist
t2_doc          Cardiologist
t3_doc          Oncology
t4_doc          Neurology
t5_doc          Gynaecology
t6_doc          Dermatology
Please enter the doctor name for your appointment
t1_doc
Pick a time slot:

8:00AM
8:30AM
9:00AM
9:30AM
10:00AM
10:30AM
11:00AM
11:30AM
1:00PM
1:30PM
2:00PM
2:30PM
3:00PM
3:30PM
4:00PM

9:00AM
Appointment Successfull!
```

```
2
Select the appointment you wish to modify:

Patient         Doctor          Slot
-----------------------------------------------------------------
demo            t1_doc          9:00AM

Select from available_slots:
8:00AM
8:30AM
9:30AM
10:00AM
10:30AM
11:00AM
11:30AM
1:00PM
1:30PM
2:00PM
2:30PM
3:00PM
3:30PM
4:00PM

Enter the new slot: 10:00AM
Appointment modified Successfully!
```

```
Select a service you need:
1.Make an appointment
2.Modify an appointment
3.Cancel an appointment
4.View all appointments
5.Chat with our officials
6.Go back
4
Patient         Doctor          Slot
-----------------------------------------------------------
demo            t1_doc          10:00AM
```

```
Select a service you need:
1.Make an appointment
2.Modify an appointment
3.Cancel an appointment
4.View all appointments
5.Chat with our officials
6.Go back
3
Select the appointment you wish to cancel:

Patient         Doctor          Slot
-------------------------------------------------------------
demo            t1_doc          10:00AM
Enter the doctor  with which you wish to cancel appointment: t1_doc
Appointment cancelled Successfully!
```

4. Chat System

```
Select a service you need:
1.Make an appointment
2.Modify an appointment
3.Cancel an appointment
4.View all appointments
5.Chat with our officials
6.Go back
5
Press 'q' to exit
>>hi
>> hello, how can i help you?
>>i am patient demo
>> ok
>>
```

```
Welcome Chat is Ready
Press 'q' to exit
>> hi
>>hello, how can i help you?
>> i am patient demo
>>ok
```

5. Email Confirmation

pavankumardesai39@gmail.com
to bcc: ashaygowda

Greetings

This is Confirmation of your Appointment.
Your appointment is scheduled as follows

Doctor:t4_doc Patient:Ashay Slot:9:00AM

THANK YOU