CLINIC MANAGEMENT & BOOKING SYSTEM

Developed By

Huda Ahed AbdelKareem

3378-FBAS/BSSE/S17

BS Software Engineering

Supervised By

Ms. Asma Khatoon and Ms. Saima Iqbal

Department of Computer Science & Software Engineering

Faculty of Basic and Applied Sciences

International Islamic University, Islamabad (2020)

In the name of ALLAH ALMIGHTY the most beneficent, the most merciful

FINAL APPROVAL

Dated: _____

It is certified that we have examined the thesis titled **CLINIC MANAGEMENT & BOOKIN SYSTEM** submitted by **Huda Ahed AbdelKareem (Reg # 3378-FBAS-BSSE-S17))).**It is our judgment that this project is of sufficient standard to warrant its acceptance by the International Islamic University Islamabad for the Degree of Software Engineering.

.

Panel Members:

Ms. Saima Iqbal
Lecturer,
Department of Computer Science& Software Engineering
Faculty of Basic and Applied Sciences
International Islamic University, Islamabad

_____

Ms. Asma Khatoon
Lecturer,
Department of Computer Science& Software Engineering
Faculty of Basic and Applied Sciences
International Islamic University, Islamabad

_____

DISSERTATION

A dissertation is submitted to the Department of Computer Science & Software Engineering, International Islamic University Islamabad, as a partial fulfillment of requirements, for the award of the degree.

BS in Software Engineering

Dedication

We commit this undertaking to Almighty Allah (SWT), Who made us and by Whose leniency we could use our capacities. He gave us inspiration and right heading to finish this undertaking. And for my great parents and their support during this journey, also special thanks for all my instructors, teachers my classmates and everyone helped in guiding me achieving this project.

Declaration

I hereby declare that this project, neither as a whole nor as apart thereof has been copied out from any source. It is further declared that we developed this project and this report entirely on the basis of our personal efforts made under the sincere guidance of our project supervisor. No portion of the work presented in this report has been submitted in support of any application for any other degree or qualification of this or any other University or institute of learning. We further declare that this software and all associated documents, report and records are submitted as partial requirements for the degree of Bachelor's in Software Engineering.

Huda Ahed AbdelKareem

3378-FBAS/BSSE/S17

Acknowledgment

All praise to Almighty Allah, who gave me the understanding, courage, and patience to complete this project.

Gratitude and appreciation to everyone in anyway encouraged me and supported me in completing this project, in my entire journey to complete this degree. My sincere thanks to my parent for their love and support for raising me to be that person

Genuine thanks and pleasures to my teachers and supervisors**, Ms. Asma Khatoon and Ms. Saima Iqbal** for their help, guidance, support to learn and enhance my knowledge.

Project in Brief

Project Title:	**Clinic Management System**
Objective:	• To automate the process of taking and scheduling appointments. • Provide easy way to track the procedure of treatment, track the test report and writes them
Undertaken By:	Huda Ahed AbdelKareem 3378-FBAS/BSSE/S17
FYP Panel:	**1-** Ms. Saima Iqbal **2-** Ms. Asma Khatoon
Date Started:	August/2020
Date Completed:	December/2020
Tools Used:	• Visual Studio 2019 • SQL Server Management Studio • ASP.NET 3.1 • Html and CSS • Bootstrap
Operating System:	Acer Aspire A515-51G,Inter Core(TM) i7-8550U CPU @ 1.80GHz

Abstract

Clinic Management System is to organize and automate the management in clinics, still these days few clinics still manually recording and managing Patients', Doctors' details, therefore manually recording these details require space and time, easy can be mishandled, receptionist could duplicate details mistakenly, difficulty in searching to find specific record, and consumes long time to sort, so automating the system is important.

Clinic Management System (CMS) will help clinics to keep and manage the personal records of patients, also the system is providing booking appointments functionality to patients to book an appointment automatically from their place with no need to visit clinic or call, as well as doctors can manage and organize their schedule on the system, write prescriptions and view the test report also the doctors and patients can track the lap test which is uploaded by the lab technician and get the result immediately when it is ready.

Contents

Chapter 1

Introduction

1. Introduction

Clinic Management System is a website developed to help Patients and Doctors, it is easily scheduling the Doctors' appointments in one place, also Doctors can write prescription through the website, can read the Patients' health history to know if the Patient need special treatment or allergic to any diseases also when the test report is ready the Doctor will be notified.

In Patients side the Patient will have profile includes their information and history, the Patient in order to take an appointment will chose the specialist in need for the case, then will choose among the available appointments, confirmation message will be sent to patient mobile phone, after taking the appointment if test needed he/she will be notified if the result is ready.

Lab technician will add reports for patient and doctors to view, as well as the receptionist can manage appointments, doctors and patients and inform doctors and patients in any case of changes happened, payment done manually.

1.1 Project Motivation

As everything is changing now from the analog world toward digital world, so digitizing the process of taking and scheduling appointments is necessary to adapt the digital world.

1.2 Scope

Clinic Management System is a website. The main aim it is to simplify and digitize the management of clinics, by minizine the manual job of scheduling appointments, provide the patient the ability to arrange appointment from their place by using the website, Doctors able to

write reports and track the lap the test, as well the Patient is able to track the test report and upcoming appointments.

The things in Scope:

- Arrange appointments.
- Doctors writes prescription.
- Track tests.
- Manage appointments
- Book An appointments

The thing out of Scope:

- Payment

1.3 The idea

Ides is inspired by the world these days, as we can see everything is digitized so this management system is intent to automate the business plus providing the ability of booking appointments through the website in easy way as well as it is providing system to Doctors, Patients, lab technician to arrange and manage their tasks easily.

1.4 Need of work

This website developed to automate the management of clinics, keeping the patients attached and ease the process of arranging appointments.

1.5 Functional requirements

1.5.1 Sign Up/Registration

Patient and Doctors can easily register by opening the register page. They have to provide the basic information.

1.5.2 Book An appointment

Patient and receptionist can book appointment by selecting the among the available timing and doctors.

1.5.3 Manage appointments

Patients, Doctors, and Receptionist are able to edit and cancel appointments

1.5.4 Track and upload test report

Patients and doctors are able to view and track test reports uploaded by lab technician

1.5.5 Manage schedule

Doctors are able to edit schedule and view appointments.

1.5.6 View and edit profile

Patients and doctors are able to view and edit their profiles.

1.5.7 Manage Doctors and Patients

Receptionist is able to edit Patients and Doctors profiles.

1.5.8Login

Receptionist, lab technician, Patients and Doctors can log in by entering email and password.

1.6 Objectives

The proposed system aims to facilitate the receptionist, doctors and patients in order to save their effort on manual work. We are providing them facility to save time and human effort.

The main objectives of this online system are:

1.6.1 Save Time:

As the management system and arranging appointment is automated, it save time for all parties.

1.6.2 Save Effort:

As the process of arranging appointment is automated, also the process of managing patients so its saves effort more the manual work.

1.6.3 No Redundancy:

The system generates unique ID for patients, and Doctors when registering so no chance of redundancy.

1.6.4 Efficient scheduling:

The system efficiently will arrange the schedule for each Doctor.

1.6.5 Easy Tracking:

The system will provide Doctors and Patients the ability to track test reports and upcoming appointments.

1.7 Problem Statement

The problem of	• Staff need to arrange schedules and manage the clinic system. • Patients needs to arrange appointments and find the available Doctors and timing, track their test and treatment procedure. • Doctors need to schedule appointment track patients' reports.
Affects	Patients, Doctors and Staff.
The Impact of which is	Wastage of time and money

The successful solution would be	Easy to use and simple web application for Administration, Patients and Doctors providing online registration, arrange appointments, track reports.

Chapter 2
Basic concepts

2.1 Existing System

The current system is to visit or to call the clinic manually looking for available appointments and Doctors, reception will arrange the process manually. The administration should keep and manage patient, and doctors manually. Patients and Doctors cannot track or be notified when the test reports are ready.

2.2 Limitation of Existing system

Following are the Limitations of Existing System of the current management system:

- Patients must register in the clinic.
- Patients have to call or visit the clinic to arrange an appointment.
- Data can be lost.
- Doctors should update and arrange their appointment by themselves.
- Patients have to hand Doctors lab reports by themselves

2.3 Problem Solution

A Solution to all mention problems is to develop an application that can:

- Save Time
- Save record
- Provide Registration
- Provide arranging appointment functionality for patients.
- Provide schedules to doctors.
- Provide tracking of test report.

2.4 Proposed System:

2.1 Existing System

I am developing a web application for Noor general hospital, this application will provide features as registration, saving the record of patients and doctors, track test reports, arrange appointments therefore these features will automated**.**

2.4.1 Roles and Characteristics:

2.4.1.1 Receptionist

Responsible of adding, deleting, editing patients and doctors, receptionist is also responsible for notifying and edit appointment in case any change happened.

2.4.1.2 Doctors

Responsible for providing his/her schedule, write prescription, track the lab test reports.

2.4.1.3 Patients

Responsible for registering himself with correct data, patient can arrange view available doctors and arrange an appointment and can track his/her reports.

2.4.1.4 Lab technician

Responsible for uploading the test reports.

2.4.2 Operating Environment

It is necessary to consider the operating environment for the product been developed. By doing this the development team and the end users are able to deal with product in a good way. Some of the major requirements for operating this system “clinic management system” are as following:

• Good Internet connection to connect online.

• Operating systems – Minimum Windows 8 version

• A server with enough capability to deal with multiple client’s request**.**

• Hardware Specifications • **Processor** of 2.0 GHz,

• **RAM with** 2 GB and a

• Hard Disk: 50 GB.

2.1 Existing System

Frameworks

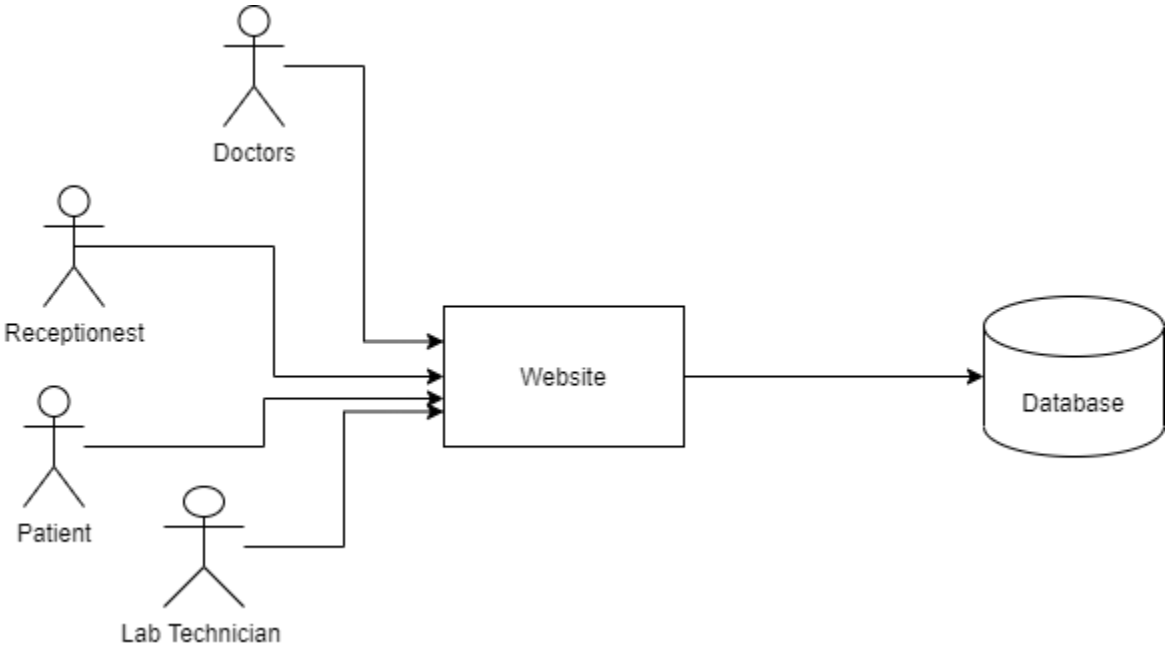
- ASP .Net
- Microsoft SQL Server Management
- Visual Studio 2019
- Bootstrap

Chapter 3
System Analysis

3.System Analysis:

System Analysis is software engineering task that bridge the gap between system level requirement and the software design.

3.1 Product overview



3.2. Software Modules:

Software modules of clinic management system are Following:

3.2.1. Registration/ Sign in:
Patient and Doctors can register himself by providing information like email address and password. User can login into his account by providing valid email address and password.

3.2.2 Login:
Doctors, Patients, Lab Technician and Receptionist can log in to the website by using their registered email and password.

3.2.3 Book an Appointment:
Patient and Receptionist can book an appointment, by selecting the required doctor and available time.

3.2.4 Manage Appointments:
Patients, Doctors and Receptionist can edit and delete appointments.

3.2.5 View and upload test report
Patients and Doctors can view the report test.

Lab technicians upload the test report.

3.2.6 Write prescription
Doctors write prescription to patient after the visit

3.2.7 Manage Doctors
Receptionist can edit and remove Doctors.

3.2.8 Manage Patients
Receptionist can edit and remove patients.

3.2.9 Manage profile
Patients and Doctors can edit their profiles.

3.3.1. User Interface:

The system will have a Graphical User Interface. The components of the interface layout will include menus, buttons, labels, text fields, grid view, list view, image view, cards, pagers etc. to increase look and feel of the application. All pages of the system will be following consistent layout and clear structure. All the records will be displayed then. In navigation drawer all the icons are placed so to easily navigate within the application will also be Logout button as well for the user to log out of the system. Our Application will have following interfaces

- For Patient
- Receptionist
- Lab Technician
- Doctor

3.3.2 Hardware Interface
• Laptop or smartphone.

• Processor with 1.2 GHz, build in memory minimum 8GB and minimum 2GB RAM.

3.3.3. Software Interface:

Software Used	Description
Operating System	Windows, or smartphone
Database	Microsoft SQL server.
Tools	Visual Studio.
Languages	ASP.NET

3.3.4. Communication Interface:

This application uses Wi-Fi Direct to communicate between devices. We will use the Internet to communicate between different users on devices as database is real time and it requires internet to sync data across the application.

3.3.5 Design and Implementation Constraints

Category	Web Application
System	Acer, Core i7, 3rd generation with 8GB RAM.
Tools	Visual studio, Microsoft SQL server management.
Language	Asp.Net

3.4 Functional Requirements

Functional requirements describe the specific behavior of the system that what a system should do. It explains functions of a software system or its component. Functional requirements are supported by non-functional requirements, which impose constraints on the design or implementation (such as reliability, security or performance requirements).

3.4.1 Patient

- Registration

  Patient can register by entering the required details.

- Login

Patient can log in by entering registered email and password.

- Sign out

  Patient can log out from the system

- Book an appointment

  Patient can book an appointment by selecting the time and doctor.

- Manage Appointments

  Patient can edit and cancel appointment.

- View reports

  Patient can view the test report.

- Edit profile

  Patient can edit their profile information

3.4.2 Doctors

- Registration

  Doctors can register by entering the required details.

- Login

  Doctors can log in by entering registered email and password.

- Sign out

  Doctors can log out from the system

- Manage schedule

  Doctors can edit schedule

- Writes prescriptions

  Doctors writes prescription after the visit.

- Manage appointment

  Doctors can edit and cancel his/her own appointments.

- Edit profile

  Doctors can edit their profile information

3.4.3 Lab Technician

- Login

Lab Technician can log in by entering registered email and password.

- Sign out

  Lab Technician can log out from the system

- Upload test report

  Lab technician can upload the test report.

3.4.3 Receptionist

- Login

  Receptionist can log in by entering registered email and password.

- Sign out

  Receptionist can log out from the system

- Book an Appointment

  Receptionist can book an appointment by selecting the time and doctor.

- Manage Appointments

  Receptionist Can cancel and edit appointments.

- Manage Doctors

  Receptionist Can remove and edit Doctors.

- Manage Patients

  Receptionist Can remove and edit Patients.

3.5. Non-Functional Requirements:

Non-Functional requirement defines quality attributes of the system. These requirements define the standard used to judge the specific operation of the system

3.5.1 Reliability

The capability to maintain the specified level of performance is what meant by reliability. This application will run and perform its operations.

3.5.2 Availability

The application will run if internet connection is available.

3.5.3 Usability:

The interface is easy to understand and user can easily navigate within application

- User friendly interface
- Easy to use
- Navigation between screens is easy

3.5.4. Performance:

Response time for every feature is minimum.

3.5.5. Modifiability:

Any modification in feature or database entity is possible.

3.5.6. Portability

The capability adapted for different specified environments without applying actions or means other than those provided for this purpose in the product. Since, phones are portable, so do the application.

3.5.7. Performance Requirements:

The system should work according to the design and requirements of the user. The system should give fast response to user.

- Minimal time between click and system response
- Minimal number of clicks.
- Minimal time for searching
- Minimal time required to place appointment.

3.5.8. Responsiveness

The system should have performed all the tasks with fewer touches in fewer seconds or instantaneous with user touch.

- System responds quickly to user requests or changes in the environment.
- System responds within 2 seconds on average to local user requests and changes in the environment.
- System responds within 4 seconds on average to remote user requests and changes in the environment.

3.6. Product Position Statement

For	Can be anyone (no specific age group or profession)
Who	• Patients • Doctors • Reception • Lab technician
The Clinic management	Is a software application
That	Provides services for, Receptionist, Doctors ,lab technician and patients, to arrange appointments and schedules, view test thay uploaded by lab technician, doctor can writes prescriptions, patient can track treatment procedure
Unlike	currently manual systems where one has to go or call for appointments, viewing test reports and arrange schedules.
Our product	Provide for users the services such as finding the doctor in need, arrange appointments, view the test report uploaded by the lab technician, where it providing to doctors and an easy way to arrange schedules , writing prescriptions and view test report, also helps managing receptionist job easily.

3.7. Stakeholders and User Description

	Web Application
Age	Any
Gender	Any
Location	Islamabad
Education	Any

3.7.2. User Summary

Name	Description	Responsibilities	Stakeholders
Anyone	Web Application end user	Use application to Book appointments, and View test reports	Self
Doctors	Web Application end user	Use application for appointments scheduling, Write prescriptions and view test report	Self
Lab technician	Web Application end user	Use application for Uploading reports.	Self
Receptionist	Web Application end user	Use application for managing appointments, Doctors and Patients.	Self

3.7.3. User Environment

Clinic management system is a web application used by Patients, Doctors, Lab Technician and Receptionist to arrange appointments with doctors and to track to appointment, reports and test reports, this application is making the process of arranging appointments easier for patients as well as managing for reception and doctors easier.

3.7.4. Stakeholder and User Profile

Patients

Description	**Any person who uses our application**
Type	This is the user who finding appointments, and tracking treatment procedure.
Responsibilities	Use the application to book an appointment, view prescriptions, and view the test reports.
Success Criteria	The success is defined as the customers continuing to use our system.
Involvement	We will have sample customers to evaluate our system which will guide our vision.
Deliverables	None
Comments / Issues	None

Doctors

Description	**A person who uses our application as schedule, to write prescriptions.**
Type	A health professional who practices in medicine
Responsibilities	• Ensure Appointments and confirm it. • Write prescription after each visit. • View test reports.
Success Criteria	The success is defined as the Doctors continuing to use our system.
Involvement	We will have sample Doctors to evaluate our system which will guide our vision.
Deliverables	None
Comments / Issues	None

Lab technician

Description	**A person who uploads test reports.**
Type	Lab technician.
Responsibilities	• Upload test report.
Success Criteria	The success is defined as the Lab technician. continuing to use our system.

Involvement	We will have sample Lab technician. to evaluate our system which will guide our vision.
Deliverables	None
Comments / Issues	None

Receptionist

Description	**A person who maintains the record of Patients, Doctors and manage the clinics.**
Type	Receptionist of clinics
Responsibilities	• Ensure Appointments and confirm it. • Manage Doctors. • Manage Appointments. • Manage Patients. • Upload test reports.
Success Criteria	The success is defined as the Receptionist continuing to use our system.
Involvement	We will have sample Receptionist to evaluate our system which will guide our vision.
Deliverables	None

Comments / Issues	None

3.7.5. Key Stakeholders and User needs

Need	**Priority**	**Concerns**	**Current Solution**	**Proposed Solutions**
Easy to use	High	Ability for users with little to no previous phone usage to navigate easily.	See proposed	Provide user friendly interface to navigate easily
Flexible(configurable)	Moderate	Ability to customize profile.	See proposed	Provide user with the facility to update record

3.7.6. Summary of Capabilities

Customer Benefit	**Supporting Features**
Enhanced ease of communication	Pushup Notifications.

Scalable	Support the controlling and monitoring of large number of users.
Convenient, flexible access to the system	Wireless access (cell phones).

3.8. Assumptions and Dependencies:

- It is assumed that user has internet connection.
- In using the onscreen keyboard, it is assumed that the user is literate and can type.
- The default language for the application shall be US English. It is assumed that users who cannot speak and write in English will not be using the text to speech features in the system, at least initially.
- It is also assumed that the network on the user's phone will be available.

3.9. Business opportunity

Normally people for arrange appointment with doctors, they go to the clinic to arrange an appointment or they call on the phone also for tracking sometimes they have to go and take by themselves so the application is providing an easier way to arrange appointments and tracking reports as well as for doctors this application provide easy interface to view the schedule and write reports, plus for receptionist as they are using manual way for recording the patient this application is providing for then easy way to record, manage and update appointments, patients and doctors.

As the world is growing, development and technology industry is expanding too. applications are easy to use so the thought for building up this application is to provide a solution is intended to develop a platform that will solve all the problems by arrange and managing clinics.

3.10. Use Case Diagram:

A **use case diagram** at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved. A use case diagram can identify the different types of users of a system and the different use cases and will often be accompanied by other types of diagrams as well. The use cases are represented by either circles or ellipses.

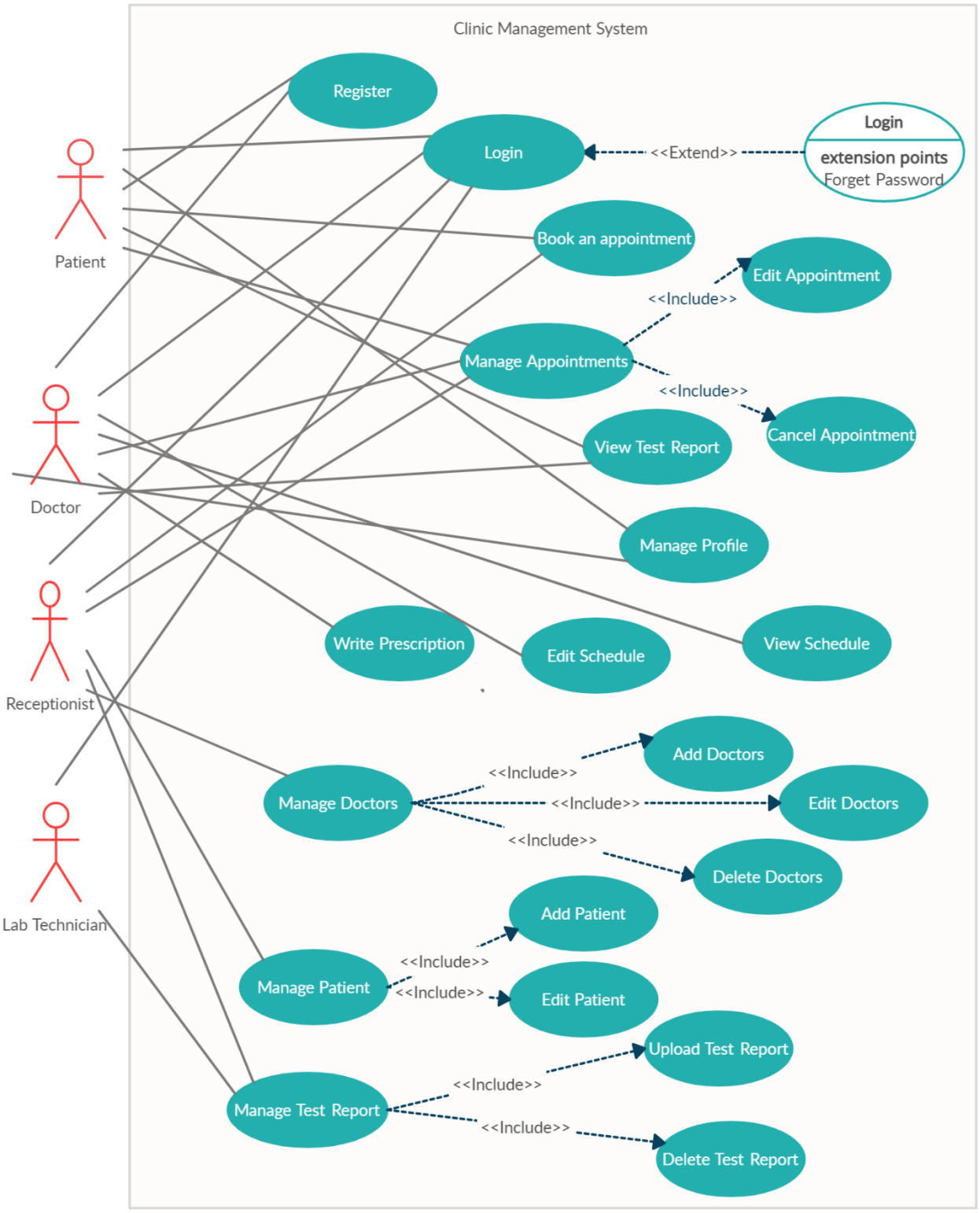
Followings are the components of Use case:

- • Boundary: A boundary defines the scope of the what the system will be.
- • Actor: perform a certain role in a use case.
- • Use cases: use cases are the roles played by the actor of the system.
- • Relationships: between the user of the system and the overall system.

In this Application the actors who interact with the system is either a admin, Doctor, Lab Technician or Patient.

3.System Analysis:

Use case Diagram:



3.10.1. Actor Goal List

Patient

- Registration
- login
- Book an appointment
- Manage Appointments
- View Test Report
- Manage profile
- Sign out

Doctors

- Registration
- Login
- View Schedule
- Edit Schedule
- Manage Appointments
- Write prescription
- View Test Report
- Manage profile
- Sign out

Lab Technician

- Registration
- Login
- Manage Test Report
- Write prescription
- Sign out

Receptionist

- Login
- Manage Appointments

3.System Analysis:

- Manage Patients
- Manage Doctors
- Sing out

3.11 Detailed Expanded format:

A use case is a methodology used in system analysis to identify, clarify, and organize system requirements.

Following are the use cases of this system.

User:

3.11.1. Sign **up**

UC- 001	
Title	Sign up
Primary Actor	Patients and Doctors.
Stakeholder and Interests	User wants to sign up
Preconditions	• Allowd internet premisson
Post Condition	Sign in Successfully.
Main Success scenario	
Actor's Action and Intension	**System Responsibility**
User signup into application by entering these required information • First Name	

• Last Name • Email • Password • Email • Gender • DOB	
	Application validates and displays the login page.
Alternative scenario	
If the user is already registered	
	Ask the user to Enter another email or try logging in with same email.
Frequency	High
Non-Functional Requirements	Security, Performance.

3.11.2. Login

UC- 002	
Title	Login
Primary Actor	Patients, Doctors and Receptionist.

3.System Analysis:

Stakeholder and Interests	User wants to login.
Preconditions	• Allowd internet premisson. • Already registerd.
Post Condition	Logged in Successfully.
Main Success scenario	
Actor's Action and Intension	**System Responsibility**
User Login into application by entering email and password.	
	User Login into application by entering email and password.
Alternative scenario	
User enters invalid details	
	Ask user to enter valid details.
Frequency	High
Non-Functional Requirements	Usability, Performance.

3.11.3. Book an Appointment.

UC- 003	
Title	Book an Appointment.
Primary Actor	Patients, and Receptionist.
Stakeholder and Interests	User wants to book an appointment.
Preconditions	• Allowd internet premisson. • User must be logged in.
Post Condition	Appointment booked Successfully.
Main Success scenario	
Actor's Action and Intension	**System Responsibility**
Patient or receptionist choose the doctor and time from the available list.	
	Appointment is sent to Doctor for confirmation.
Receives confirmation or rejection.	
Alternative scenario	
User enters invalid details	
	Ask user to enter valid details.
Frequency	High

Non-Functional Requirements	Usability, Performance.

3.11.4. Cancel Appointment.

UC- 004	
Title	Cancel Appointment.
Primary Actor	Patients, and Receptionist.
Stakeholder and Interests	User wants to canecl appointment.
Preconditions	• Allowd internet premisson. • User must be logged in. • Patient must have an appointment booked
Post Condition	Appointment Canceled Successfully.
Main Success scenario	
Actor's Action and Intension	**System Responsibility**
Patient or Receptionist choose cancel appointment option.	
	The appointment is cancelled successfully
Frequency	Medium
Non-Functional Requirements	Usability, Performance.

3.11.5. Edit Appointment.

UC- 005	
Title	Edit Appointment.
Primary Actor	Patients, and Receptionist.
Stakeholder and Interests	User wants to Edit appointment.
Preconditions	• Allowd internet premisson. • User must be logged in. • Patient must have an appointment booked
Post Condition	Appointment is edited Successfully.
Main Success scenario	
Actor's Action and Intension	**System Responsibility**
User choose Edit appointment option	
User edit the appointment.	
	The appointment is edtited successfully
Frequency	Medium
Non-Functional Requirements	Usability, Performance.

3.11.6. Manage Profile.

UC- 006	
Title	Manage profile.
Primary Actor	Patients, and Doctors.
Stakeholder and Interests	Doctor or Patient wants to Update profile.
Preconditions	• Allowd internet premisson. • User must be logged in.
Post Condition	Profile is Updated Successfully.
Main Success scenario	
Actor's Action and Intension	**System Responsibility**
User choose Edit profile option	
User update the profile.	
	The Profile is edtited successfully
Frequency	Medium
Non-Functional Requirements	Security, Performance.

3.11.7. View Test Report.

UC- 007	
Title	View Test Report.
Primary Actor	Patients, and Doctors.
Stakeholder and Interests	Doctor or Patient wants to view the test report.
Preconditions	• Allowd internet premisson. • User must be logged in. • Patient must have given the test
Post Condition	Report is viewed Successfully.
Main Success scenario	
Actor's Action and Intension	**System Responsibility**
User choose view test option.	
	Send the test report.
Report is viewed.	
Alternative scenario	
User choose view test option.	
	Test is not issued yet.
Frequency	Medium

Non-Functional Requirements	Usability, Performance.

3.11.8 View Schedule.

UC- 008	
Title	View Schedule.
Primary Actor	Doctors.
Stakeholder and Interests	Doctor wants to view schedule.
Preconditions	• Allowd internet premisson. • User must be logged in.
Post Condition	Profile is Updated Successfully.
Main Success scenario	
Actor's Action and Intension	**System Responsibility**
User choose Edit profile option	
User update the profile.	
	The Profile is edtited successfully
Frequency	Medium
Non-Functional Requirements	Security, Performance.

3.11.9 **Write prescriptions.**

UC- 009	
Title	Write prescriptions.
Primary Actor	Doctors.
Stakeholder and Interests	Doctor wants to write prescriptions.
Preconditions	• Allowd internet premisson. • User must be logged in. • Doctors must seen the patient
Post Condition	Prescription is written Successfully.
Main Success scenario	
Actor's Action and Intension	**System Responsibility**
Doctor open the target patient page.	
Choose write prescirption option.	
	The prscription is uploaded successfully
Frequency	Medium
Non-Functional Requirements	Security, Performance.

03.11.10 Add Patient.

UC- 011	
Title	Add Patient.
Primary Actor	Receptionest and Patient.
Stakeholder and Interests	Receptionest wants to add Patient.
Preconditions	• Allowd internet premisson. • User must be logged in.
Post Condition	Patient is added Successfully.
Main Success scenario	
Actor's Action and Intension	**System Responsibility**
Receptionist Clicks add Patient option.	
Entet these requried data: • First Name • Last Name • DOB • Gender • Email	
Receptionist click add patient.	
	The new entry of patient will be added.

Frequency	High
Non-Functional Requirements	Usability, Performance.

3.11.11 Edit Patient.

UC- 012	
Title	Edit Patient.
Primary Actor	Receptionest and Patient.
Stakeholder and Interests	Receptionest wants to edit Patient profile.
Preconditions	• Allowd internet premisson. • User must be logged in. • Patient is already registerd.
Post Condition	Patient is edited Successfully.
Main Success scenario	
Actor's Action and Intension	**System Responsibility**
Receptionist Clicks edit Patient option.	
Update the information.	
Receptionist click edit patient.	
	Update the information in database.

Frequency	High
Non-Functional Requirements	Usability, Performance.

3.11.12 Delete Patient.

UC- 013	
Title	Delete Patient.
Primary Actor	Receptionest and Patient.
Stakeholder and Interests	Receptionest wants to Remove Patient.
Preconditions	• Allowd internet premisson. • User must be logged in. • Patient is already registerd.
Post Condition	Patient is Removed Successfully.
Main Success scenario	
Actor's Action and Intension	**System Responsibility**
Receptionist Clicks Delete Patient option.	
	Ask for confirmation
Confirm the action	
	Remobe and upate the database.

Frequency	Low
Non-Functional Requirements	Usability, Performance.

3.11.13 Add Doctors.

UC- 014	
Title	Add Doctors.
Primary Actor	Receptionest and Doctors.
Stakeholder and Interests	Receptionest wants to add Doctor.
Preconditions	• Allowd internet premisson. • User must be logged in.
Post Condition	Doctor is added Successfully.
Main Success scenario	
Actor's Action and Intension	**System Responsibility**
Receptionist Clicks add Doctor option.	
Entet these requried data: • First Name • Last Name • Specialization • Clinic No. • Gender	

• Email	
Receptionist click add Doctor.	
	The new entry of Doctor will be added.
Frequency	High
Non-Functional Requirements	Usability, Performance.

3.11.14 Edit Doctor details.

UC- 015	
Title	Edit Doctor.
Primary Actor	Receptionest and Doctor.
Stakeholder and Interests	Receptionest wants to edit Doctor profile.
Preconditions	• Allowd internet premisson. • User must be logged in. • Doctor is already registerd.
Post Condition	Doctor is edited Successfully.
Main Success scenario	
Actor's Action and Intension	**System Responsibility**
Receptionist Clicks edit Doctor option.	

Update the information.	
Receptionist click edit Doctor.	
	Update the information in database.
Frequency	Medium
Non-Functional Requirements	Usability, Performance.

3.11.15 Delete Doctor.

UC- 016	
Title	Delete Doctor.
Primary Actor	Receptionest and Patient.
Stakeholder and Interests	Receptionest wants to Remove Doctor.
Preconditions	• Allowd internet premisson. • User must be logged in. • Doctor is already registerd.
Post Condition	Doctor is Removed Successfully.
Main Success scenario	
Actor's Action and Intension	**System Responsibility**
Receptionist Clicks Delete Doctor option.	

	Ask for confirmation
Confirm the action	
	Remove and upate the database.
Frequency	Low
Non-Functional Requirements	Usability, Performance.

3.11.16 Upload Test Report.

UC- 017	
Title	Upload Test Report.
Primary Actor	Lab Technician and Patient.
Stakeholder and Interests	Lab Technican wants to Upload test report.
Preconditions	• Allowd internet premisson. • User must be logged in. • Patient must have given the test.
Post Condition	Test report is uploaded Successfully.
Main Success scenario	
Actor's Action and Intension	**System Responsibility**
Lab technician Clicks update lab report option.	

Upload the test report.	
	Test is uploaded successfully.
Frequency	High
Non-Functional Requirements	Usability, Performance.

3.11.17 Delete Test Report.

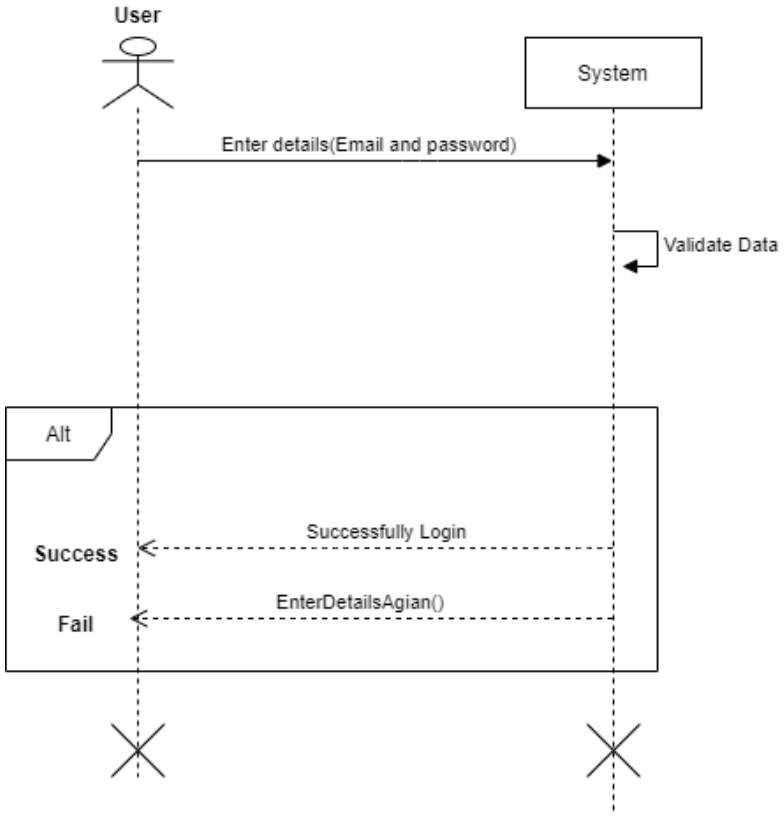
UC- 018	
Title	Delete Test Report.
Primary Actor	Lab Technician and Patient.
Stakeholder and Interests	Lab Technican wants to Delete test report.
Preconditions	• Allowd internet premisson. • User must be logged in. • Patient must have given the test. • Test is already uploaded.
Post Condition	Test report is removed Successfully.
Main Success scenario	
Actor's Action and Intension	**System Responsibility**
Lab technician Clicks delete lab report option.	
	Ask for confirmation

Confirm action.	
	Test is removed successfully and database is updated.
Frequency	Low
Non-Functional Requirements	Usability, Performance.

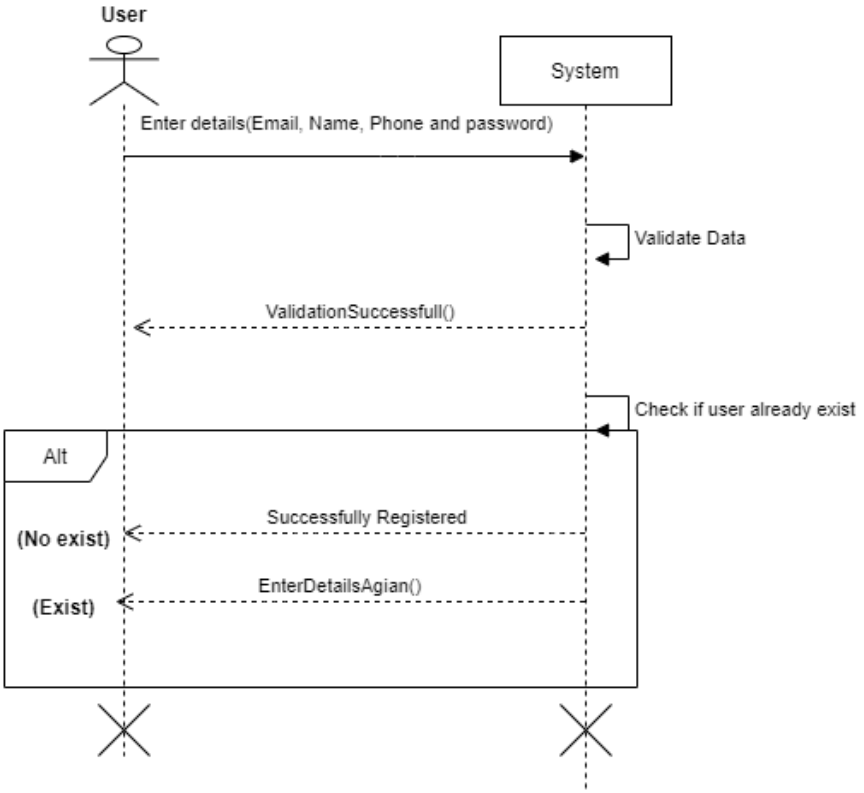
3.12 System Sequence Diagram

A system sequence diagram (SSD) is a sequence diagram that shows, for a particular scenario of a use case, the events that external actors generate their order, and possible inter-system event. System sequence diagrams are visual summaries of the individual use cases.

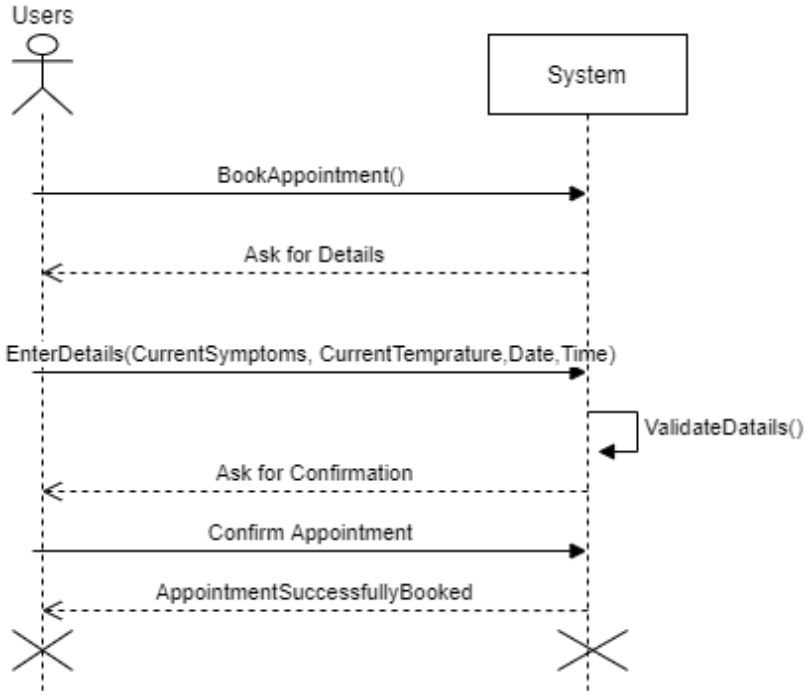
3.12.1. Login (Patient, Receptionist, Doctors and lab technician)



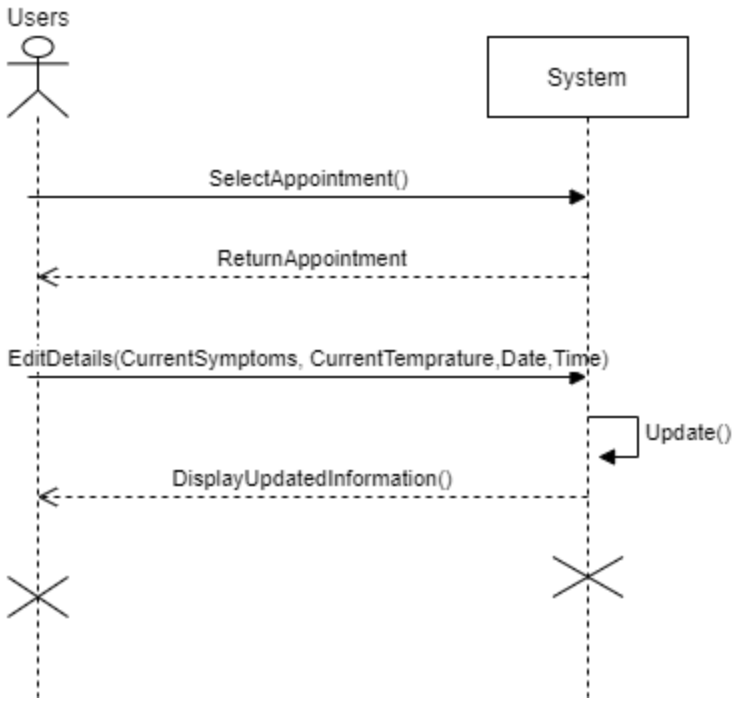
3.12.1.2. Sign Up (Patients, Doctors)



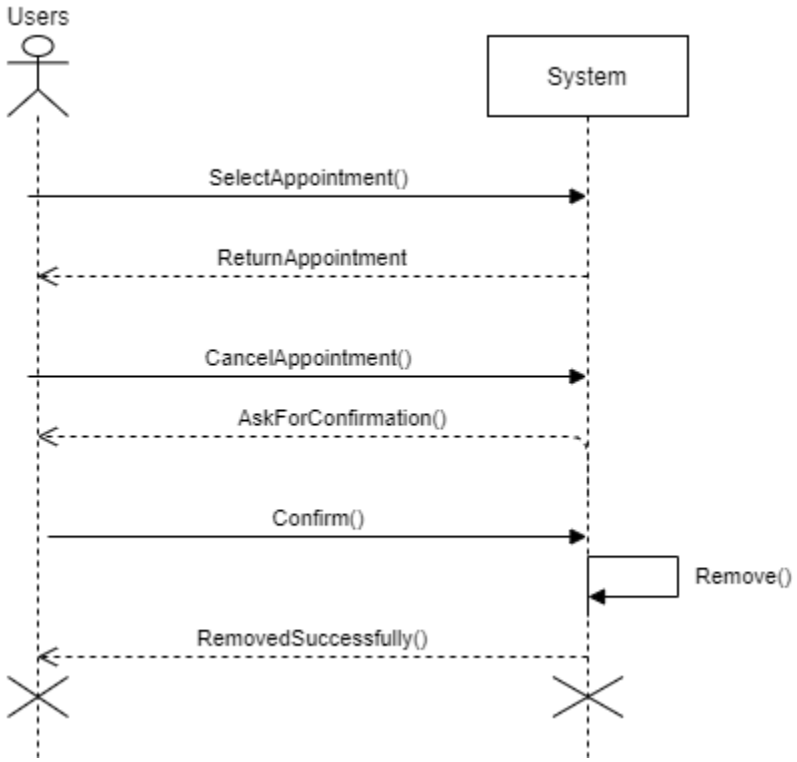
3.12.1.3. Book Appointment (Patients, Receptionist)



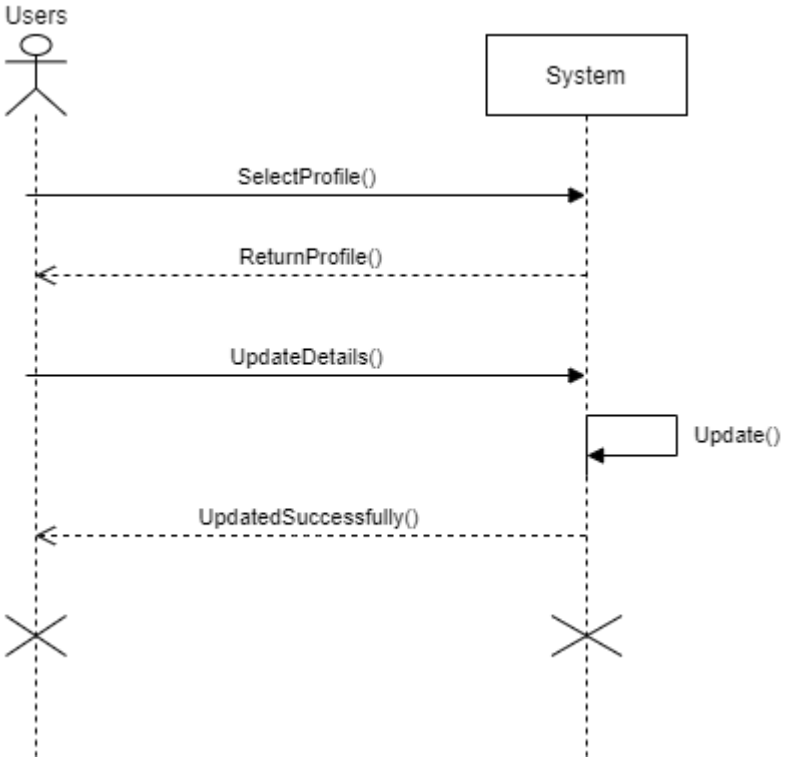
3.12.1.4. Edit Appointment (Patients, Receptionist and Doctor)



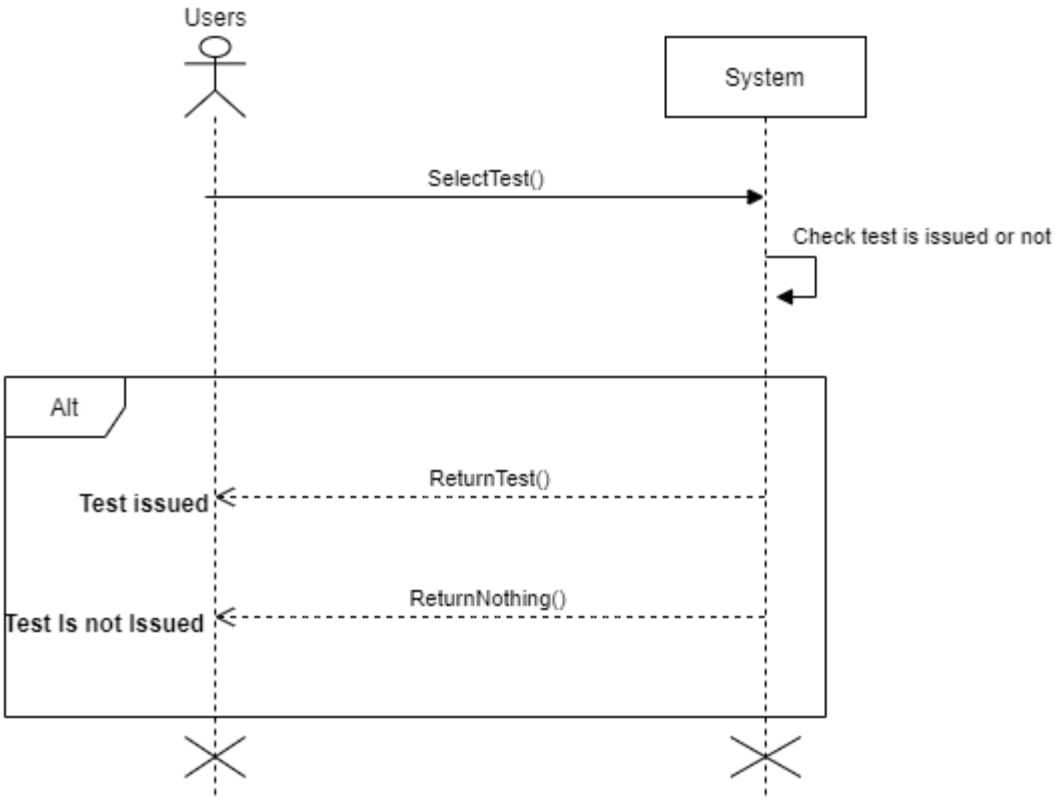
3.12.1.5. Cancel Appointment (Patients, Receptionist, and Doctor)



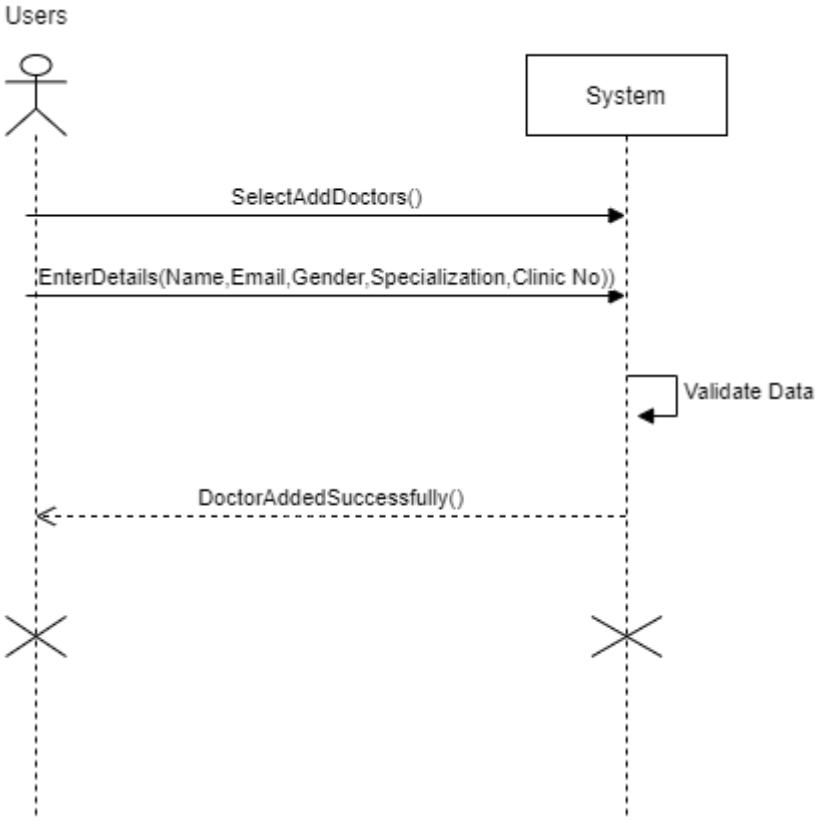
3.12.1.6 Edit profile (Patients, and Doctor)



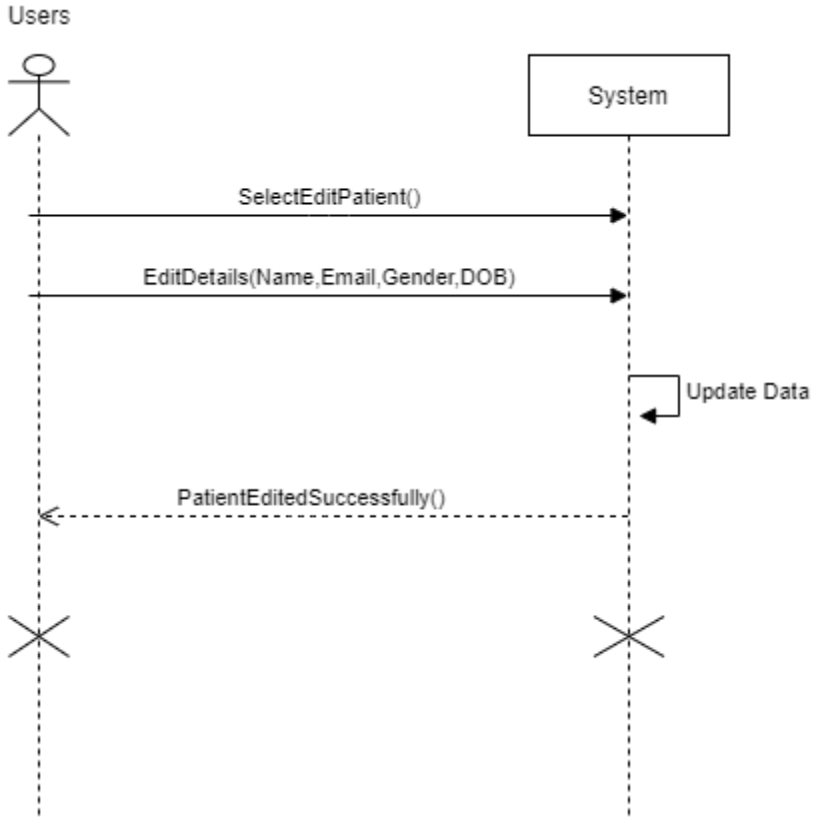
3.12.1.7 View Test Report (Patients, and Doctor)



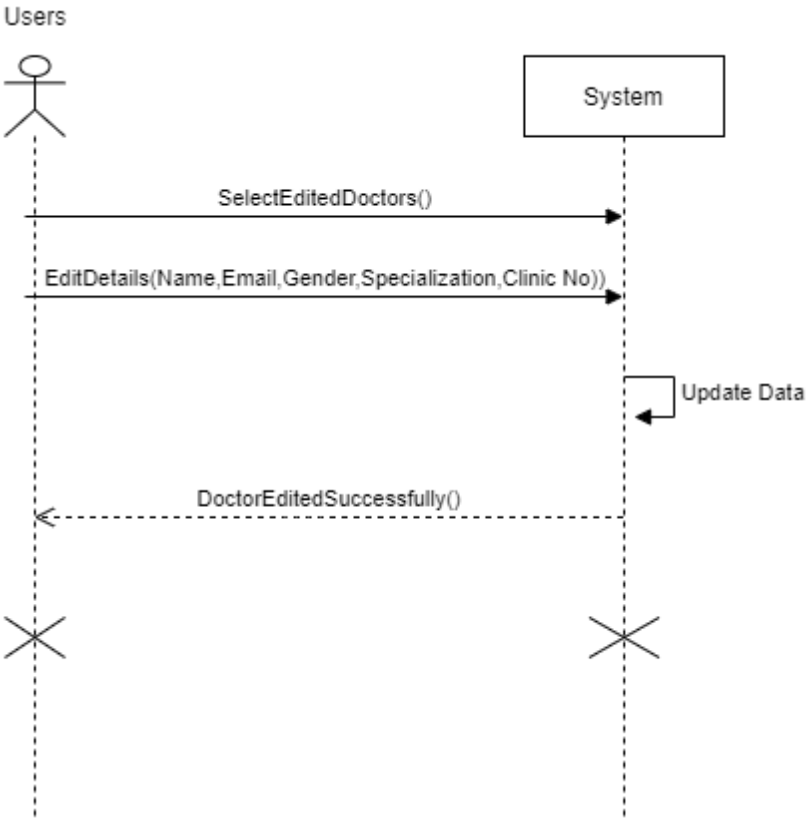
3.12.1.8 Add Patients (Receptionist)



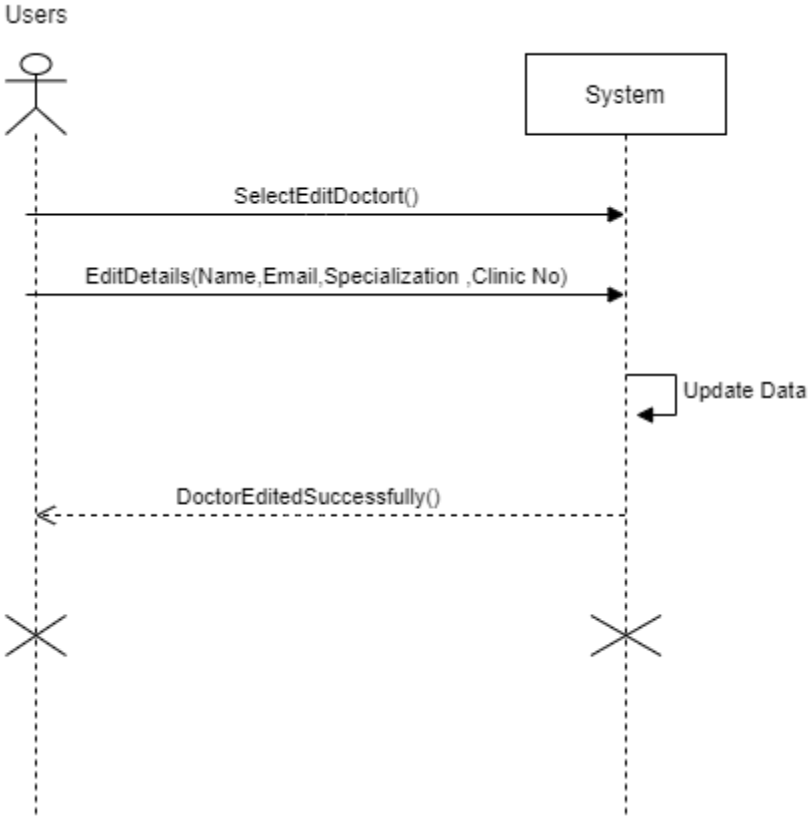
3.12.1.9 Edit Patients (Receptionist)



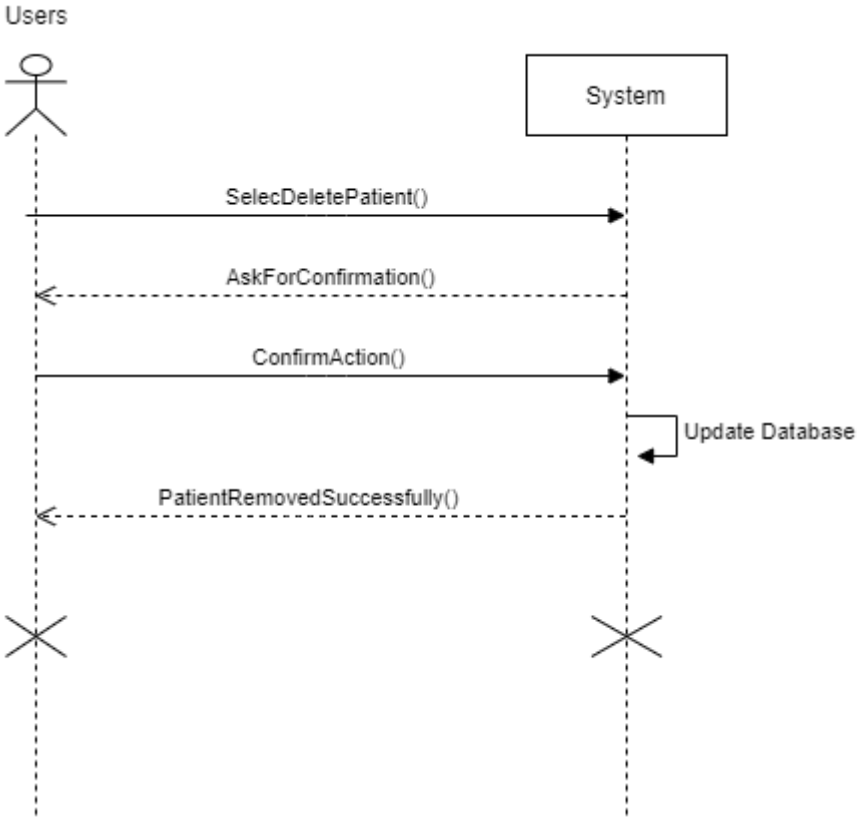
3.12.1.10 Add Doctors (Receptionist)



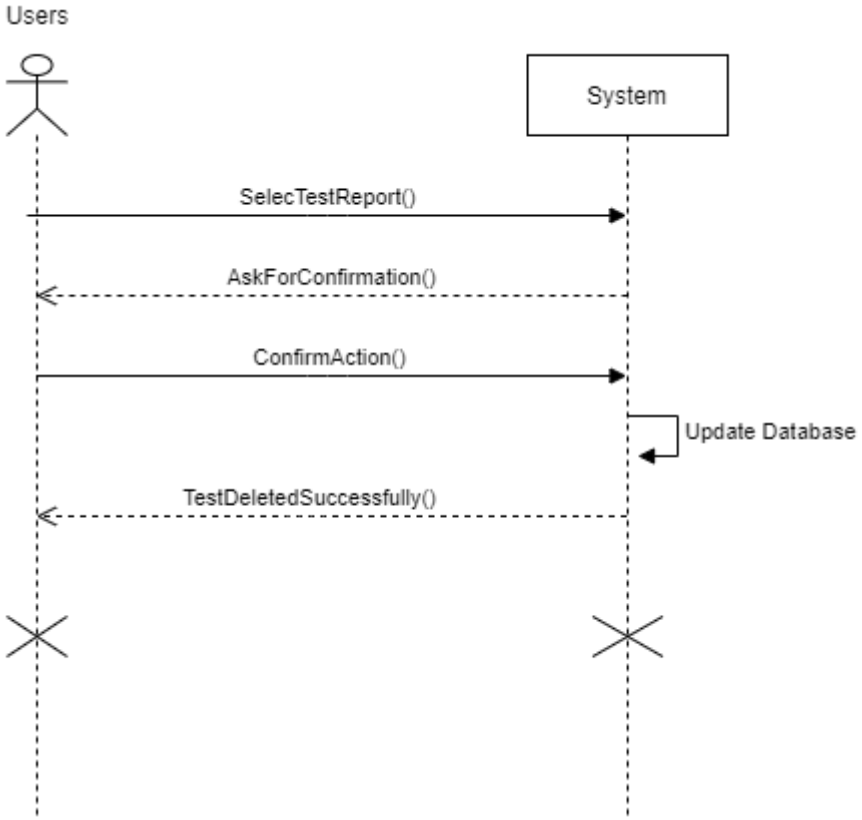
3.12.1.11 Edit Doctors (Receptionist)



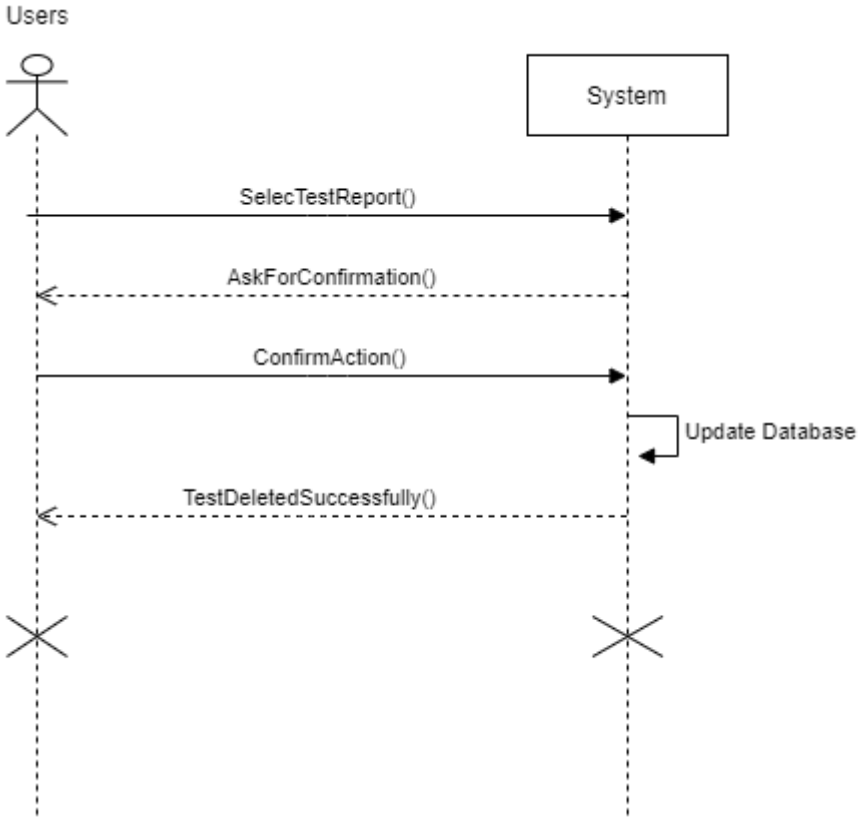
3.12.1.12 Delete Doctor (Receptionist)



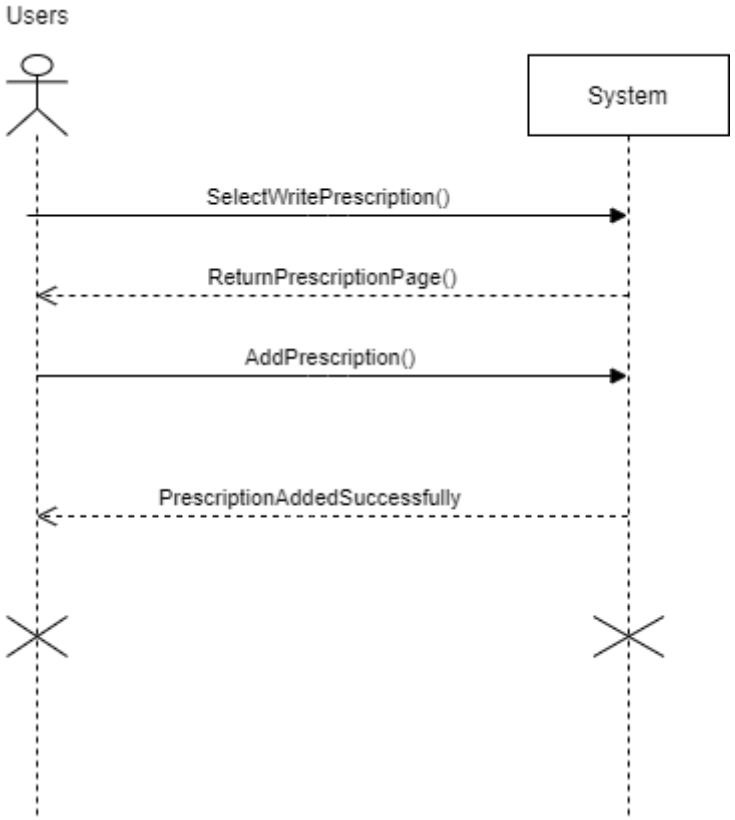
3.12.1.13 Delete Patient (Receptionist)



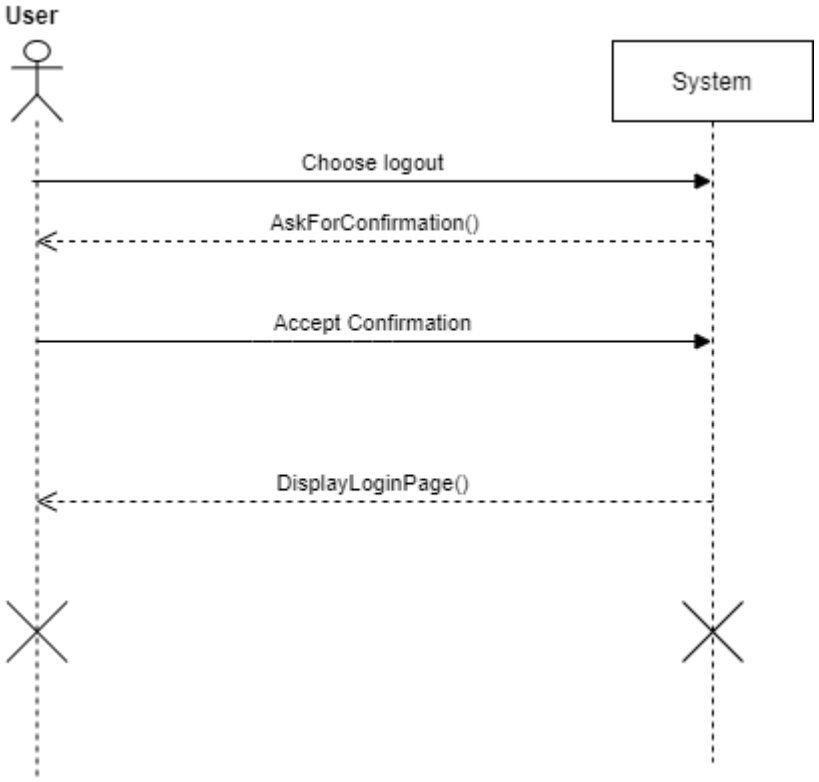
3.12.1.14 Delete Test (Lab Technician)



3.12.1.15 Write Prescription (Lab Technician)



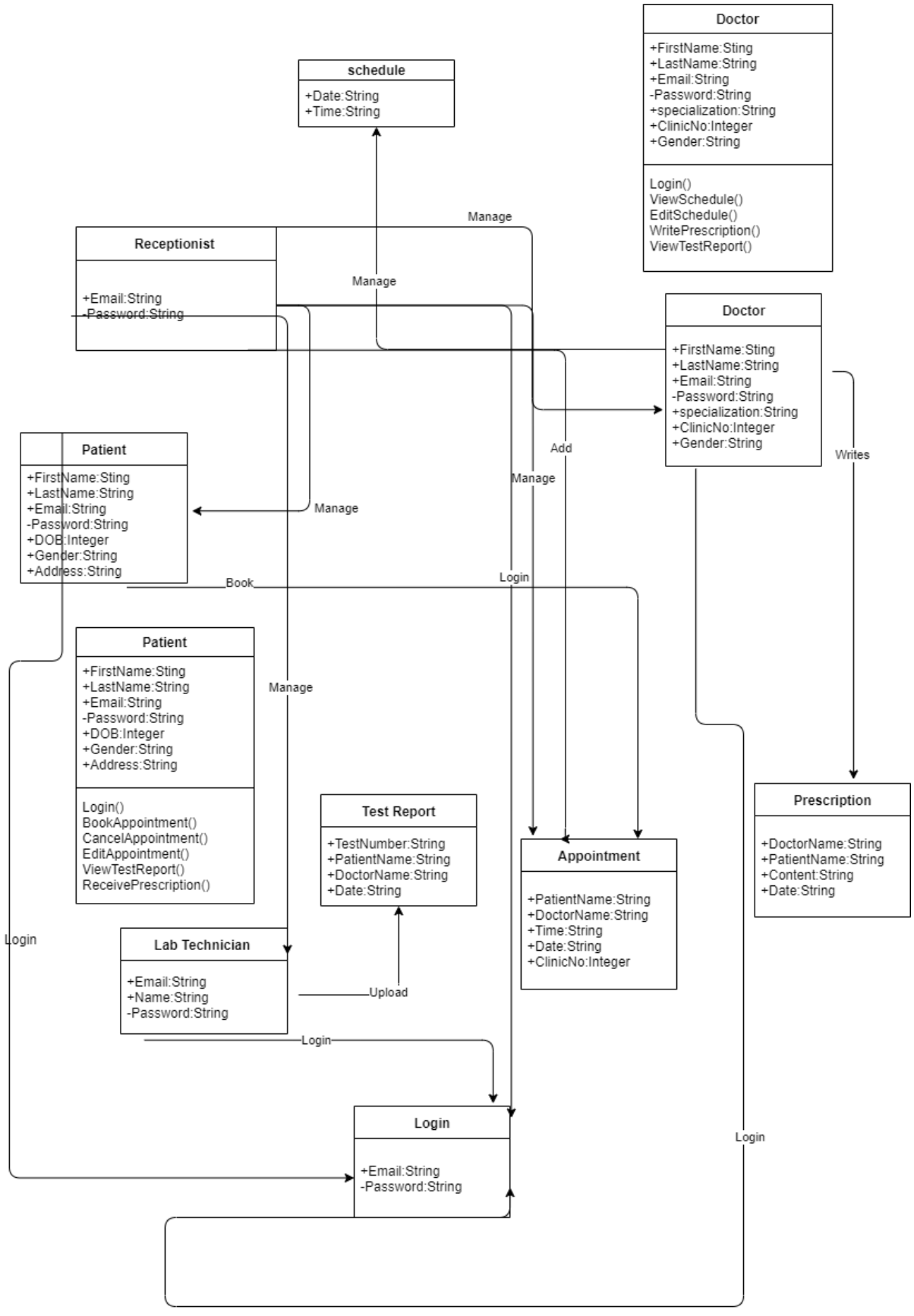
3.12.1.16 Logout (Patients, Doctors, Admin)



3.13. Domain Model

The Domain Model represent the vocabulary and key concepts of the problem domain and it identify the relationships among all of the entities within the scope of the domain.

3.System Analysis:



Chapter 4
System Design

4. System Design

Systems design is the process of defining the architecture, components, modules, interfaces, and data for a **system** to satisfy specified requirements. **Systems design** could be seen as the application of **systems** theory to product development

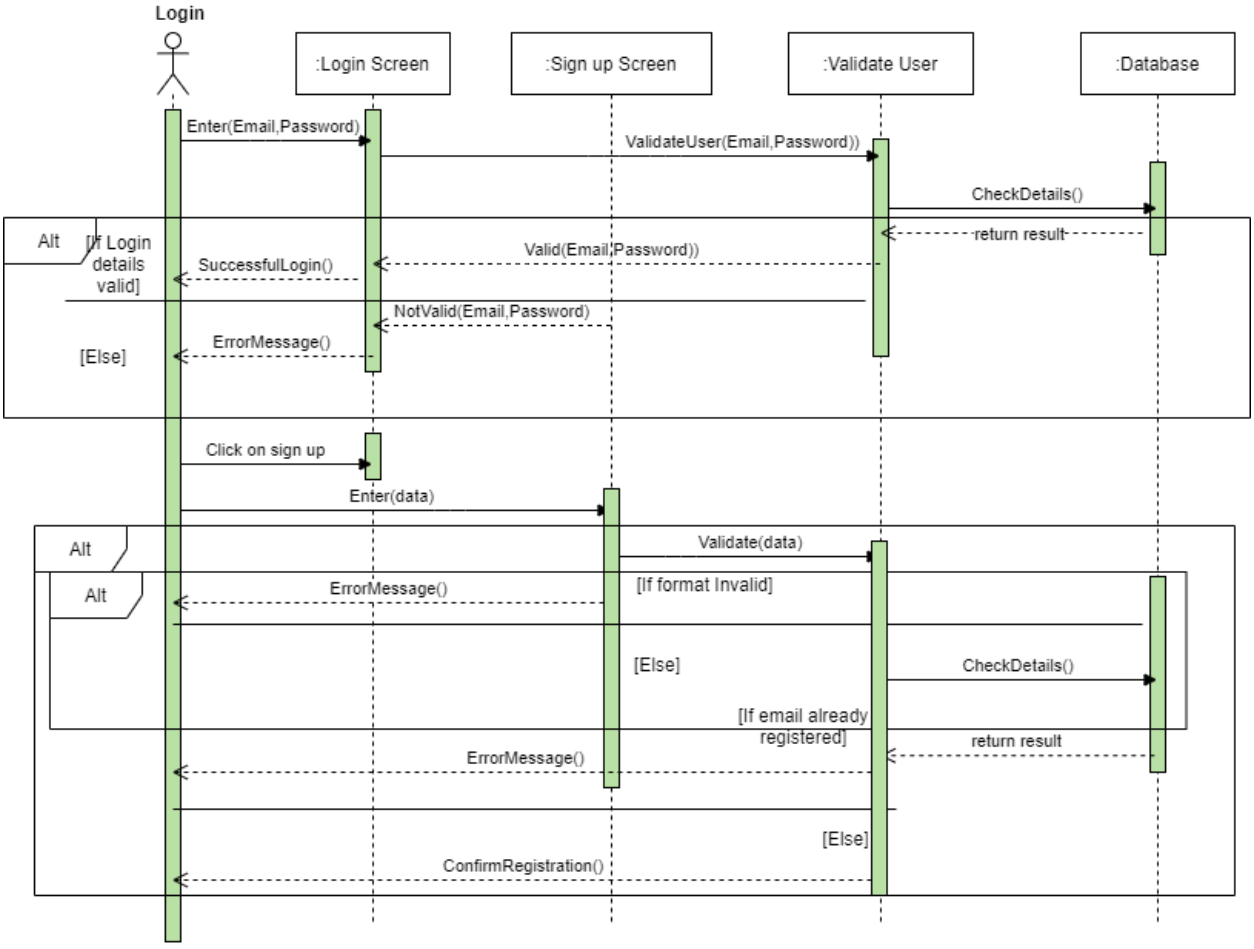
4.1. Interaction Diagram:

Interaction diagrams help you to visualize the interactive behaviour of a system. Interaction diagrams are used to represent how one or more objects in the system connect and communicate with each other.

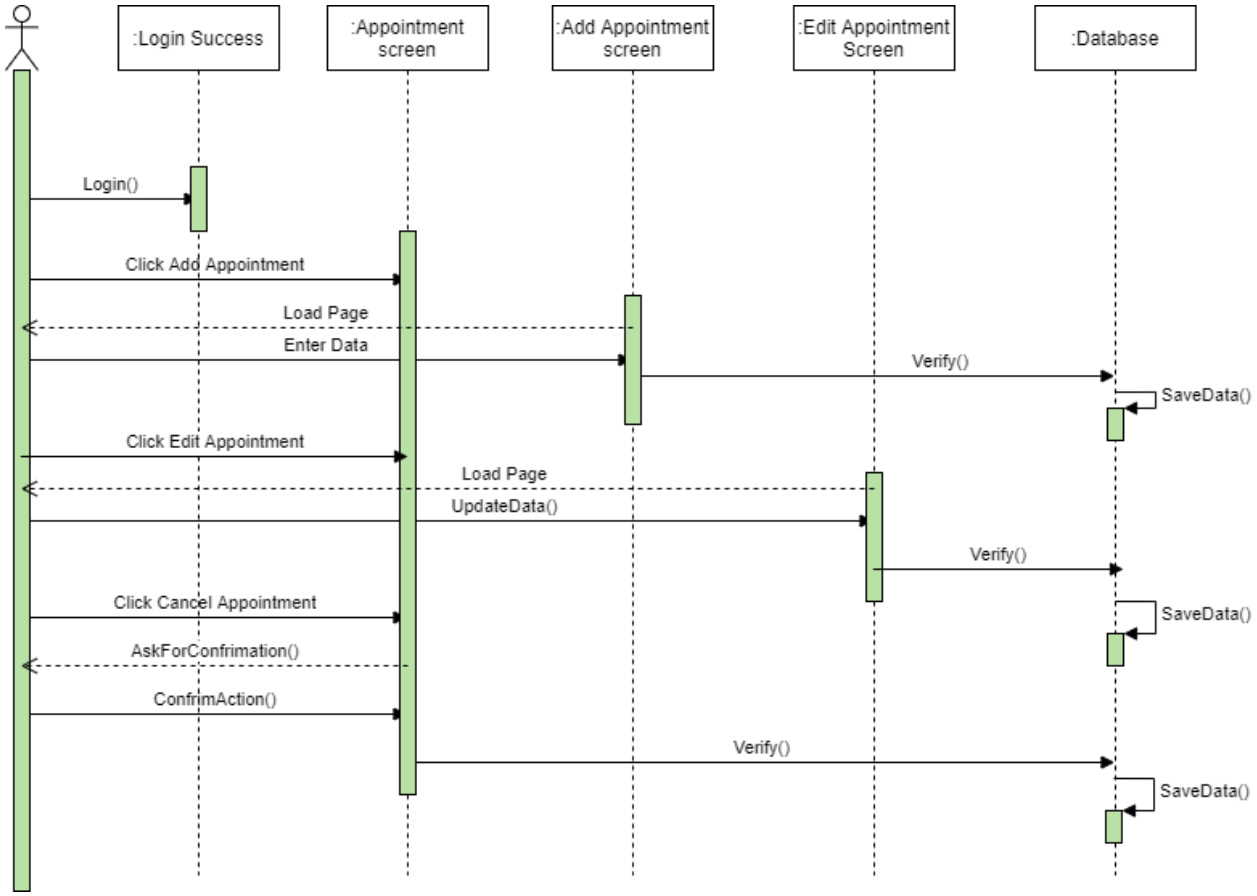
4.1.1. Sequence Diagram:

Sequence Diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development.

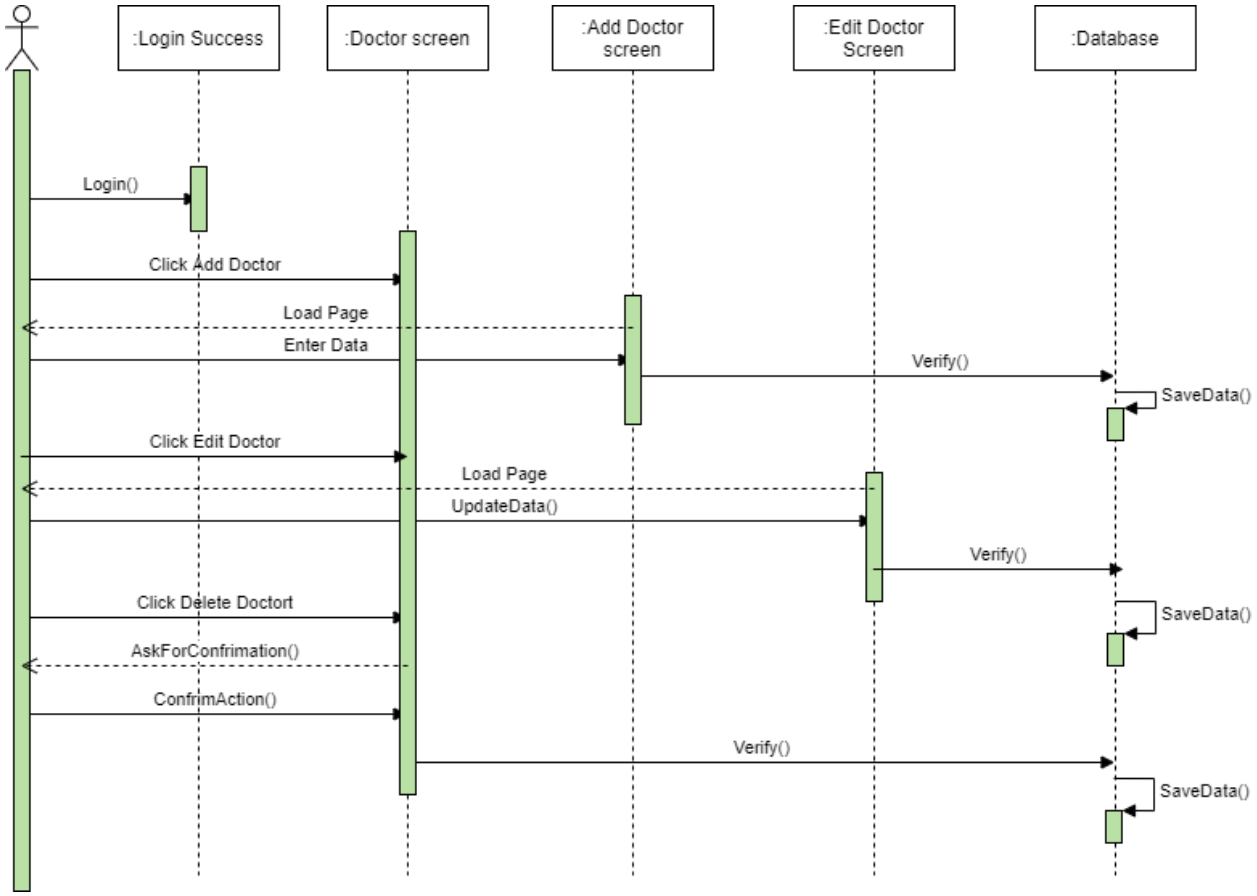
4.1.1.1. Login, and Signup:



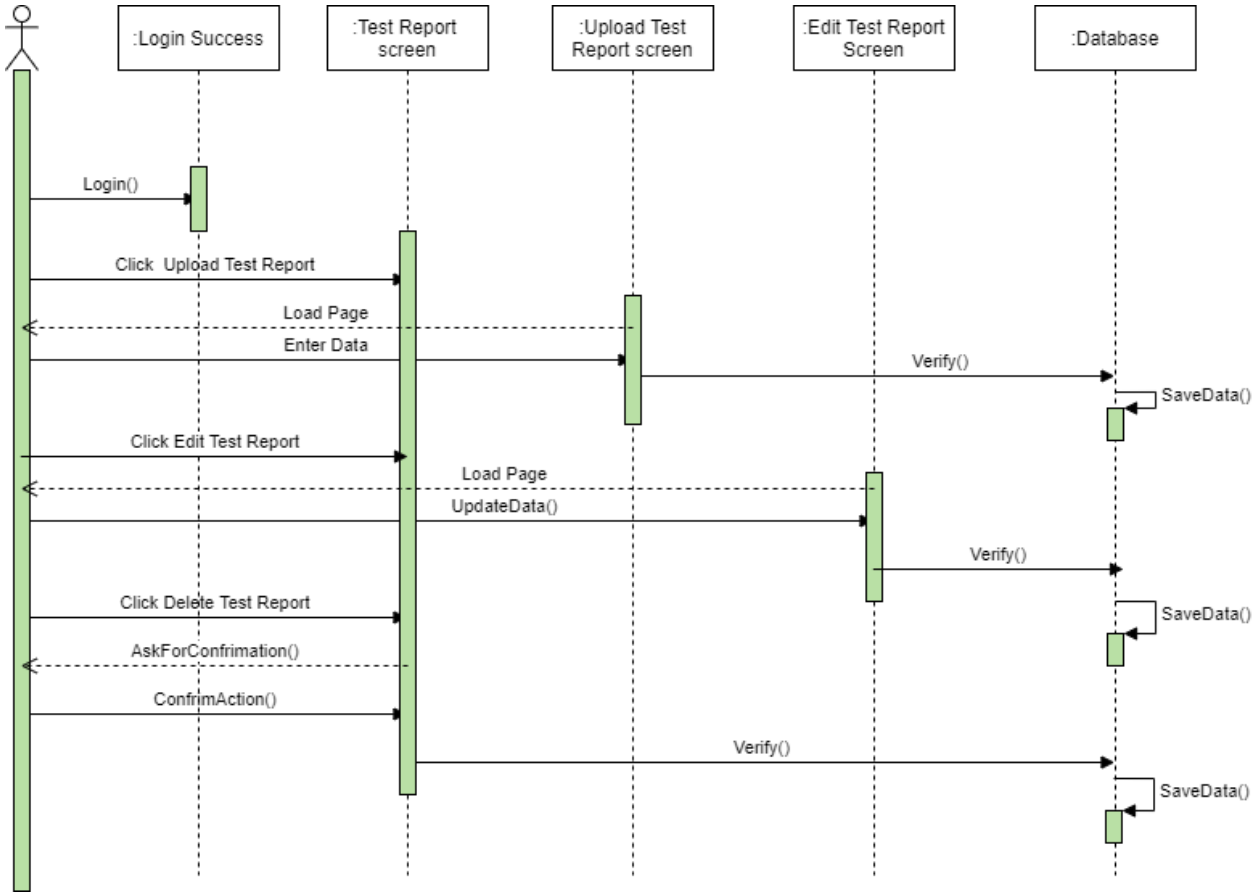
4.1.1.2. Add, Edit, and cancel Appointment:



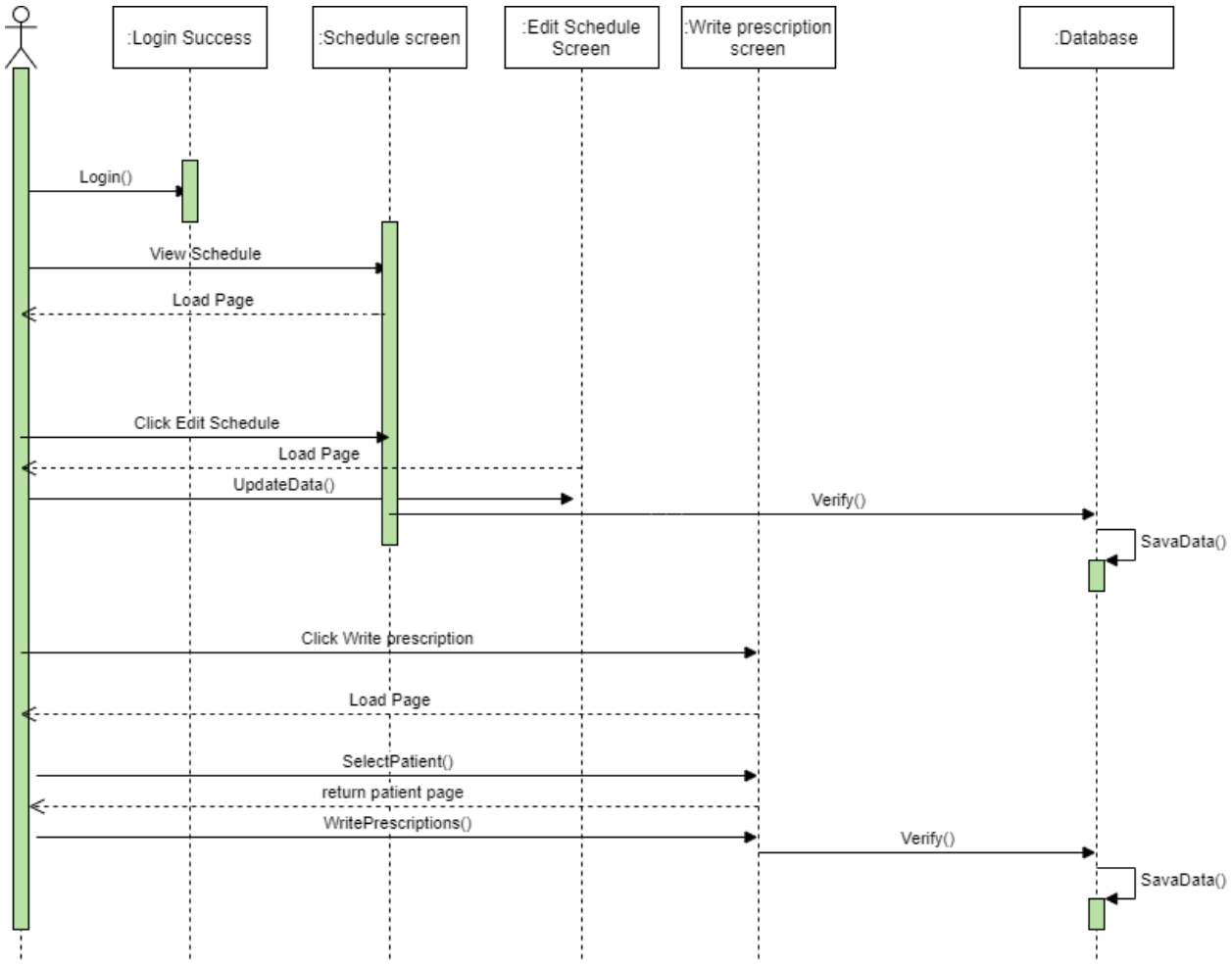
4.1.1.3. Add, Edit, and Delete Doctor user



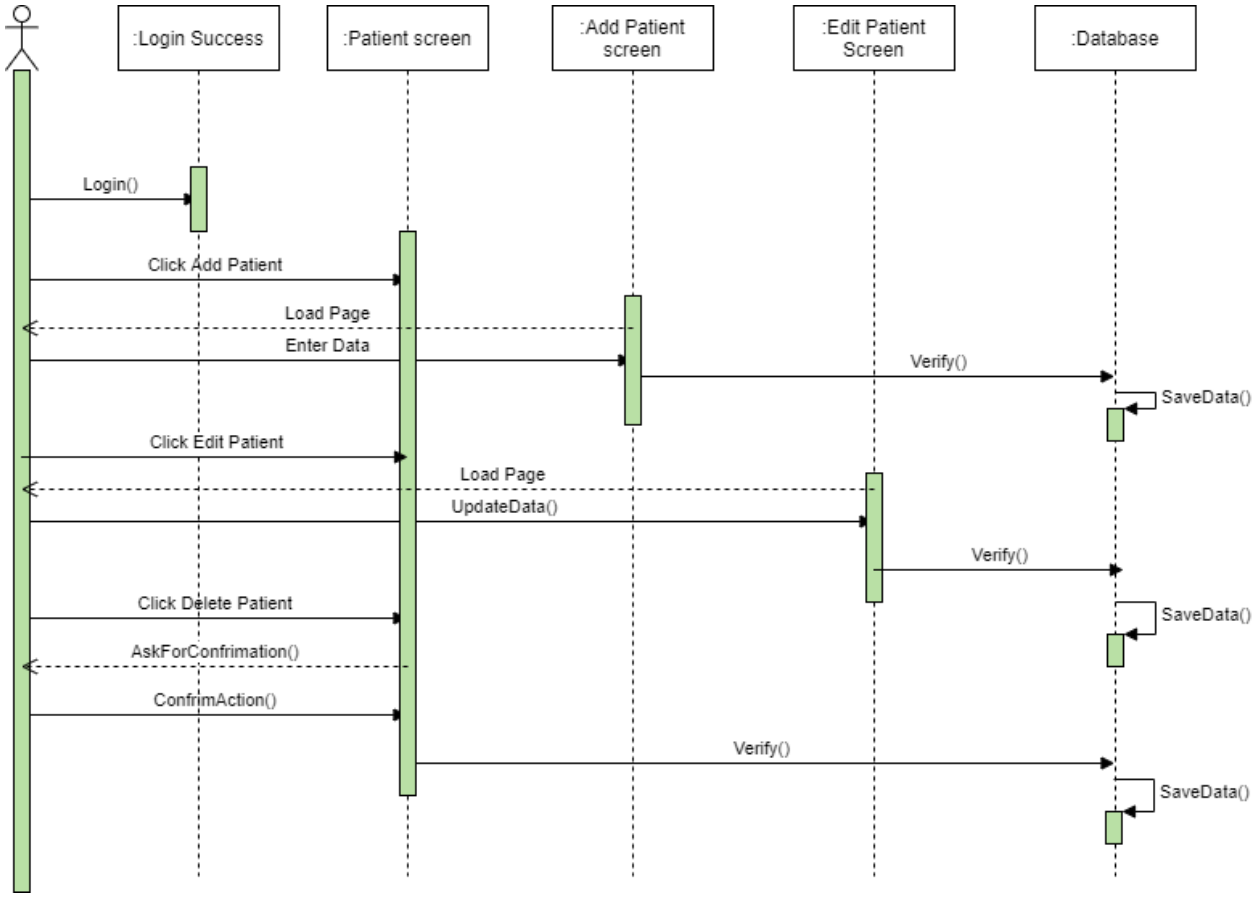
4.1.1.4. Add, Edit, and Delete Test Report



4.1.1.5. View, Edit Schedule and write prescription

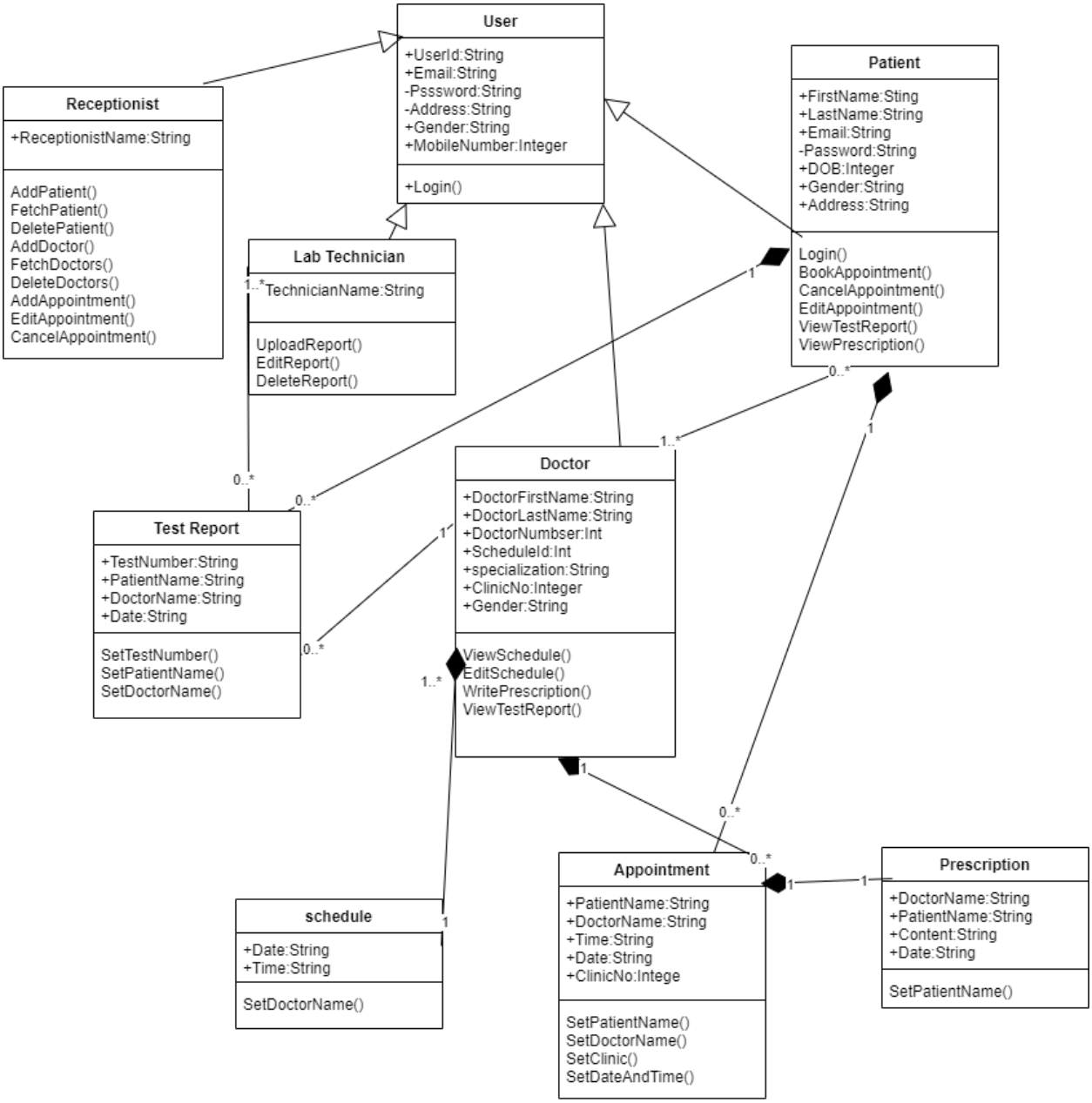


4.1.1.6. Add, Edit and Delete Patient user



4.1.2. Class Diagram:

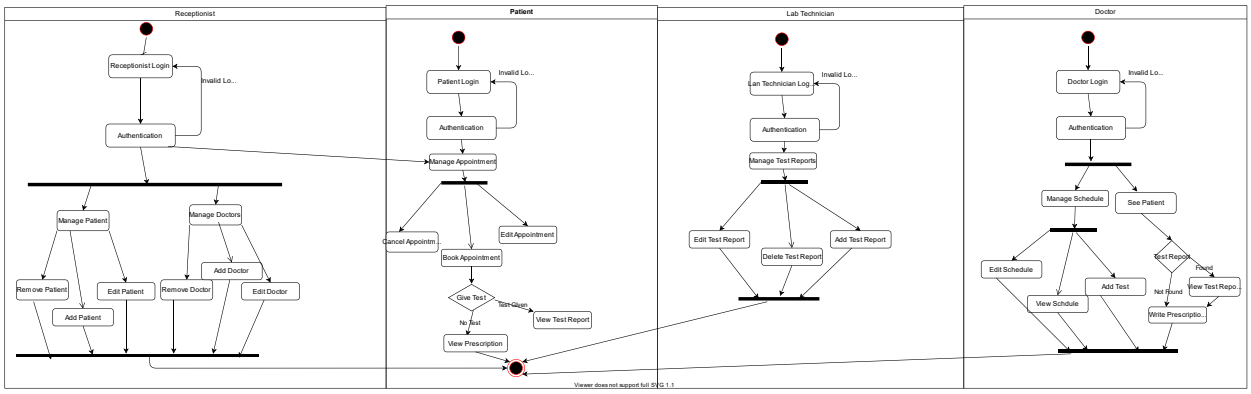
Class diagram is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects or classes.



4.2. Activity Diagram:

Activity diagrams are Graphical representations of workflows of stepwise activities and actions with the support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operations step by step workflows of components in a system.

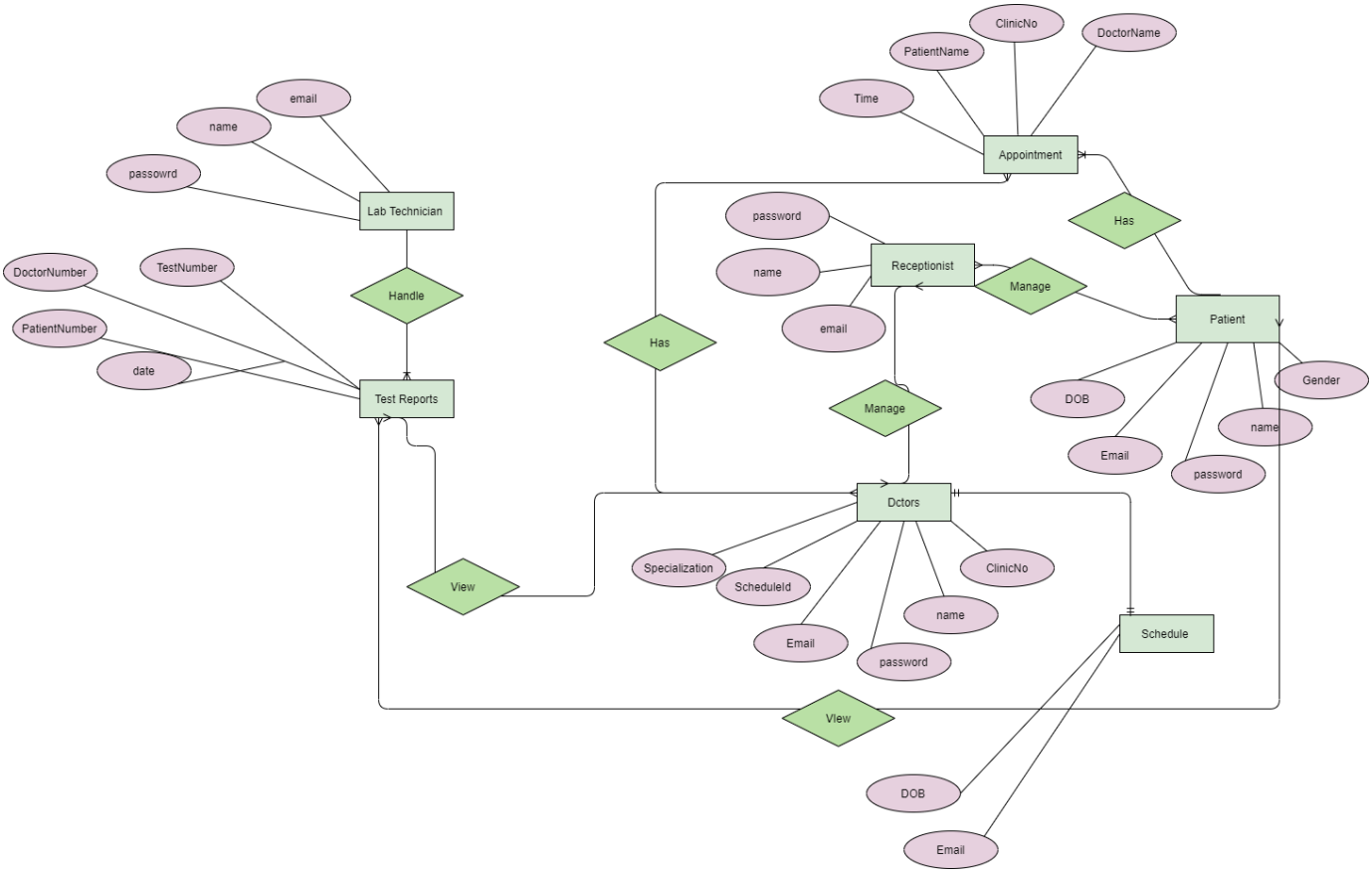
`



4.3. Entity Relationship Diagram:

An entity-relationship (ER) diagram is a specialized graphic that illustrates the interrelationships between entities in a database. ER diagrams often use symbols to represent different types of information. Boxes are commonly used to represent entities. Diamonds are normally used to represent relationships.

4. System Design



Chapter 5
Implementation

5. Implementation

Implementation of the project is given below:

5.1 Development Tools

5.1.1 Web Application

- Visual studio
- Microsoft SQL server management
- Bootstrap
- IIS web server

5.2 Reasons for selecting these tools:

5.2.1 Visual studio:

- Microsoft visual studio is used widely among developers, hence it easy to find solutions on internet.
- Microsoft visual studio is open source and free to use.
- Providing simple and friendly interface.
- One of the key features of Visual Studio Code is its great debugging support. VS Code's built-in debugger helps accelerate your edit, compile and debug loop.
- In visual studio there is thousands of extensions that helps in suiting the software for our need.

5.2.2 Microsoft SQL server management:

- SSMS is the Object Explorer, which allows the user to browse, select, and act upon any of the objects within the server.
- SQL Server Management Studio (SSMS) is used to manage both SQL Server instances and its databases with ease, efficiency and speed including out of the box security features.
- It is one of the most popular database management and server administration tools for SQL Server and Microsoft is constantly improving the features of the tool.

5.2.3 Bootstrap:

- Bootstrap is equipped with responsive layout and it made our web portal quite responsive.
- Bootstrap's responsive CSS adjusts to phones, tablets, and desktops.
- Bootstrap is compatible with all modern browsers (Chrome, Firefox, Internet Explorer, Edge, Safari, and Opera)

5.2.4 IIS web server:

- built-in authentication options such as Basic, ASP.NET, and Windows auth.
- ISS used commonly with ASP.NET.
- Speed up the website through built-in dynamic caching and enhanced compression.
- Maximize web security through a reduced server foot print and automatic application isolation.

5.3 Pseudo Code:

Pseudo code is a kind of structured English, written in natural language to describe a set of rules for the step by step process and operations for problem solving or other calculations. Pseudo code is considered as planning stage, written ahead of the syntax of corresponding computer language.

5.3.1 Pseudo code for Register/signup:

```
BEGIN
Open signup Page
Fill all the required fields
Checks format of fields
IF(all Checks == true)
{
        Connect to server( database)
        POST all Fields
```

```
        IF( EnteredEmail is already exist)
         DISPLAY Email already exist
  ElSE
   {
          Insert data in database
          DISPLAY Account Created
          Login Page
    }
 ELSE
    {
          IF(EnteredName== false)
          DISPLAY Enter valid name
          ENDIF
          IF(EnteredPhoneno= false)
          DISPLAY Enter valid Phone no
          ENDIF
          IF(EnteredEmail== false)
          DISPLAY Enter valid email
          ENDIF
       IF(EnteredPassword== false)
       DISPLAY Enter valid Password
       ENDIF
       IF(EnteredConfirmPassword==false)
       DISPLAY Password not Match
       ENDIF
       }
END
```

5.3.2 Pseudo code login:

```
IF(already Login== false)
{
```

```
Open Login Page
Enter EnteredEMAIL
Enter EnteredPASSWORD
GET EnteredEMAIL
GET EnteredPASSWORD
Connect to server (database)
POST EMAIL
POST PASSWORD
IF ( EnteredEMAIL==EMAIl AND EnteredPASSWORD=PASSWORD)
THEN
Login Successful
ELSE
  {
    IF (EMAIL != EnteredEMAIL)
    Error message: “Please enter valid email address”
    ENDIF
    IF (Password != EnteredPASSWORD)
    Error message: “Please enter valid Password”
    ENDIF
    IF (EnteredEMAIL is empty)
    Error message: “Please enter Email”
    ENDIF
    IF (EnteredPASSWORD is empty)
    Error message: “Please enter Password”
    ENDIF
    DISPLAy Login Failed
  }
 ELSE
 Home Page
END
```

5.3.3 Pseudo code for Booking appointment:

```
BEGIN
Open booking appointment Page
Fill all the required fields
Checks format of fields
IF(all Checks ==  true)
{
        Connect to server( database)
        POST all Fields

  ElSE
   {
          Insert data in database
          DISPLAY Appointment Booked successfully
          Home Page
    }
 ELSE
    {
          IF(DoctorName== false)
          DISPLAY Select Doctor name
          ENDIF
          IF(Time= false)
          DISPLAY Select valid Time
          ENDIF
      ENDIF
      }
END
```

5.3.4 Pseudo code for Adding Patient:

5. Implementation

```
BEGIN
Open Add Patient Page
Fill all the required fields
Checks format of fields
IF(all Checks ==  true)
{
        Connect to server( database)
        POST all Fields

  ElSE
   {
          Insert data in database
          DISPLAY Patient Added successfully
          Home Page
    }
 ELSE
    {
          IF(PattientName== false)
          DISPLAY Enter Patient name
          ENDIF
          IF(DOB= false)
          DISPLAY Enter DOB
          ENDIF
          IF(Gender= false)
          DISPLAY Select Gender
          ENDIF
          IF(Email= false)
          DISPLAY Enter Email
          ENDIF
          IF(MobileNumber= false)
          DISPLAY Enter Mobile Number
```

```
            ENDIF
            IF(Address= false)
            DISPLAY Enter Address
            ENDIF
            IF(CurrentMedication= false)
            DISPLAY Enter Current Medication
            ENDIF

        ENDIF
        }
END
```

5.3.5 Pseudo code for Adding Doctor:

```
BEGIN
Open Add Doctor Page
Fill all the required fields
Checks format of fields
IF(all Checks ==  true)
{
          Connect to server( database)
          POST all Fields

   ElSE
    {
            Insert data in database
            DISPLAY Doctor Added successfully
            Home Page
     }
 ELSE
```

```
    {
            IF(DoctorName== false)
            DISPLAY Enter Doctor name
            ENDIF
            IF(Specialization= false)
            DISPLAY Enter Specialization
            ENDIF
            IF(Gender= false)
            DISPLAY Select Gender
            ENDIF
            IF(Email= false)
            DISPLAY Enter Email
            ENDIF
            IF(MobileNumber= false)
            DISPLAY Enter Mobile Number
            ENDIF
            IF(ClinicNo= false)
            DISPLAY Enter ClinicNo
            ENDIF
            IF(Schedule= false)
            DISPLAY Enter Schedule
            ENDIF

        ENDIF
        }
END
```

5.3.6 Pseudo code for Update profile:

```
BEGIN
Open Home Page
```

```
Select Profile Option
Select edit profile
Update the required fields
Checks format of fields
IF(all Checks ==  true)
{
        Connect to server( database)
        POST all Fields

    }
END
```

Chapter 6
Testing

6.1 Testing

Software testing is an analysis conducted to provide information about quality of product with respect to the context in which it is intended to operate. Testing is the process of executing program with the intent of finding an error.

6.2 Test Cases

A test case is a set of conditions or variables under which a tester determine whether an application or software is working correctly or not.

6.2.1. Sign Up

Test case ID	TC-01
Associated Use Case:	UC-001 Sign Up
Functionality to be Tested	Successful Sign up
Actor	Patient, and Doctor.
Pre-conditions	System must be connected to internet

Sr.No.	**Action**	**Expected Results**	**Actual result**	**Pass/Fail**
1	User click on sign up button, user enters existing email. .	System will not show the user login screen and display error message.	System will display the error message that email already exists.	Pass

2	User click on sign up button, user enters email only.	System will not show the user login screen and display error message.	System will display the error message that enter all the fields to sign up.	Pass
3	User click on sign up button, user enters email and password and other details according to the format.	System will show the user login screen.	System will show the user login screen.	Passed
Post-condition:	User registered into system successfully.			

6.2.2. Log in

Test case ID		TC-02		
Associated Use Case:		UC-002 Login		
Functionality to be Tested		Successful Login		
Actor		Patient, Lab Technician, Doctor, and Receptionist.		
Pre-conditions		Patient, Lab Technician, Doctor, and Receptionist must be registered.		
Sr.No.	**Action**	**Expected Results**	**Actual result**	**Pass/Fail**

1	User click on login button, and types its right email and wrong password.	System will not allow the user to login.	System does not allow the user to login.	Pass
2	User enters wrong email with right password.	System will not allow the user to login.	System does not allow the user to login.	Pass
3	User enter wrong email and wrong password.	System will not allow the user to login.	System does not allow the user to login.	Passed
4	User enters right email and right password.	System will not allow the user to login.	System does not allow the user to login.	Pass
Post-condition:	User logged into system successfully.			

6.2.3. Book an Appointment

Test case ID	TC-03
Associated Use Case:	UC-003 Book an Appointment.
Functionality to be Tested	Successful adding an Appointment.
Actor	Patient, and Receptionist.
Pre-conditions	Patient, and Receptionist must be logged in.

Sr.No.	Action	Expected Results	Actual result	Pass/Fail
1	User click on book an appointment button, and leave fields empty	System will not allow the user to Book an appointment and will show warning message to fill fields.	System does not allow the user to book an appointment and message is shown .	Pass
2	User click on book an appointment button, and fill all fields.	System will book the appointment.	System booked the appointment	Pass
Post-condition:	Appointment booked and added into the system.			

6.2.4. Cancel an Appointment

Test case ID	TC-04
Associated Use Case	UC-004 cancel an Appointment.
Functionality to be Tested	Successful canceling an Appointment.
Actor	Patient, and Receptionist.
Pre-conditions	Patient, and Receptionist must be logged in, and Patient must have an appointment booked

Sr.No.	Action	Expected Results	Actual result	Pass/Fail
1	User click on cancel an appointment button, and will confirm canceling the appointment.	System will allow the user to cancel the appointment and will show confirmation message.	System allow the user to cancel the appointment and message is shown.	Pass
Post-condition:	Appointment canceled and updated in the system.			

6.2.5. Edit an Appointment

Test case ID		TC-05		
Associated Use Case		UC-005 Edit an Appointment.		
Functionality to be Tested		Successful Editing an Appointment.		
Actor		Patient, and Receptionist.		
Pre-conditions		Patient, and Receptionist must be logged in, and Patient must have an appointment booked		
Sr.No.	**Action**	**Expected Results**	**Actual result**	**Pass/Fail**
1	User click on edit an appointment button, and will change the date from the list.	System will allow the user to edit the appointment and	System allow the user to edit the appointment and date is changed.	Pass

		the date will be changed.		
2	User click on edit an appointment button, and will change the Doctor from the list.	System will allow the user to edit the appointment and the Doctor will be changed.	System allow the user to edit the appointment and date is changed.	Pass
Post-condition:	Appointment edited and updated in the system.			

6.2.6. Manage Profile

Test case ID		TC-06		
Associated Use Case		UC-006 Manage profile		
Functionality to be Tested		Successful editing profile.		
Actor		Patient, and Doctors.		
Pre-conditions		Patient, and Doctors must be logged in.		
Sr.No.	**Action**	**Expected Results**	**Actual result**	**Pass/Fail**
1	User click on profile button, and will view the profile.	User will be redirect it to profile page.	System redirect the user to profile page.	Pass

2	User click on edit profile button, and will update details.	System will allow the user to edit the profile and will apply the changes..	System allow the user to edit the profile and is changes is applied.	Pass
Post-condition:	Profile edited and updated in the system.			

6.2.7. View Test Report

Test case ID	TC-07
Associated Use Case	UC-007 View Test Report.
Functionality to be Tested	Successful Viewing Test Report.
Actor	Patient, and Doctors.
Pre-conditions	Patient, and Doctors must be logged in, and Patient must have given the test

Sr.No.	**Action**	**Expected Results**	**Actual result**	**Pass/Fail**
1	User click on View test report option.	User will be able to view or download the report.	User is able to view or download the report.	Pass
2	User click View test report option.	System will show message “” report is not ready yet and the expected date to	Message is shown as “” report is not ready yet and the expected date to view the report is nn/nn/nnn’’’	Pass

		view the report is nn/nn/nnn””		
Post-condition:	Test is viewed or expected date.			

6.2.8. View Schedule

Test case ID		TC-08		
Associated Use Case		UC-008 View Schedule.		
Functionality to be Tested		Successful Viewing Schedule.		
Actor		Doctor.		
Pre-conditions		Doctor must be logged in.		
Sr.No.	**Action**	**Expected Results**	**Actual result**	**Pass/Fail**
1	User click on View Schedule option.	User will be able to view the schedule.	User is able to view schedule.	Pass
Post-condition:	Schedule is viewed.			

6.2.9. Write prescription

Test case ID	TC-09
Associated Use Case	UC-009 Write prescription.
Functionality to be Tested	Successful writing prescription.
Actor	Doctor.

Pre-conditions		Doctor must be logged in, Doctors must seen the patient		
Sr.No.	**Action**	**Expected Results**	**Actual result**	**Pass/Fail**
1	User click on write prescription option. And write the prescription.	User will be able to write the prescription and will be added to the system.	User is able to write the prescription and prescription is added to the system.	Pass
Post-condition:	Prescription is written and added to the system.			

6.2.10. Send Notification

Test case ID		TC-10		
Associated Use Case		UC-010 Send Notification.		
Functionality to be Tested		Successful sending notification.		
Actor		Doctor, Patient and Receptionist.		
Pre-conditions		Receptionist must be logged in.		
Sr.No.	**Action**	**Expected Results**	**Actual result**	**Pass/Fail**
1	Receptionist click on send notification, write the message and chose the receiver is Doctor then click on send.	Receptionist will be able to write the message and Doctor will receives the message.	Receptionist is able to write the message and Doctor received the message.	Pass

2	Receptionist click on send notification, write the message and chose the receiver is Patient then click on send.	Receptionist will be able to write the message and Patient will receives the message.	Receptionist is able to write the message and Patient received the message.	Pass
Post-condition:	Notification is written, sent and added to the system.			

6.2.11. Add Patient

Test case ID	TC-11			
Associated Use Case	UC-011 Add Patient.			
Functionality to be Tested	Successful Adding Patient.			
Actor	Patient and Receptionist.			
Pre-conditions	Receptionist must be logged in.			
Sr.No.	**Action**	**Expected Results**	**Actual result**	**Pass/Fail**
1	Receptionist click on Add Patient option and leave fields empty	System will not allow to add the patient and will show warning message to fill fields.	System did not add the patient and message is shown.	Pass
2	Receptionist click on Add Patient	System will allow to add the	System added the patient.	Pass

	option and fill the fields.	patient and show patient is added.		
Post-condition:	Patient is added into the system successfully.			

6.2.12. Edit Patient

Test case ID		TC-12		
Associated Use Case		UC-012 Edit Patient.		
Functionality to be Tested		Successful Editing Patient.		
Actor		Patient and Receptionist.		
Pre-conditions		Receptionist must be logged in, Patient is already registered.		
Sr.No.	**Action**	**Expected Results**	**Actual result**	**Pass/Fail**
1	Receptionist click on Edit Patient option and update the fields	System will allow to edit the patient and will update patient's details.	System allowed updating the patient and details updated.	Pass
Post-condition:	Patient is edited and system is updated successfully.			

6.2.13. Delete Patient

Test case ID	TC-13
Associated Use Case	UC-013 Delete Patient.

Functionality to be Tested		Successful Deleting Patient.		
Actor		Patient and Receptionist.		
Pre-conditions		Receptionist must be logged in, Patient is already registered.		
Sr.No.	**Action**	**Expected Results**	**Actual result**	**Pass/Fail**
1	Receptionist click on Delete Patient option and confirm the action.	System will allow to Delete the patient.	System allowed Deleting the patient.	Pass
Post-condition:	Patient is Deleted and system is updated successfully.			

6.2.14. Add Doctor

Test case ID		TC-14		
Associated Use Case		UC-014 Add Doctor.		
Functionality to be Tested		Successful Adding Doctor.		
Actor		Doctor and Receptionist.		
Pre-conditions		Receptionist must be logged in.		
Sr.No.	**Action**	**Expected Results**	**Actual result**	**Pass/Fail**
1	Receptionist click on Add Doctor option and leave fields empty	System will not allow to add the Doctor and will show warning	System did not add the Doctor and message is shown.	Pass

		message to fill fields.		
2	Receptionist click on add Doctor option and fill the fields.	System will allow to add the Doctor and show Doctor is added.	System added the Doctor.	Pass
Post-condition:	Doctor is added into the system successfully.			

6.2.15. Edit Doctor

Test case ID		TC-15		
Associated Use Case		UC-015 Edit Doctor.		
Functionality to be Tested		Successful Editing Doctor.		
Actor		Doctor and Receptionist.		
Pre-conditions		Receptionist must be logged in, Doctor is already registered.		
Sr.No.	**Action**	**Expected Results**	**Actual result**	**Pass/Fail**
1	Receptionist click on Edit Doctor option and update the fields	System will allow to edit the Doctor and will update Doctor details.	System allowed updating the Doctor and details updated.	Pass
Post-condition:	Doctor is edited and system is updated successfully.			

6.2.16. Delete Doctor

Test case ID		TC-16		
Associated Use Case		UC-016 Delete Doctor.		
Functionality to be Tested		Successful Deleting Doctor.		
Actor		Doctor and Receptionist.		
Pre-conditions		Receptionist must be logged in, Doctor is already registered.		
Sr.No.	**Action**	**Expected Results**	**Actual result**	**Pass/Fail**
1	Receptionist click on Delete Doctor option and confirm the action.	System will allow to Delete the Doctor.	System allowed Deleting the Doctor.	Pass
Post-condition:	Doctor is Deleted and system is updated successfully.			

6.2.17. Upload Test Report

Associated Use Case		UC-017 Upload Test Report.		
Functionality to be Tested		Successful Uploading Test Report.		
Actor		Lab Technician and Patient.		
Pre-conditions		Lab Technician must be logged in, and Patient must have given the test.		
Sr.No.	**Action**	**Expected Results**	**Actual result**	**Pass/Fail**

1	Lan technician click on upload test and will upload the test.	Lab technician will be able to update the test and patient will receive the notification.	System allowed uploading the test and patient received the message.	Pass
2	Lan technician click on upload test and will upload nothing the test.	Lab technician will not be able to update the test and warning message shown to update file.	System did not allow uploading the test and message is shown.	Pass
Post-condition:	Test is uploaded and added into system.			

6.2.18. Delete Test Report

Associated Use Case	UC-0177 Delete Test Report.
Functionality to be Tested	Successful Deleting Test Report.
Actor	Lab Technician.
Pre-conditions	Lab Technician must be logged in, and test is uploaded.

Sr.No.	**Action**	**Expected Results**	**Actual result**	**Pass/Fail**
1	Lan technician click on Delete test and will confirm the action.	Lab technician will be able to delete the test after confirmation..	System allowed Deleting the test.	Pass

Post-condition:	Test is Deleted and System is updated.

Chapter 7
Conclusion

7. conclusion

Nowadays computing business is important to keep business competing in the market, people get attached to things connected to their phones and computers because we use them all day, so in this system, I tried to compute the business of clinics to save time, effort, and keep patient attached to the clinic, as well as making tasks easier for staff, Considering Covid-19 pandemic this period we try to avoid going out and interacting with others so this system helps to avoid avoiding interacting with people, especially for clinics and hospital it is important to avoid going there much to avoid Infectious diseases, the following results achieved by persistent learning and applying what I have learned in my four years degree program.

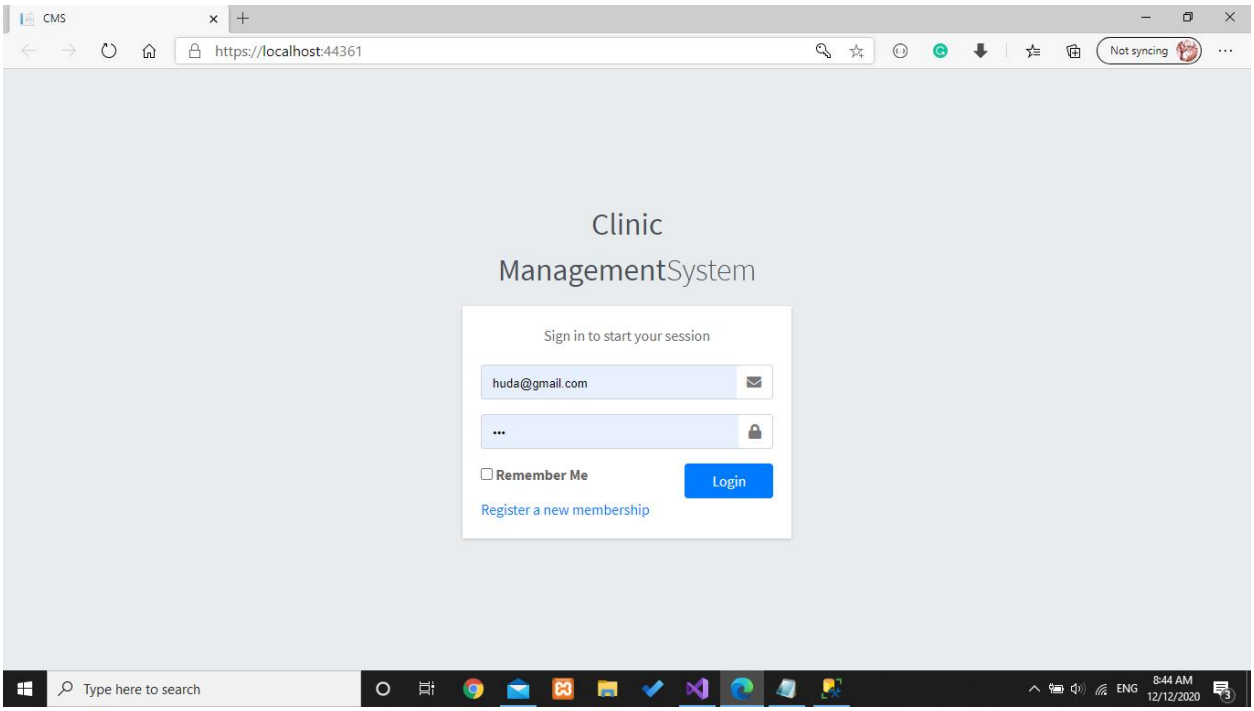
7.2 Future Enhancement

The scope for future will involve adding more feature for the benefit for patient, doctor and staff Following features can be added in future work

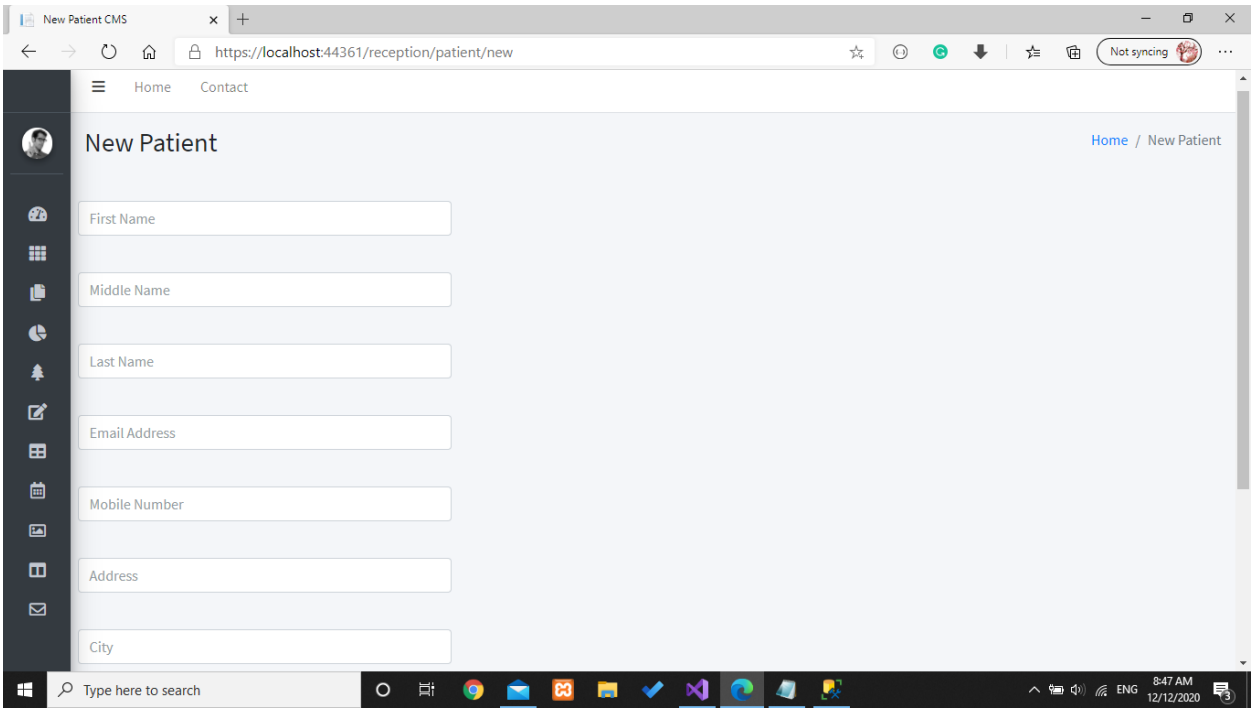
- Online Payment
- Online buying of medicines
- consulting doctor online

Appendix A

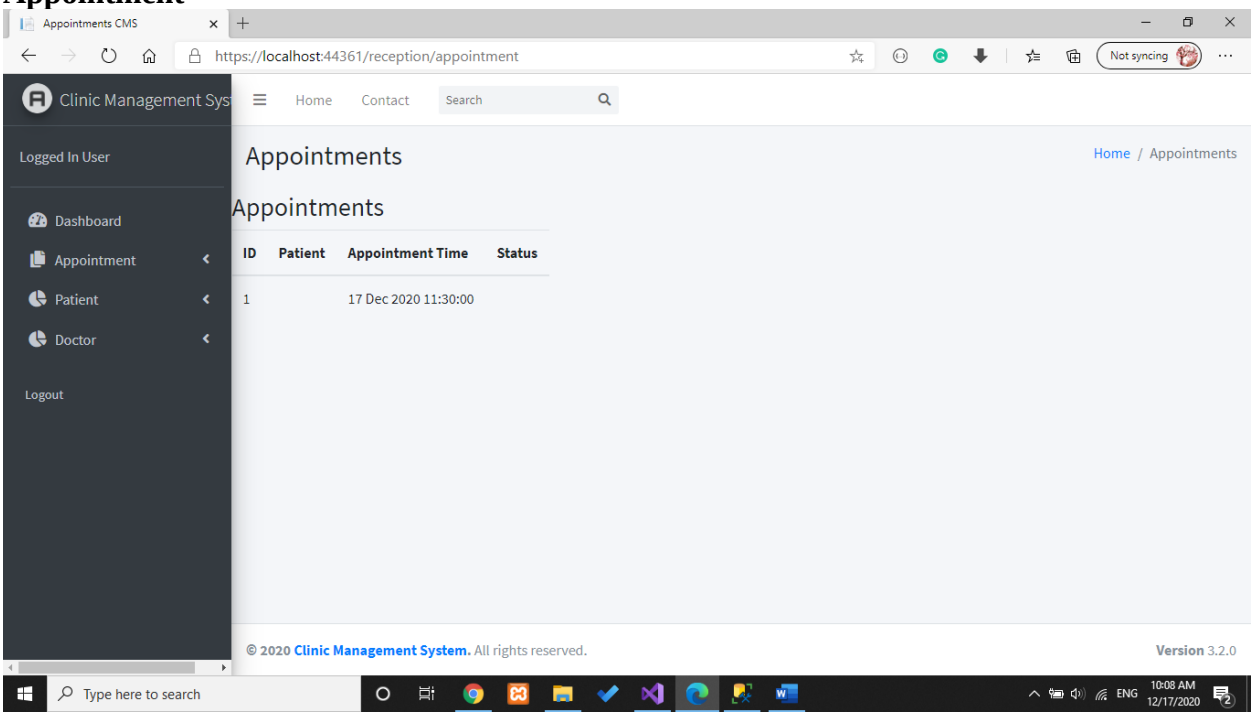
Login



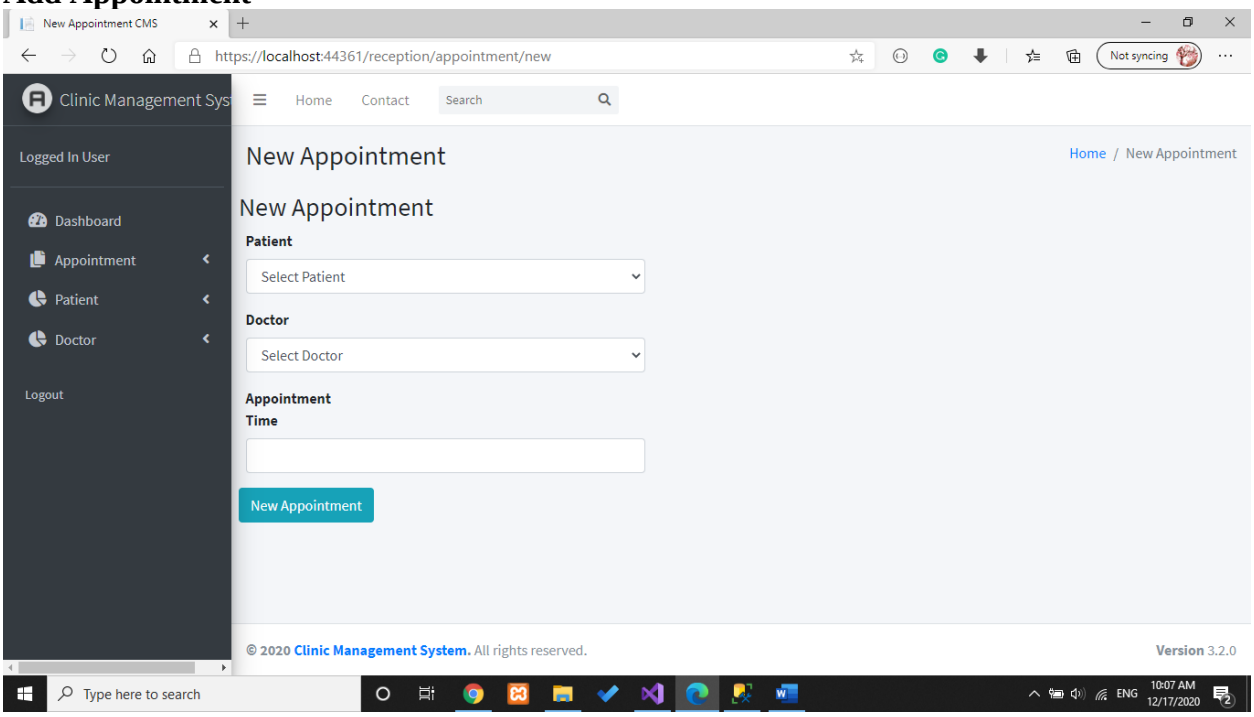
Register new patient



Appointment

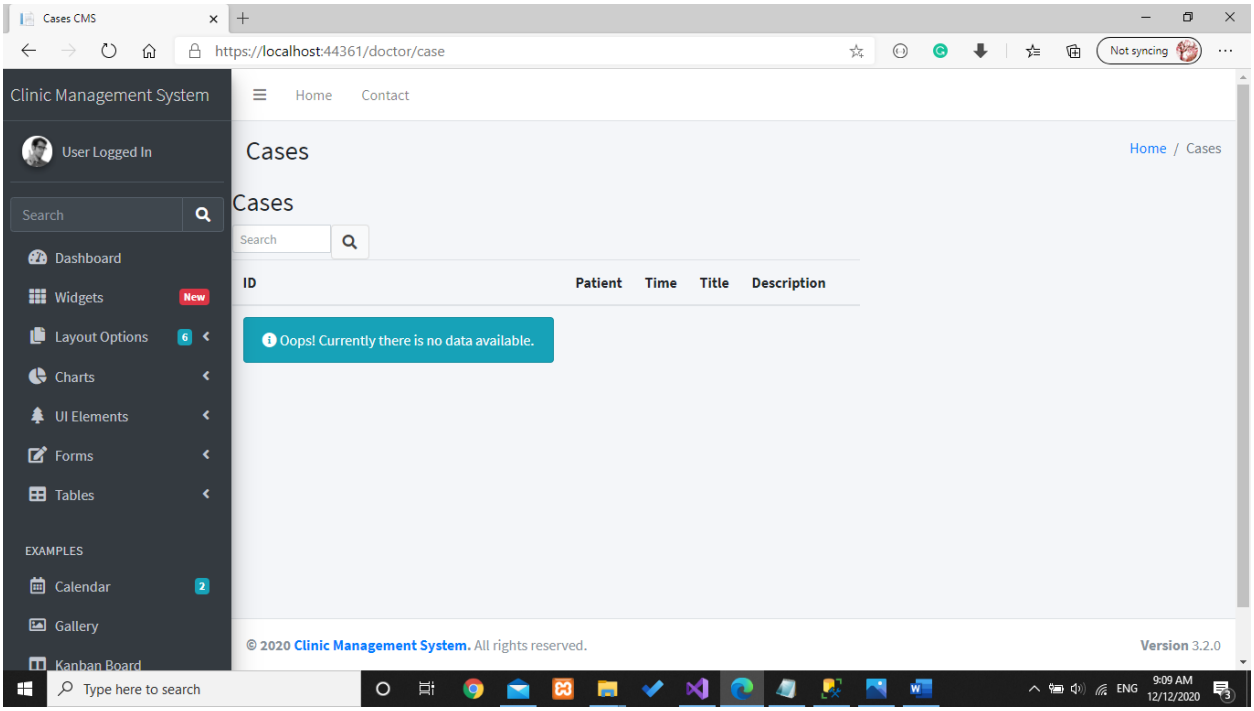


Add Appointment



Doctor side: Case

7. conclusion



Appendix B

7. conclusion

References:

- diagrams.net
- **https://app.creately.com/diagram**
- Introduction · Bootstrap v5.0 (getbootstrap.com)
- Free Bootstrap Admin Template | AdminLTE.IO
- ASP.NET documentation | Microsoft Docs
- SQL Server Management Studio (SSMS) - SQL Server Management Studio (SSMS) | Microsoft Docs
- Visual Studio product family documentation | Microsoft Docs