ARBA MINCH UNIVERSITY

INSTITUTE OF TECHNOLOGY

Faculty of Computing and Software Engineering

Department of Software Engineering

Doctor Appointment App for Arba Minch General Hospital

Group Members	ID. No
Hawi Kaba ……	NSR/1269/12
Rahel Getachew ……	NSR/1971/12
Sisay Fikadu ……	NSR/2171/12
Sorawak Abaya ……	NSR/2195/12
Tewodros Mekuriya ……	NSR/2313/12

Advisor: Mr. Melaku M.

A Senior Project Proposal

Submitted to Department of Software Engineering, Computing & Software Engineering faculty, AMIT, Arba Minch University, in Partial fulfillment for the requirement of the Degree of Bachelor Science in Software Engineering.

Arba Minch, Ethiopia

June 2024

Submitted by:

		Signature	Date
1.	Hawi Kaba	_____	_____
2.	Rahel Getachew	_____	_____
3.	Sisay Fikadu	_____	_____
4.	Sorawak Abaya	_____	_____
5.	Tewodros Mekuriya	_____	_____

Approved by:

Advisor Name: Mr. Melaku Meskele _____ _____

Signature Date

Declaration

We hereby affirm that this project is our original work and has not been submitted for a degree at any other institution.

Declared by:

		Signature	Date
1.	Hawi Kaba	_____	_____
2.	Rahel Getachew	_____	_____
3.	Sisay Fikadu	_____	_____
4.	Sorawak Abaya	_____	_____
5.	Tewodros Mekuriya	_____	_____

School: Faculty of Computing and Software Engineering

Program: Software Engineering

Project subject: Doctor Appointment Application

I certify that this project satisfies all the requirements as a project for the degree of Bachelor of Science.

_____ _____

Name of program coordinator Signature

This is to certify that I have read this project and that in my opinion, it is fully adequate, in scope and quality, as a project for the degree of Bachelor of Science.

Advisor Name: _____ _____ _____

Examiner Name: _____ _____ _____

Signature Date

ACKNOWLEDGEMENTS

First and foremost, we extend our heartfelt gratitude to the Almighty for blessing us with the strength and guidance throughout our lives and during the course of this project. We recognize that without divine assistance, none of this would have been possible.

We also wish to express our deep appreciation to our advisor, Melaku Meskele, for his invaluable guidance, unwavering support, and dedication in devoting his time to assist us in completing this endeavor.

Additionally, we are indebted to the faculty members and former fellow students of the Faculty of Computing and Software Engineering for their invaluable insights, suggestions, and encouragement throughout the project.

We would like to extend our sincere appreciation and gratitude to Arba Minch General Hospital and its esteemed doctors for their invaluable provision of crucial information during we collect information about their appointment service.

Lastly, we extend our respect and gratitude to all individuals involved in various aspects of our project. Their contributions are invaluable to the ongoing success of our project.

Table of Contents

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

Technology has indeed played a significant role in streamlining processes, revolutionizing industries (i.e. healthcare, finance, education, etc.), and making life easier for everyone involved. However, nowadays Ethiopian healthcare institutions (Hospitals) are performing patient appointment processes or operations in a traditional awkward way. Due to manual processing of the appointment processes they have faced the following challenges: such as missed appointments, inefficient appointment scheduling, long wait times, inefficient resource allocation, poor communication, limited accessibility, and increased health costs. To overcome the aforementioned challenges, we will develop a mobile and web-based doctor appointment application to address these challenges and improve the overall healthcare experience for both patients and hospitals. The proposed system, a doctor's appointment application for hospitals will make a paradigm shift in the healthcare setting. Ultimately, it will enhance the appointment booking process, revolutionize the way patients access healthcare services and improve patient experience, remind appointment time via text message, help doctors manage their schedules efficiently, and provide administrators with valuable insights for optimizing services of the hospital's doctors. In generally, this project aims to modernize appointment management, improve patient satisfaction, and ultimately enhance healthcare delivery at Arba Minch General Hospital.

Keywords: Doctors appointment, Hospital, Streamline process, Patient, Technology, Increased health cost

ABBREVIATIONS

AMGH …….. Arba Minch General Hospital

Appt. …….Appointment

BR ……. Business Rule

DB ……..Database

IDE…….Integrated Development Environment

Mgmt. ……Management

SRS…… Software/System Requirement System

UAT……User Acceptance Testing

UML……Unified Modeling language

UI…….User interface

UX…….User Experience

CHAPTER ONE
INTRODUCTION

The Doctor Appointment System is a comprehensive digital solution aimed at streamlining the appointment management process and enhancing overall healthcare delivery. With the increasing demand for efficient healthcare services and the growing reliance on digital technologies, there is a pressing need for a robust system that can facilitate seamless communication between patients and healthcare providers. The current system at AMGH relies heavily on manual processes, leading to inefficiencies, long waiting times, and suboptimal patient experiences. To address these challenges, the Doctor Appointment System seeks to leverage technology to automate appointment scheduling, improve patient-doctor communication, and optimize administrative tasks.

At its core, the Doctor Appointment System aims to provide a user-friendly platform that empowers patients to conveniently schedule appointments, receive timely reminders, and access essential healthcare information. By transitioning from manual to digital processes, the system seeks to reduce patient waiting times, minimize administrative burdens on healthcare staff, and enhance overall patient satisfaction. Moreover, the system's integration of features such as doctor ratings, patient feedback, and appointment tracking will enable AMGH to gather valuable insights into patient preferences, improve service quality, and foster better patient-provider relationships.

The primary objectives of the Doctor Appointment System include improving appointment management efficiency, enhancing patient access to healthcare services, and optimizing resource utilization at AMGH. The scope of the project encompasses the development of a user-friendly web-based platform accessible to both patients and healthcare providers. Key functionalities will include appointment scheduling, patient registration, doctor profiles, appointment tracking, feedback management, and administrative tools for health officers. The project will utilize modern web development technologies, databases, and secure payment gateways to ensure scalability, reliability, and data security. Through the implementation of the Doctor Appointment System, AMGH aims to modernize its healthcare delivery processes, improve patient outcomes, and elevate the overall quality of healthcare services provided to the community.

1.1 Background Information of the Organization

Arba Minch General Hospital (AMGH) is a public hospital located in Arba Minch town, Ethiopia. It is one of the general hospitals in SNNPR region of Gamo Zone. It was founded on December 7, 1961 EC. It has a bed capacity for over 200 patients. The hospital has been working in collaboration with Arba Minch University. It serves as a place of practice for students of the college of medicine and health sciences.

The medical interns, general practitioners and specialists employed by the university also work in the hospital and serve the community. The first medical doctor of the hospital was an Icelandic national called Dr Johannes Olafsson, who worked in Ethiopia from 1960 to 1980. Arba Minch hospital introduced the first sonography machine in 1977, which was also the first for the country. Arba Minch General Hospital is the only hospital serving a population of 200,747 people per year in Gamo and other nearby zones.

According to an Ethiopian health system model, a general hospital should serve 1-1.5 million population with an average of 234 staff. Even though it is named as a general hospital it is functioning as a referral hospital with the number of staff and several services it is providing.

1.1.1 Vision

Inspire seeing Arba Minch general hospital as the center of excellence in the country in quality health care service delivery, reduced critical shortage of high-level health professionals, and having a healthy, productive, and prosperous community.

1.1.2 Mission

To reduce morbidity, mortality, disability, and improve the health status of the people through the provision of quality curative, rehabilitative, promotive, and preventive health services. To train competent high-level health professionals and to generate important evidence by conducting problem-solving research.

1.2 Background of the project

A doctor appointment app is a mobile solution that permits patients to schedule appointments with doctors online. These applications supply patients with a list of available doctors. Consumers can also see doctors' specialties & their availability. They can then opt for a doctor & plan an appointment at a time that is convenient for them. A doctor appointment app's

primary purpose is to facilitate the scheduling of medical visits. The app's goal is to make it easy for patients to check the schedules of their preferred doctors, find an available appointment time, and book it digitally.

An all-inclusive app for scheduling medical appointments would provide in-depth doctor profiles that include education, areas of expertise, years in practice, and feedback from previous patients. Patients rely on this data to help them find a physician who best suits their requirements. The app aims to make it simple for patients to find and consider physician performance feedback. Complete and trustworthy doctor profiles bolster users' faith in the app's suggestions and the doctors.

1.3 Team composition

Our team consists of five graduate class students, each bringing their skills and expertise to the project. All the group members will participate to do the project documentation part as well as have full responsibility and unique role during implementation.

Table 1.1: Team composition

Project title	MediHub - Healthcare Platform for Arba Minch City			
Prepared by:	**Name**	**ID**	**Email/mobile phone**	**Responsibility**
	Hawi Kaba	NSR/1269/12	hawikaba123@gmail.com	All activity
	Rahel Getachew	NSR/1971/12	richgetu3@gmail.com	All activity
	Sisay Fikadu	NSR/2171/12	sisayfk@gmail.com	All activity
	Sorawak Abaya	NSR/2195/12	soraab7070@gmail.com	All activity
	Tewodros Mekuriya	NSR/2313/12	tewodrosd47@gmail.com	All activity
Date	14 Feb, 2024			
Advisor	Instructor: Mr. Melaku Meskele			

1.4 Statement of the problem

The current healthcare landscape demands an efficient and user-friendly Doctor Appointment System to bridge the gap between patients and healthcare providers. Despite the widespread use of digital tools, the absence of a comprehensive and integrated solution has led to several challenges in managing appointments, doctor information, and patient feedback.

Users don't want to waste their time waiting in queues at health stations. It can be solved by allowing users to schedule appointments and minimize waiting times. Some users forget to take their medications on time. The app offer a prescription management feature that sends medication reminders and notifications to users, ensuring timely medication intake and reducing the likelihood of missed doses.

Other identified problems include:

Inefficient Appointment Management:

- Existing systems lack a seamless process for patients to book appointments with doctors, leading to confusion and potential scheduling conflicts.
- Hospital health officers face challenges in managing and controlling appointment activities, resulting in suboptimal resource utilization.

Limited Doctor Information Accessibility:

- Patients often struggle to access comprehensive information about doctors, including their specializations, qualifications, and patient reviews, hindering informed decision-making.
- Hospital health officers lack a centralized platform to efficiently add, update, or remove doctors from the system, leading to information gaps.

Complicated Payment Processes:

- Patients encounter difficulties in making secure payments during the appointment booking process, affecting the overall user experience.
- Hospital health officer lack streamlined tools to track and manage payments effectively.

Absence of Real-Time Feedback Mechanism:

- Patients currently lack a convenient way to provide feedback and ratings for their healthcare experiences.
- Hospital health officer do not have a systematic approach to review patient feedback and utilize it for continuous improvement.

Limited Appointment History Visibility:

- Patients face challenges in accessing their complete appointment history, impacting their ability to track and manage their healthcare interactions.
- Hospital health officer lack a centralized system for efficient appointment history tracking and review.

1.5 Objective of the project

1.5.1 General Objective

The general objective of this project is to develop a doctor appointment platform that enhances the overall experience for both patients and doctors in Arba Minch General Hospital.

1.5.2 Specific objectives

- To develop a user-friendly mobile app enabling patients to easily book, reschedule, or cancel appointments.
- To provide a web app for hospital health officer with elements for efficient appointment management.
- To offer patients detailed doctor information, including specializations, qualifications, and patient reviews through the mobile app.
- To implement a secure payment gateway within the mobile app for patients.
- To integrate a real-time feedback mechanism in the mobile app for patients to provide reviews and ratings.
- To design both the patient mobile app and doctor & hospital health officer web app to be intuitive and user-friendly, catering to users of varying technological proficiency.

1.6 Feasibility Analysis

1.6.1 Operational feasibility

To determine the operational feasibility of the system we should take into consideration the knowledge level of the users, specifically patients. Initially this system is proposed to be designed for those who use mobile phones. The application is expected to be very flexible for its users. And this is also one of the factors that make it operationally feasible. With a team of five students and access to widely-used development tools, the project can be effectively managed and implemented within the available resources. Additionally, local languages can be utilized for implementation, further simplifying technical requirements.

1.6.2 Technical feasibility

The project is technically feasible given the expertise of the development team and the use of established development frameworks and languages. The team's familiarity with HTML, CSS, JS, Laravel PHP framework, and Flutter ensures that technical challenges can be addressed effectively.

1.6.3 Economic feasibility

Since the project primarily involves the utilization of open-source and widely-used development tools, the costs associated with software development are minimal. Recurrent costs are also expected to be low, given the simplified nature of the project and the availability of local languages for implementation.

The development materials are freely available; the only thing the user requires is that a mobile phone and an emulator can be used representing the mobile phone too. As all the resources are freely available, it gives an indication that the system is economically possible for development.

1.6.4 Behavioral/Political feasibility

Assuming acceptance by Arba Minch General Hospital, the project aligns with stakeholder interests by providing an innovative solution to enhance healthcare delivery. The project team members will build the system without violating the rules and regulations of the governments as well as the organization. The system being built is for the sake of the productivity of the organization and customers so that the project is legally feasible

1.6.5 Schedule feasibility

The project is schedule feasible, given the availability of resources like internet access, time spent on, active team members' participation, No and/or a little time overload of class schedules with our project developing schedule plan, and the simplified nature of the feasibility issues. With a team of five developers and simplified feasibility issues, the project can be completed within the proposed time frame. Additionally, the use of established development tools and frameworks accelerates the development process.

Table 1.2: Schedule feasibility

SN	Phases	1st quarter	2st quarter	3st quarter	4st quarter	5st quarter
		Dec15-25	Jan26-Feb10	Feb11-Mar30	Apr1-May20	May21-May30
1	Project proposal					
2	Requirement analysis					
3	System design					
4	Coding & Implementation					
5	Installation and Testing					

1.7 Scope and significance of the project

The Doctor Appointment System project encompasses the development of a comprehensive healthcare platform designed to significantly improve the overall experience for patients, hospital health officer and doctors. This initiative involves the creation of a user-friendly mobile app tailored for patients, facilitating seamless appointment management, including booking, rescheduling, and cancellations. Concurrently, the project includes the development of a web app specifically designed for hospital health officer, equipped with tools to efficiently manage and control appointment activities, thereby minimizing scheduling conflicts and optimizing resource utilization.

The system provides a simple way for patients to arrange visits with the healthcare providers of their choice, in our case, AMGH. And secondly, these systems store all appointments and data entries under the same roof. Patients can book appointments from anywhere at any time, without the need for phone calls or in-person visits. This can be particularly beneficial for patients with busy schedules, mobility issues, or those who live far away from the medical facility.

With an app, one can set reminders and notifications to keep no-shows at bay. And even if patients miss their visits, he/she can reschedule back-to-back appointments straight away to avoid losing money due to an unused time slot. By developing a medical appointment booking system, Hospitals can also prevent overlapping appointments. They are the bane of all healthcare businesses. Double bookings often result from human error, and an automated solution can help prevent them from being made in the first place.

Another critical aspect of the project is to provide patients with a detailed understanding of their healthcare providers. The mobile app will offer comprehensive doctor profiles, featuring specializations, qualifications, and patient reviews. Simultaneously, the web app for hospital health officer will serve as a centralized hub for adding, updating, or removing doctors, ensuring that the information presented to patients remains accurate and up-to-date.

To ensure a seamless experience, the project focuses on integrating a secure payment gateway within the mobile app for patients. Central to the system is the integration of a real-time feedback mechanism within the mobile app. This feature empowers patients to provide immediate reviews and ratings, contributing to enhanced patient engagement. Additionally, the

hospital health officer can systematically review and utilize these insights for continuous improvements in service quality.

Effective communication channels are established within the mobile app for patients to address queries or concerns and receive timely updates. The web app empowers hospital health officer to efficiently notify relevant parties about appointment updates and changes, fostering effective communication.

1.8 Target beneficiaries of the system

The target beneficiaries of this Doctor Appointment System are diverse and include various stakeholders within the healthcare ecosystem. The system is designed to cater to the needs of:

Patients:

- Patients benefit from the system by gaining access to a user-friendly mobile app that facilitates seamless appointment management, access to comprehensive doctor information, secure payment processes, and a real-time feedback mechanism. The system empowers patients to have greater control over their healthcare experience.

Doctors and Hospital:

- Doctors and Arbaminch General Hospital benefit from efficient appointment scheduling and management tools provided by the system. The centralized doctor information hub ensures accurate representation, and real-time feedback allows them to continuously improve the quality of their services based on patient insights.

Hospital health officer:

- Hospital health officer play a crucial role in managing the system, ensuring appointment activities are streamlined, doctor information is up-to-date, and payments are tracked transparently. The web app provides tools for efficient management and control of various aspects of the system.

1.9 Methodology for the project

1.9.1 Data Source

To do this project, the project team uses data collection methodology to gather the necessary information that is needed to develop the project. The methodology used for information gathering is primary & secondary sources. The methods we employed in the data

collection are observation, interviews, document of analysis and internet that are references or indirectly related with the doctor appointment application system. The data sources include:

- Arba Minch General Hospital
- Doctors in Arba Minch General Hospital
- Internet and websites
- Existing documents about doctor appointment system
- Advisor

1.9.2 Fact Finding Techniques

By employing the following fact-finding techniques, we gained diverse perspectives and ensured a well-rounded understanding of our project's scope and requirements.

Observation:

We have observed the existing system in Arba Minch general hospital how they assist patients with chronic diseases who visit the hospital usually. We followed and observed doctors, hospital staff, and patients in their daily routines to observe workflows, identify pain points, and understand existing practices firsthand.

Interview:

We did an interview to get information about the existing system for developing our project. We have conducted interviews with health care professionals.

Document analysis:

We used this method to get additional information from different online resources using the internet. We visited similar healthcare platforms like Practo mobile application to gain insights into their functionalities and user experiences.

1.10 Systems Analysis and Design

Systems analysis and design (SAD) is a methodical approach to developing, improving, and maintaining information systems. SAD helps ensure our system is well-defined, meets user needs, and functions effectively. We are going to model the requirements that we gather using an object-oriented approach because object-oriented technology is of broad means used to handle highly structured applications. Object-oriented technology provides numerous advantages in applications by making them easy to use and maintain.

The model that we will use in our system development life cycle is the Iterative model. Because we aimed to do the project with it. Whenever we want to add or expand the scope we can expand as well if we cannot implement all the proposed scope by different reasons like shortage of time, we can minimize the scope. During this phase we used a draw.io online tool to design different diagrams.

1.11 Development Tools

Table 1.3: Development tools

Category	Tools/Technologies
Integrated Development Environments (IDEs) and Editors	Android Studio
	Visual Studio Code
	Web Tools
Version Control and Collaboration Tools	Git
Mobile App Development Frameworks and SDKs	Flutter SDK
Modeling and Design Tools	draw.io
UI/UX Design Tools	Figma/canva Adobe photoshop
Backend Development Frameworks	Laravel
Database	MySQL
API Development and Testing Platforms	Postman
Using Firebase Authentication	Firebase Authentication
Testing	Flutter Testing Framework
Documentation and Collaboration Tools	Microsoft Word & Google Docs
	Trello

1.12 Testing procedures

We will perform different testing for checking the functionality of our proposed system.

Unit testing:

- First we will test each unit at each system. So, if a problem is encountered it will immediately maintain at which the problem occurs.

Integration testing:

- After we test each unit of the proposed system, we will perform an integration test to check whether the system meets all the functional requirements. When several

components are complete, they will test to ensure that they integrate well with each other, the operating system, and other components.

System testing:

- After all of the above testing is checked we will test our system with other people and we will conduct some comments on how they get our system. To validate the system as a whole and ensure it meets the specified requirements.

User Acceptance Testing (UAT)

- To obtain feedback from end-users and ensure the system meets their expectations and needs. End-users, including patients, clinics, and city health officials, will be invited to participate in UAT sessions.

1.13 Implementation (Parallel/Partial/Direct)

The present way of handling appointments for patients with critical health problems who usually visit the hospital is still working although patients cannot schedule appointments by themselves. We choose to develop the newly proposed system parallel to the existing system. We are going to change the system after this proposed system is finished until the customers use the manual system.

1.14 Limitation of the project

- Reliable internet access: Online appointment scheduling requires a reliable internet connection. If you can't guarantee a constant connection, you might miss new, canceled, or updated bookings.
- Lack of medical knowledge: Patients may not be able to self-classify their appointment type or urgency.
- Usability: most clients of Arba Minch General Hospital came from different rural parts of the city so they cannot use this digital system because there is familiarity issue with digital systems and they have no smartphone so they prefer the manual based system.

1.15 Risks (*What* if Analysis?), Assumptions and Constraints

1.15.1 Risks – The what if analysis

Potential technical challenges such as system crashes during development which can occur in any project. System crashes can occur due to various reasons such as software bugs, hardware failures, compatibility issues, or inadequate testing procedures.

Risks associated with low user adoption and engagement, including resistance to change, lack of digital literacy, or competing priorities among target users.
Time Constraints and Student Commitments: The risk of incompleteness or inability to develop the system as intended due to time constraints and competing priorities, such as class schedules, exams, and other academic obligations faced by team member students.

1.15.2 Assumptions

Assuming that in the coming years, a majority of the target population will have access to basic digital devices (e.g., smartphones) and internet connectivity to use this system, and they will become familiar with these technologies and online systems as the world delves further into technological advancements.

Assuming that team member students will be able to allocate sufficient time and effort to the project despite academic commitments, such as class schedules, exams, and university exit exam preparation.

1.15.3 Constraints

Time and Academic Commitments: The project is constrained by limited availability of student team members due to academic commitments, including class schedules, exams, university exit exam preparation, and other academic obligations.

Resource Limitations: Student team members may face limitations in terms of time, expertise, and resources available for project development and implementation, potentially impacting the scope and quality of the platform to be developed.

CHAPTER TWO

DESCRIPTION OF THE EXISTING SYSTEM

2.1 Introduction of Existing System

At Arba Minch General Hospital, the current appointment system relies predominantly on manual processes and direct communication between patients and doctors. The system primarily caters to patients with critical cases, where immediate attention and treatment are paramount. Here's an overview of how the existing system operates:

Manual Appointment Scheduling: Patients with critical cases typically visit the hospital in person seeking urgent medical attention. Upon arrival, they are assessed by hospital staff to determine the severity of their condition. If deemed critical, patients are directed to the appropriate department or specialist for immediate treatment.

Doctor-Patient Interaction: In the existing system, doctors play a central role in the appointment process. They evaluate the urgency of each patient's condition and prioritize appointments accordingly. Doctors communicate appointment details directly to patients verbally or through handwritten notes, relying on personal discretion and memory to manage their schedules. Hence patients, even the doctors may forget the date.

Inefficient Communication: With appointment scheduling primarily reliant on verbal communication, there is a risk of miscommunication or missed appointments. Patients may forget their appointment times, leading to disruptions in the treatment process and potential complications in managing critical cases.

2.2 Players in the existing system

In the existing manual appointment system at Arba Minch General Hospital, several key players are involved in the appointment-related processes. These players include:

Patients:
Patients are the primary stakeholders in the appointment system. They visit the hospital seeking medical care and require appointments with doctors to receive treatment for their health conditions. Patients provide information about their medical history, symptoms, and concerns during appointments.

Doctors and Specialists:

Doctors and specialists play a crucial role in assessing patients' medical needs and scheduling appointments accordingly. They evaluate the urgency of patients' conditions, diagnose illnesses, and prescribe treatments. Doctors communicate appointment details to patients and may also coordinate with other healthcare professionals involved in patient care.

Hospital Receptionists:

Receptionists are responsible for managing appointment bookings, handling patient inquiries, and maintaining appointment schedules. They assist patients in registering for appointments, verifying insurance information, and providing guidance on hospital procedures.

Medical Nurses:

Medical nurses support doctors and specialists in delivering patient care. They may assist with patient examinations, perform medical tests, administer treatments, and provide patient education. Medical assistants and nurses may also help coordinate appointments and follow-up care for patients.

2.3 Major functions/activities in the existing system

In the existing manual appointment system at Arba Minch General Hospital, various functions and activities occur, encompassing inputs, processes, and outputs.

Inputs:

- Patient Information: Input includes personal details, medical history, and symptoms provided by the patient upon arrival at the hospital.
- Doctor Availability: Information about the availability of doctors and specialists to schedule appointments.
- Hospital Resources: Availability of medical facilities, examination rooms, and equipment required for patient consultations.
- Urgency Assessment: Evaluation of the urgency of each patient's condition to prioritize appointments accordingly.

Processes:

- Appointment Scheduling: Receptionists manually schedule appointments based on doctor availability and the urgency of patients' conditions.

- Patient Triage: Medical staff assess the severity of patients' conditions to determine the order of appointments and prioritize critical cases.
- Doctor-Patient Interaction: Doctors and specialists interact with patients to diagnose illnesses, prescribe treatments, and provide medical advice during scheduled appointments.
- Record-Keeping: Hospital staff maintain paper-based records of appointment schedules, patient information, and medical histories.
- Communication: Verbal communication between hospital staff, doctors, and patients to convey appointment details, instructions, and follow-up care.

Outputs:

- Appointment Schedule: Output includes a schedule of appointments listing the date, time, and doctor assigned to each patient.
- Treatment Plans: Doctors provide patients with treatment plans, prescriptions, and recommendations based on their diagnoses.
- Patient Records: Paper-based records containing patient information, medical histories, and treatment notes are generated and updated after each appointment.
- Patient Satisfaction: The outcome includes patient satisfaction with the appointment process, quality of care received, and overall experience at the hospital.
- Follow-up Instructions: Patients receive instructions for follow-up appointments, medication schedules, and additional tests or procedures as needed.

2.4 **Business Rules**

Business Rules refer to a set of guidelines, policies, and procedures that govern various aspects of appointment-related processes. Business rules of the existing system is listed as follow:

BR1: A doctor can be scheduled for many appointments, but may not have any scheduled at all.

BR2: Patient records are typically updated with each visit to reflect the encounter and contribute to their medical history.

BR3: One appointment is scheduled with exactly one patient.

BR4: Each appointment is scheduled with exactly one doctor.

BR: A patient can schedule one or more appointments.

BR6: Emergency cases do not require an appointment. However, for appointment management purposes, an emergency is entered in the appointment book as “unscheduled.”

BR7: Patients with critical or time-sensitive conditions receive priority scheduling.\

BR8: If kept, an appointment yields a visit with the doctor specified in the appointment. The visit yields a diagnosis and, when appropriate, treatment.

2.5 Report generated in the existing system

In an existing system at AMGH, there are different reports generated for different purposes. Those reports include work status report, payment report, customer report, manager report etc.

- Appointment Schedule Report: This report provides an overview of all scheduled appointments within a specific time period. It includes details such as appointment date, time, patient name, doctor/specialist assigned, and appointment status (e.g., confirmed, pending, canceled).
- Manual Record-Keeping: The hospital likely relies on manual record-keeping processes. Patient information, medical histories, appointment schedules, and treatment plans are documented using paper-based systems, which can be time-consuming and prone to errors.
- Daily Appointment Schedule Report: This report lists the appointments scheduled for each day, including the patient's name, appointment time, and doctor/specialist assigned. It provides a snapshot of the day's appointment schedule for hospital staff to reference.
- Missed appointments by patient: A list of patients with frequent missed appointments, potentially highlighting compliance issues.

2.6 Forms and other documents of the existing systems

There are various forms used for different purposes.

- Patient Registration Form: Captures patient demographics, contact information, medical history, insurance details, and emergency contact information.
- Appointment Book: Physical book or register recording scheduled appointments, including date, time, doctor, patient name, and reason for visit.

- Referral Slips: Doctors use these to refer patients to other specialists, providing details of the case and recommendations.
- Paper Charts: Physical folders containing patient medical records, including consultations, diagnoses, medications, lab results, and other clinical notes.
- Patient Follow-up Form: After each appointment, a follow-up form is filled out to document the visit's outcome, prescribed treatments, and recommendations for follow-up care.
- Prescription Forms: Doctors write medication prescriptions on standardized sheet/paper forms that include the patient's name, medication name, dosage instructions, and any special considerations.
- Manual Bills: Handwritten or printed invoices detailing charges for services rendered.
- Payment Receipts: Physical receipts issued to patients upon payment.

2.7 Bottlenecks of the existing systems

In a manual paper-based appointment system in AMGH, where only doctors can appoint patients, several bottlenecks and challenges can arise, impacting the efficiency and effectiveness of healthcare delivery. Here are some potential bottlenecks in such a system:

2.7.1 Performance requirements

- Slow response times: Appointments can take a long time to schedule due to manual processes, including searching paper records, and data entry.
- Limited accessibility: Patients can only book appointments during clinic hours and require direct communication, leading to long wait times and potential delays in receiving care.
- Lack of appointment reminders: Patients might miss appointments due to relying on memory or manual reminders.
- Difficult appointment changes: Rescheduling or canceling appointments requires contacting the doctor or staff, creating another bottleneck.
- High risk of missed appointments: Patients and doctors relying on memory or paper reminders can easily forget appointments, leading to wasted time, lost revenue, and potential health consequences.

2.7.2 Input (Inaccurate/redundant/flexible) and Output (Inaccurate)

Input of appointment information relies on manual data entry, which is prone to errors, redundancy, and lacks flexibility. Staff may make mistakes when recording appointment details, leading to inaccuracies and inefficiencies.

- Limited flexibility: The system lacks the flexibility to accommodate diverse patient needs or preferences for scheduling (e.g., specific time slots, online booking).
- Lack of transparency: Patients lack access to their own appointment history and may not be fully informed about their care plan.

2.7.3 Security and Controls

- Every appointment record of the AMGH and customer information in the existing system is manually stored on paper. This makes controlling and securing that information extremely difficult. In addition, those paper records can be easily accessed by unauthorized persons, which results in damage, loss, and misplacement of those documents. The existing manual system is not secure; there is no authentication mechanism for documenting information.

2.7.4 Efficiency

- It is difficult to view through papers to check the patient information as needed, and the appointment service schedules. It is also difficult to search for specific patient doctor information from the list on the paper. Because the list on the paper becomes huge and some of the sections of the paper records may be lost or wrongly organized. Due to lack of appointment reminders, patients might miss appointments since relying on memory or manual reminders. This leads to wasted time, lost revenue, and potential health consequences.

2.7.5 Service

- The existing doctor appointment system taking place through doctors provides services that is slow and facilitated by poor schedule management, which reduces patient satisfaction and the hospital's productivity

2.8 Practices to be preserved

In the AMGH manual doctor appointment system, there are some practices worth preserving despite the system's limitations. These practices can contribute to the efficient operation of the appointment system and the delivery of quality patient care. Here are some practices to consider preserving:

- Direct Doctor-Patient Communication: With doctors directly involved in appointment scheduling, there is an opportunity for direct communication between doctors and patients.

- Direct in-person scheduling: Patients can visit the hospital in person and doctors schedule appointments directly with them. This might be inconvenient, but it allows those with limited access to book appointments.
- In a manual system, doctors have the flexibility to prioritize urgent cases and accommodate patients with critical health needs more quickly. This can help ensure timely access to care for patients in urgent situations, potentially saving lives and preventing complications.
- Continuity of Care: With doctors directly involved in appointment scheduling, there is an opportunity to promote continuity of care for patients. Doctors can ensure that patients receive follow-up appointments as needed, monitor their progress over time, and make adjustments to treatment plans as necessary to achieve optimal health outcomes.

2.9 Proposed solution for the new system that address problems of the existing system

The proposed solution for the new appointment system addresses key challenges identified in the existing manual system in AMGH. By implementing an automated appointment reminder system, both patients and doctors will receive timely reminders via SMS, reducing the risk of missed appointments due to reliance on memory. Additionally, the development of an online appointment management platform will empower patients with self-service options, allowing them to view available slots, schedule appointments, and request for making changes at their convenience.

Furthermore, providing patients with access to their appointment history and care plans through a secure online portal will improve transparency and communication between patients and healthcare providers. Streamlining appointment change processes and adopting a patient-centric scheduling approach will enhance flexibility, accommodate patient preferences, and optimize patient satisfaction. Together, these solutions will transform the appointment system, improving efficiency, accessibility, patient experience, and overall quality of care.

2.10 Requirements of the Proposed System

2.10.1 Functional requirements

Functional requirement explains and describes what things are performed by the system. It shows the functionalities we are going to get from this system.

Performance Requirements:

- The system should be capable of handling a large volume of appointment requests and users without degradation in performance.
- The system should be available 24/7 to accommodate appointment scheduling and changes outside of business hours.

Process Requirements:

- User registration and authentication: Allow users (patients, doctors, health officer) to register for accounts and securely authenticate themselves using login credentials.
- The system should streamline appointment scheduling, changes, and cancellations processes, minimizing administrative burden and bottlenecks.
- The system should allow patients to request scheduling, or rescheduling appointments online through a user-friendly interface.
- The system should automatically send reminders to patients and doctors via SMS before their scheduled appointments.

Input Related Requirements:

- The system should accept input of appointment details, including patient information, doctor preferences, appointment date, time, and location.
- Doctors should be able to set their availability status for upcoming days through the system.
- The system should require user authentication for patients and doctors to access appointment management features securely.

Output Related Requirements:

- The system should generate appointment schedules for doctors and patients, displaying upcoming appointments and availability.
- The system should generate appointment requests and confirmation messages for doctors and patients respectively, providing details about the appointment.
- The system should display the (scheduled) availability status of doctors for the upcoming days for patients, receptionists, and doctors themselves.
- The system should send appointment reminders to patients and doctors via SMS before the scheduled appointment time.

Storage Related Requirements:

- Database Storage: The system should store appointment-related data securely in a centralized database, including patient information, appointment schedules, and appointment history.

2.10.2 Non-functional requirements

Performance

- The system shall perform its operations within a minimum amount of time. That means it takes short response time for a given piece of work. The designed system will use low utilization of system resources in terms. The system works in an efficient manner.

User Interface

- The UI should be intuitive and easy to navigate, requiring minimal training for both patients and staff to use effectively. Its design should maintain consistency across different screens and functionalities, providing a cohesive user experience.

Security and Access permissions

- Users will have their own password and username through which they could gain access to the system. Users should be required to authenticate themselves before accessing sensitive functionalities (such as appointment scheduling or availability setting). Access permissions should be granular, allowing administrators to define roles and permissions for different user groups (e.g., patients, doctors, receptionists). User without the right access permission or role cannot able to access the feature of other users.

Backup and Recovery

- The system should perform regular backups of all data, including appointment records, availability settings, and user information. Backup data should be stored in geographically diverse locations to ensure redundancy and minimize the risk of data loss.

CHAPTER THREE

SYSTEM ANALYSIS

3.1 Introduction

In this chapter, we dig into the System Analysis phase, which includes a detailed examination of client needs, system functionalities, and limitations to guarantee the effective plan and usage of the proposed arrangement. Various artifacts generated during System Analysis, including Use Case Diagrams, Use Case Documentation, Sequence Diagrams, Activity Diagrams, Analysis Level Class Diagrams, User Interface Prototypes, and Supplementary Specifications are presented. Each component defines the behavior, structure, and functionality of the system, guiding subsequent phases.

3.2 System Requirement Specifications (SRS)

This SRS serves as a blueprint for software development, providing a clear understanding of what the system is expected to do and how it should perform. It defines not only what the system should do but also how it should behave in various scenarios. This level of detail is essential for guiding the development process and mitigating misunderstandings or discrepancies later on.

3.2.1 Use case diagram

A use case diagram is a graphical depiction of a user's possible interactions with a system. A use case diagram shows various use cases and different types of users the system has and will often be accompanied by other types of diagrams as well. The use cases are represented by either circles or ellipses. It is essential to analyze the whole system before starting with drawing a use case diagram, and then the system's functionalities are found. And once every single functionality is identified, they are then transformed into the use cases to be used in the use case diagram. The following are use case components:

- Actors: It is a person, or external system that interacts with a system with the system. An actor can be a person, an organization, or an outside system that interacts with your application or system. They must be external objects that produce or consume data.
- System: A specific sequence of actions and interactions between actors and the system. A system may also be referred to as a scenario.

- Goals: The end result of most use cases. A successful diagram should describe the activities and variants used to reach the goal.

In our project we have three actors that interact with our project: hospital health officer as admin, doctor, and patient.

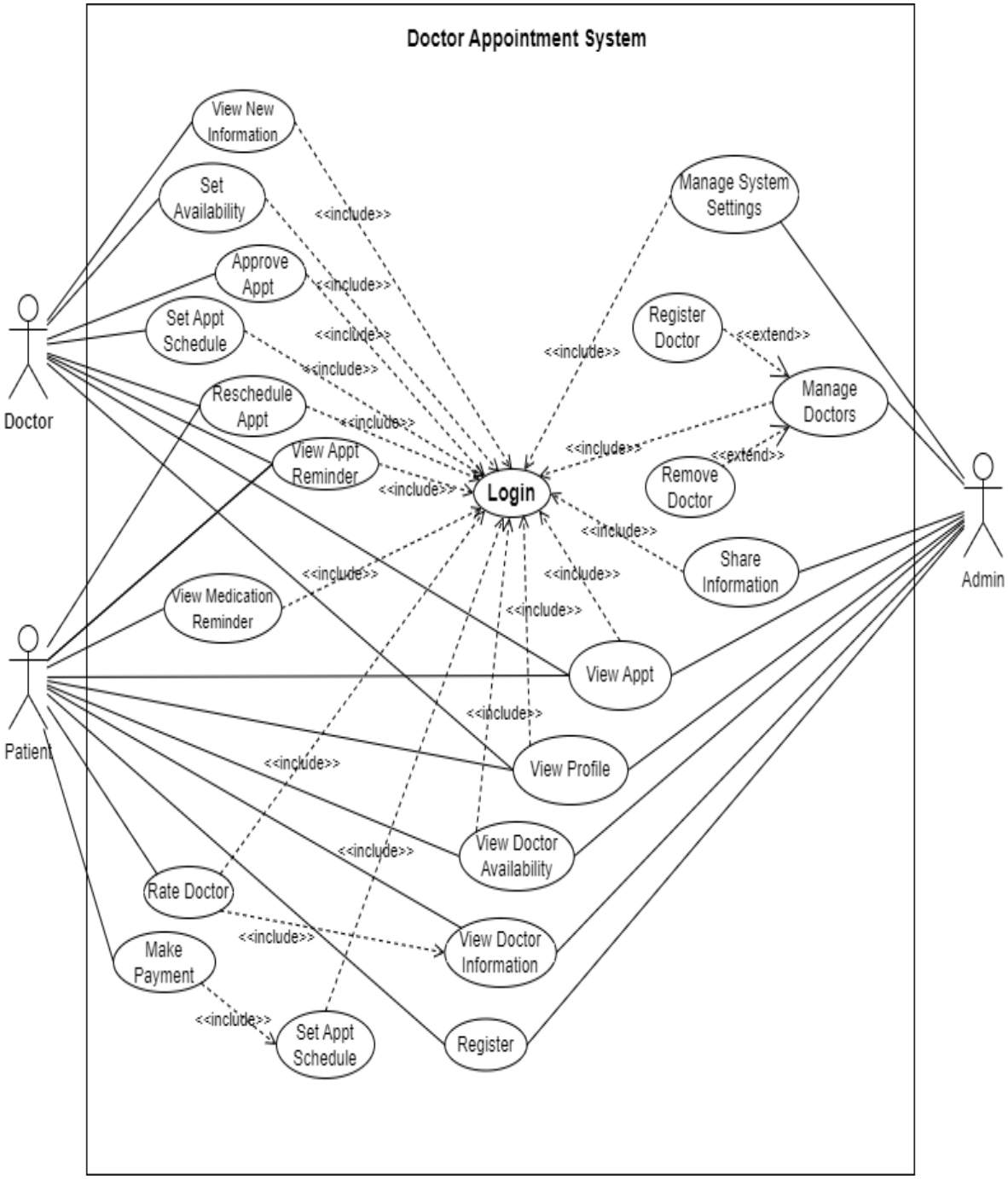


Figure 3.1: Use case diagram

3.2.2 Use case documentation

Here are some use case descriptions of the system:

Table 3.1: Use case description for patient registration

Use case name	Patient Registration
Use case ID	UC-1
Actor	Patient
Description	Patients can register in the system by providing necessary details for identification and authentication.
Pre-condition	The mobile app must be installed.
Basic course of action	1. Patient clicks register button on the landing page. 2. System displays registration form to patient. 3. Patient inputs his/her details as required. 4. System verifies entered patient details. 5. System saves patient information and sends messages.
Post condition	System confirms successful registration.

Table 3.2: Use case description for requesting appointment

Use case name	Request Appointment
Use case ID	UC-2
Actor	Patient
Description	Patients can schedule appointments with their preferred doctors through the system. The booking process involves selecting a doctor, choosing an available time slot, and making a reservation.
Pre-condition	Patients must be logged into the system.
Basic course of action	1. Patient selects/clicks on the desired doctor from which system displays on the patient app home page. 2. System displays a page containing calendar. 3. Patient chooses an available time slot. 4. Patient confirms the appointment request. 5. System records the appointment request.
Post condition	Appointment request is submitted for processing.

Table 3.3: Use case description for rescheduling/postponing appointment

Use case name	Reschedule/Postpone Appointment
Use case ID	UC-3
Actor	Patient
Description	Requests to change the date or time of an existing appointment.
Pre-condition	Patient must have an existing appointment.

Basic course of action	1. Patient clicks appointments' tab after opening the app. 2. System displays the existing list of patient appointments. 3. Patient selects the appointment to be rescheduled. 4. System displays calendar with the appointed date. 5. Patient chooses a new date and/or time for the appointment. 6. System updates the appointment details.
Post condition	Appointment is successfully rescheduled.

Table 3.4: Use case description for Making Payment

Use case name	Make Payment
Use case ID	UC-4
Actor	Patient
Description	Patient initiates the payment process to settle appointment fees online.
Pre-condition	Patient must be logged in and schedule appointment.
Basic course of action	1. Patient fill new appt. form to schedule. 2. System pops out payment method display form to be filled. 3. Patient chooses the payment method (like telebirr) and enters payment details. 4. System validates the payment information. 5. Payment is processed securely through the selected payment gateway. 6. System confirms successful payment and updates the appointment status as confirmed.
Alternative course of action	6.1 System displays an error message indicating the reason for payment failure (e.g., insufficient balance, invalid card details)
Post condition	Payment is successfully processed, and the appointment is confirmed.

Table 3.5: Use case description for Providing Feedback and Rating

Use case name	Provide Feedback
Use case ID	UC-5
Actor	Patient
Description	Patient submits feedback regarding their experience with a specific doctor
Pre-condition	Patient must have completed an appointment with the doctor.
Basic course of action	1. Patient opens doctor appt. mobile app. 2. System displays doctors. 3. Patient selects the doctor for whom feedback will be provided. 4. System displays feedback form.

	5. Patient enters the feedback details, including ratings and comments. 6. Patient submits the feedback. 7. System records the feedback and updates the doctor's profile with the new information.
Post condition	Feedback is successfully submitted and recorded in the system.

Table 3.6: Use case description for Viewing Appointments

Use case name	View Appointments
Use case ID	UC-6
Actor	Patients, Doctors
Description	Doctors and patients can view their appointments. Doctors have the ability to approve or reject appointment requests.
Pre-condition	Patient, doctor or admin must be logged into the system.
Basic course of action	1. Actor clicks on the appt. mgmt. section/bottom-screen tab. 2. System retrieves appointment related menu lists. 3. The actor selects the option to view scheduled appointments. 4. System retrieves and displays the list of appointments. 5. Patients, doctors or admins can scroll through the list and click on view details button on each individual appointment. 6. System displays the clicked appt. with date, time, doctor, and appointment status (pending, accepted, or declined).
Post condition	List of scheduled appointments is successfully displayed.

Table 3.7: Use case description for Approving Appointments

Use case name	Approving Appointments
Use case ID	UC-7
Actor	Doctor
Description	Doctors can approve or reject appointment requests submitted by patients.
Pre-condition	Doctors must be logged into the system.
Basic course of action	1. Doctor clicks on the appt. mgmt. section/bottom screen tab. 2. System retrieves & displays appt. related menu lists for doctor. 3. Doctor selects the option for pending appointment requests. 4. System displays & retrieves list of pending appt. 5. Doctor reviews the details of each appointment request, including patient information & requested appointment time. 6. Doctor approves the appointment request for scheduling. 7. System updates the appointment status to "confirmed" and notifies the patient and doctor about the approved appointment.

Post condition	Appointment request is approved, and the appointment is confirmed.

Table 3.8: Use case description for Registering Doctor

Use case name	Register Doctor
Use case ID	UC-8
Actor	Health Officer (HO)
Description	HOs can register new doctors in the system. This includes entering details such as the doctor's name, specialization, qualifications, and contact information.
Pre-condition	The Health Officer must be logged into the system.
Basic course of action	1. The HO clicks on the doctor. mgmt. bottom-screen tab on his/her home page and clicks the register new doctor button. 2. System displays a form for registering doctor with required fields. 3. He/she enters the doctor's details, including name, contact information, specialty & qualifications, and clicks submit button. 4. System validates the information provided and creates a new doctor profile in the system. 5. The doctor is added to the list of registered doctors.
Post condition	Doctor is successfully registered in the system.

Table 3.9 : Use case description for Scheduling Appointment

Use case name	Schedule Appointment
Use case ID	UC-9
Actor	Patient, Health Officer, Doctor
Description	HOs, as administrative staff, have the capability to add schedules for doctors. This involves specifying the days and time slots when a particular doctor is available for appointments.
Pre-condition	The person must be logged in.
Basic course of action	1. Actor clicks on the appt. mgmt. bottom screen tab. 2. System displays appt. related menu lists. 3. Actor selects the option for appointment scheduling. 4. System displays a page with calendar. 5. Actor selects the desired date & time for the appointment. 6. Actor (doctor) verifies & submits the appointment. 7. System confirms the appointment scheduling and sends notifications to both the patient and the assigned doctor.
Alternate course of action	4.1 If actor patient, the actor selects the desired doctor. 6.1 If the actor is HO, the actor assigns the appointment to an available and patient-preferred doctor & doctor availability.
Post condition	Appointment is successfully scheduled in the system, and

	patients & doctors are notified.

Table 3.10: Use case description for Removing Doctors

Use case name	Remove Doctors
Use case ID	UC-10
Actor	Health Officer
Description	Registers new doctors or removes existing ones from the system.
Pre-condition	The Health Officer must be logged into the system.
Basic course of action	1. The HO clicks on the doctor mgmt. bottom-screen tab on his/her home page and selects the remove doctor button. 2. System displays a list of doctors each with remove button. 3. The HO clicks the remove button. 4. The system displays a smaller form asking reason to remove. 5. System prompts for confirmation to remove the doctor.
Post-condition	Doctor is added or removed from the system.

Table 3.11: Use case description for viewing profile

Use case name	View Profile
Use case ID	UC-11
Actor	Doctor, Patient, Health Officer
Description	Accesses and views personal profile information.
Pre-condition	They must be logged into the system.
Basic course of action	1. The actor clicks on his profile photo on the top-left side of the app. 2. System retrieves and displays his/her profile details.
Alternate course of action	1.1 The actor clicks the ‘my profile’ bottom menu.
Post condition	The actor’s profile information is displayed.

Table 3.12: Use case description for setting availability

Use case name	Set Availability
Use case ID	UC-12
Actor	Doctor
Description	Specifies working hours and availability for appointments.
Pre-condition	Doctors must be logged into the system.
Basic course of action	1. Doctors clicks on the appt. mgmt. bottom-screen tab. 2. System displays options under appt. mgmt. section. 3. Doctor accesses the ‘set availability’ setting section. 4. System displays a page with calendar to the doctor.

	5. Doctor specifies working hours, days off, and any exceptional circumstances. 6. System saves the updated availability settings.
Post-condition	Doctor's availability settings are updated.

Table 3.13: Use case description for Postponing/Rescheduling Appointment

Use case name	Postpone/Reschedule Appointment
Use case ID	UC-13
Actor	Doctor
Description	Change the date or time of an existing appointment.
Pre-condition	Doctor must have an existing appointment.
Basic course of action	1. Doctors clicks on the appt. mgmt. bottom-screen tab on his/her home page. 2. Doctor clicks on the scheduled appointments option 3. System displays list of scheduled appointments of the doctor. 4. Doctor selects the appointment to be rescheduled. 5. System displays a page with calendar. 6. Doctor chooses a new date and/or time for the appointment. 7. System asks for new time and date confirmation. 8. Doctor confirm or decline the process. 9. System updates the appointment details.
Post-condition	Appointment is successfully rescheduled.

Table 3.14: Use case description for View Patient Feedback/Ratings

Use case name	View Patient Feedback/Ratings
Use case ID	UC-14
Actor	Doctor, HO, Patient
Description	The actor accesses patient feedback and ratings (If the actor is doctor, helps for personal review and improvement; else if the actor is HO, for some decision making when required; else if the actor is patient to help him/her whom to appoint in the future).
Pre-condition	Doctors must be logged into the system.
Basic course of action for HO	1. The HO accesses the system interface using own credentials. 2. HO navigates to the profile of the specific doctor. 3. System displays the profile of the selected doctor. The HO views the integrated "Doctor ratings" section within the doctor's profile.
Basic course of action for doctors	1. Doctor logs in to the system using credentials. 2. Navigate to their own profile page. 3. View the integrated "Doctor Ratings" or "Feedback" section within their profile.

	4. Review own ratings and feedback.
Basic course of action for patient	1. Patient logs in to the system using credentials. 2. Navigate to the profile page of the doctor of interest. 3. View the integrated "Doctor Ratings" or "Feedback" section within the doctor's profile 4. Review ratings and feedback the doctor has.
Alternate course of action	If the desired doctor's profile cannot be accessed or the ratings and feedback section is unavailable, the system displays an error message or notification.
Post-condition	Doctor can view patient feedback and ratings.

The remaining use case descriptions can be similarly structured for other functionalities. These tables provide a clear overview of each use case's description, actors involved, preconditions, post-conditions, and main flow of events.

3.2.3 Sequence diagram

A Sequence diagram is an interaction diagram that shows how processes operate with one another and in what order. It 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.

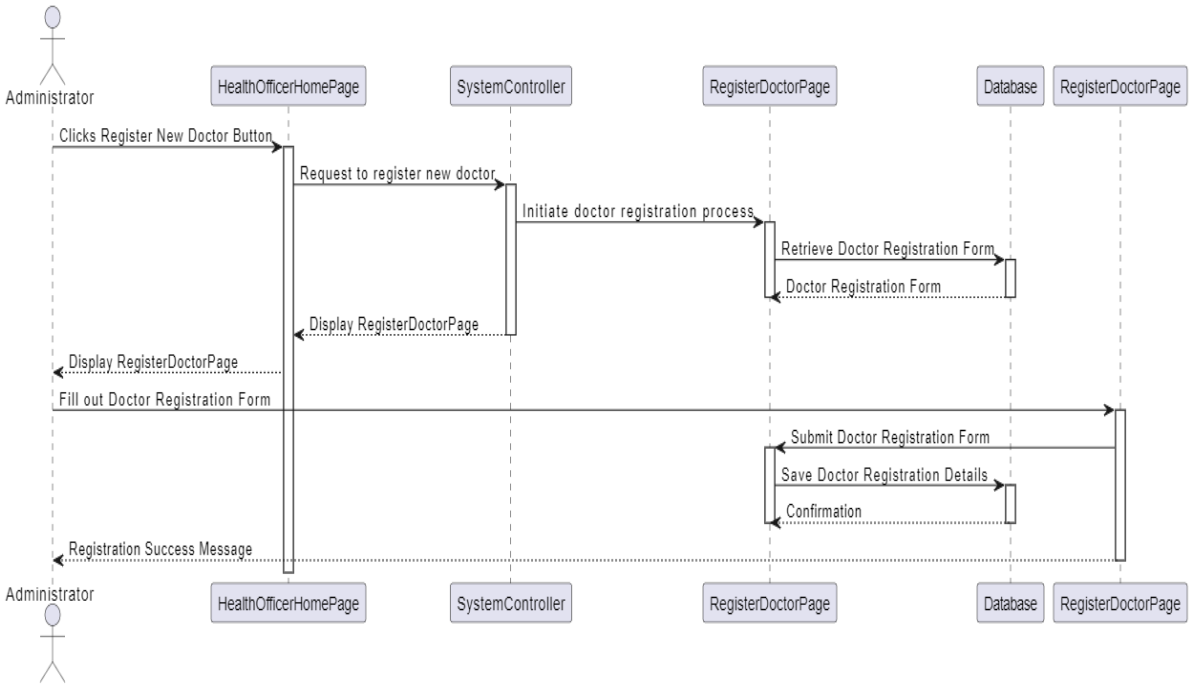


Figure 3.2: Sequence diagram for doctor registration

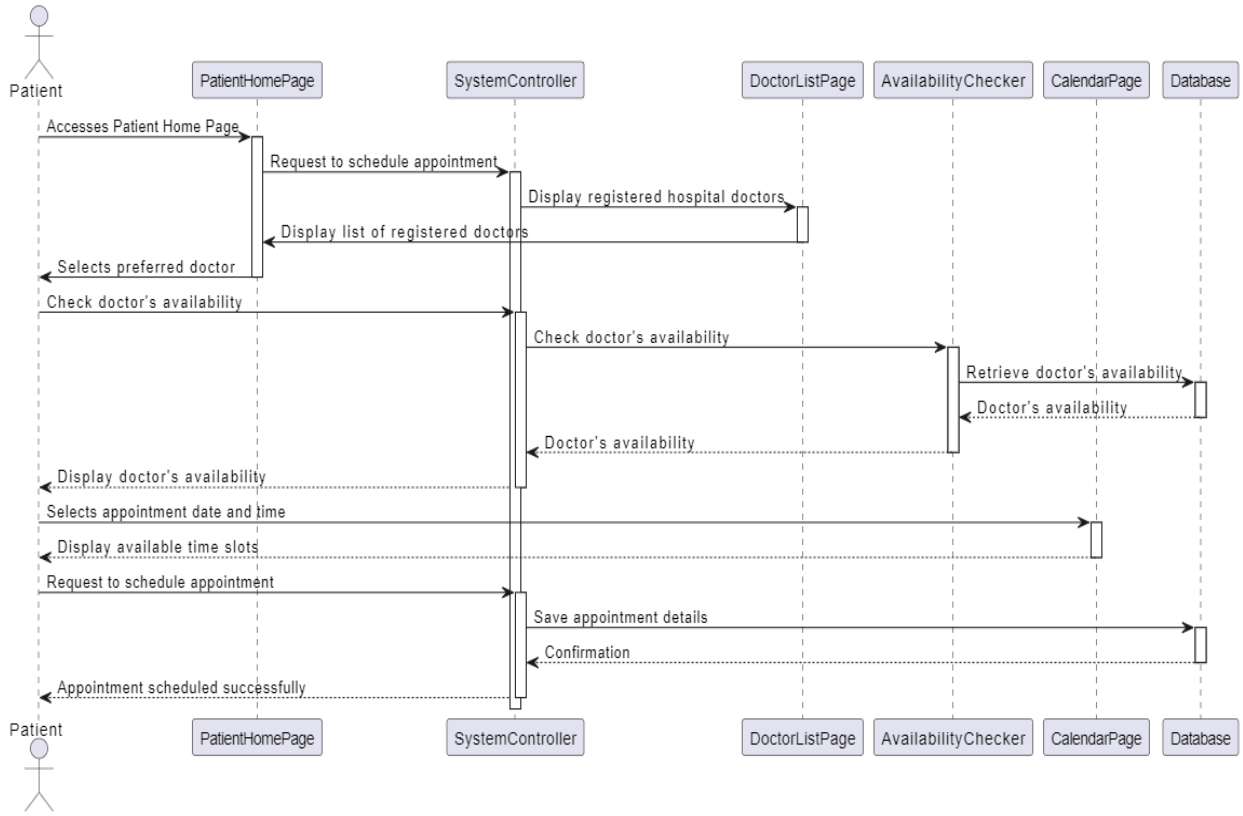


Figure 3.3: Sequence diagram for appointment scheduling by patient

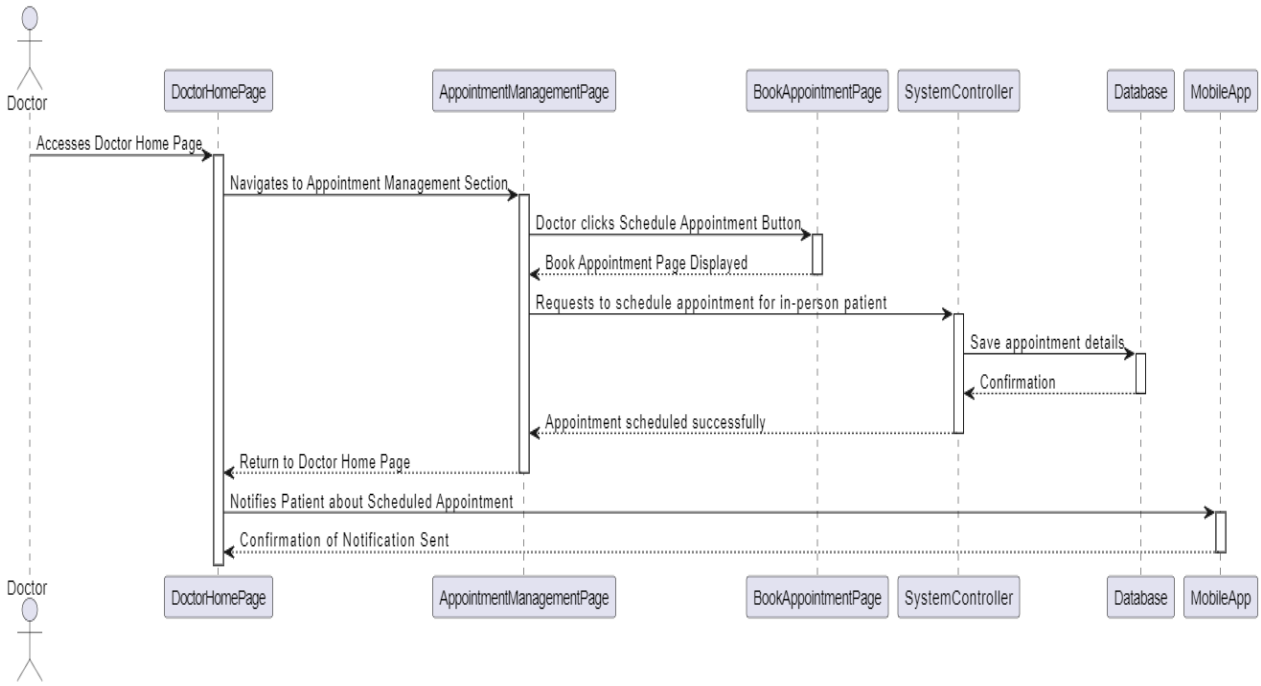


Figure 3.4: Sequence diagram for appointment scheduling by doctor

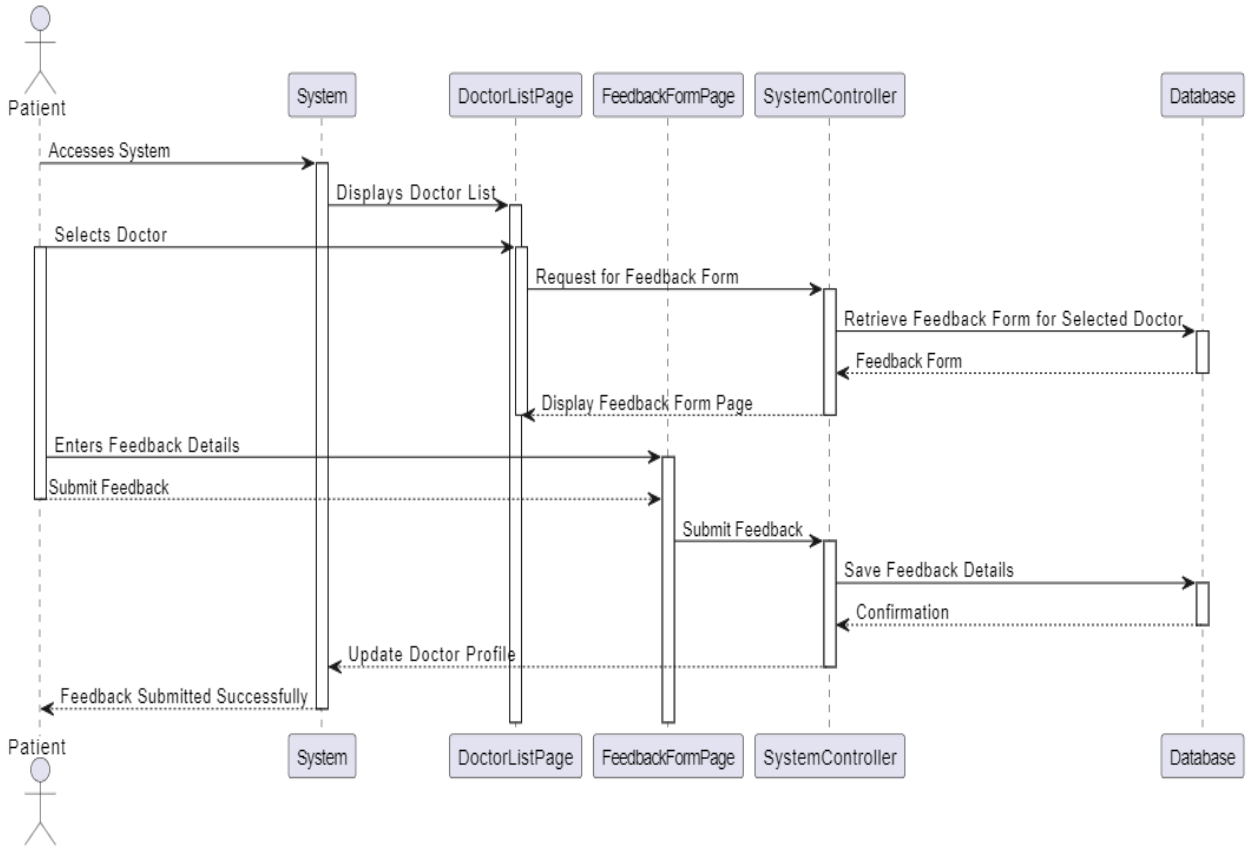


Figure 3.5: Sequence diagram for feedback submission

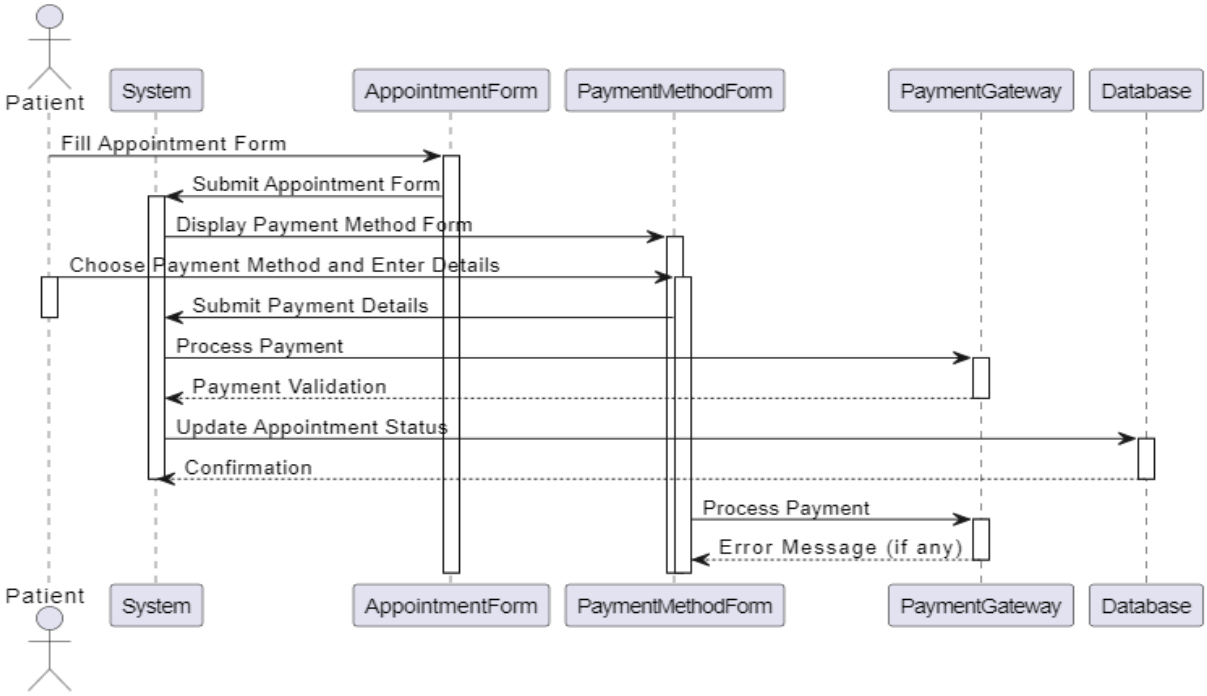


Figure 3.6: Sequence diagram for payment processing

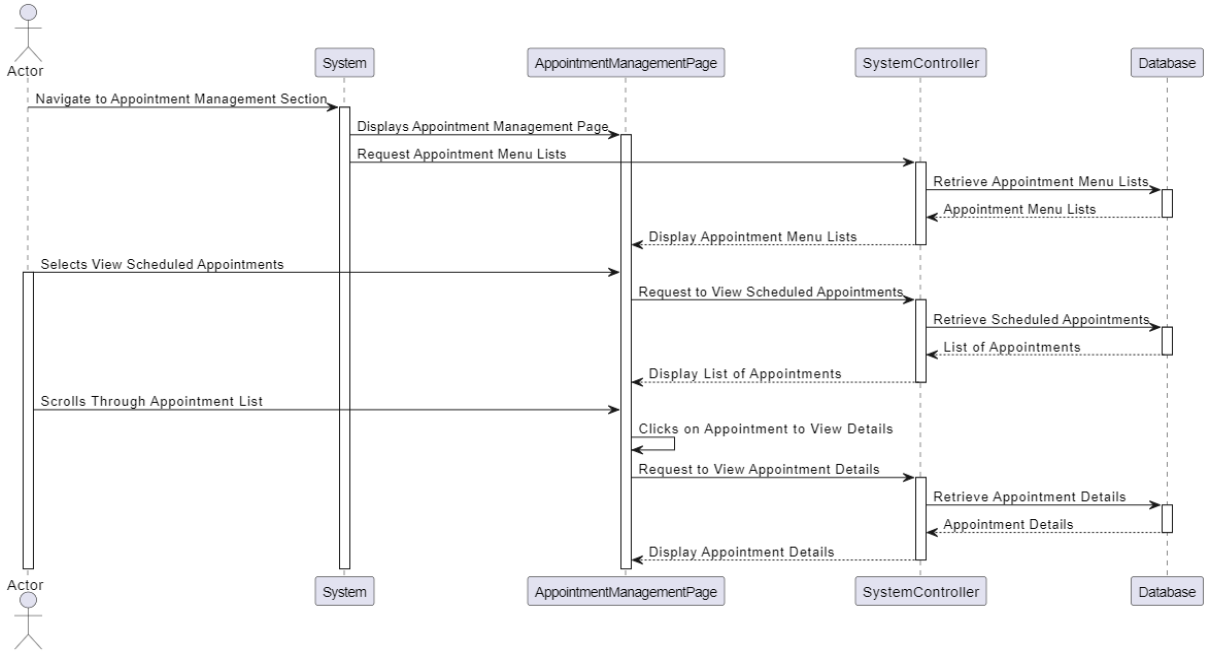


Figure 3.7: Sequence diagram for viewing appointments exist

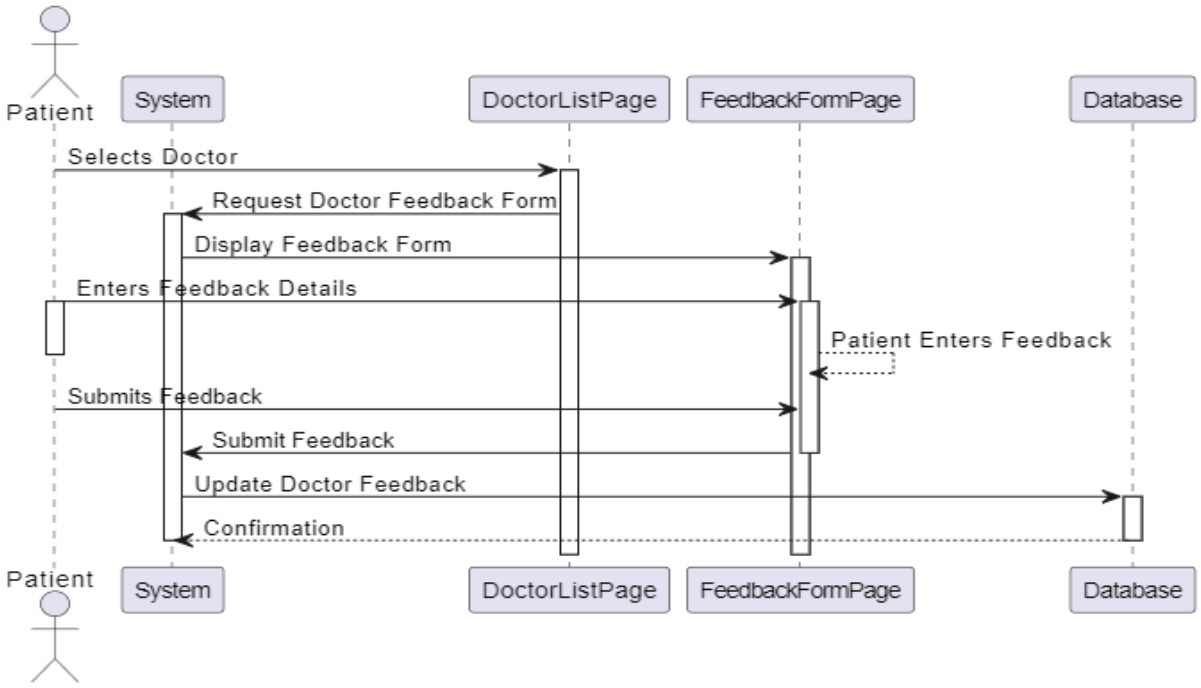


Figure 3.8: Sequence diagram for doctor rating

3.2.4 Activity Diagram

An activity diagram is a diagram that illustrates the flow of activities and actions within a system. They are useful for representing the flow of activities or actions within a system, particularly in scenarios where there are multiple possible paths or decision points.

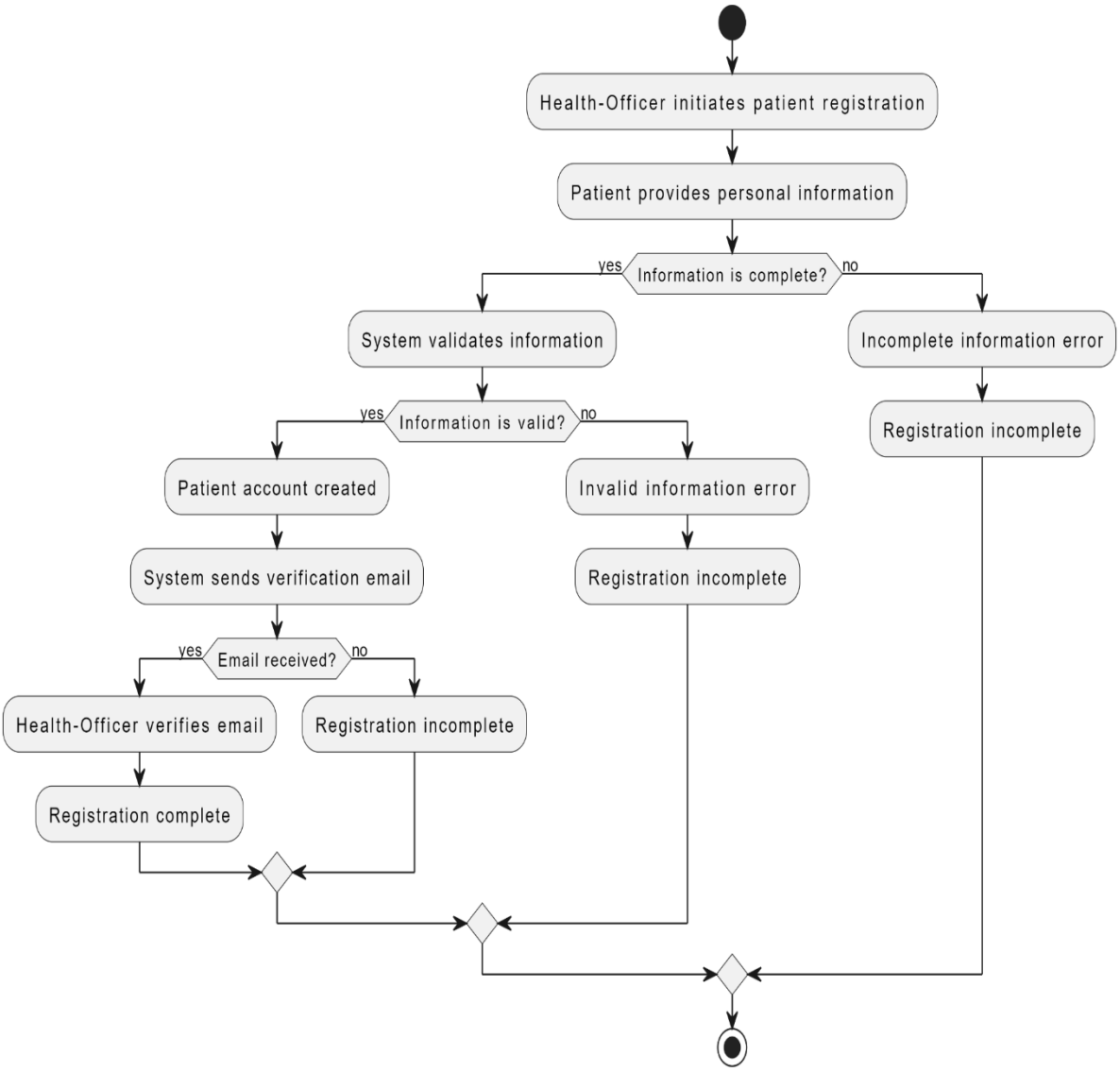


Figure 3.9: Activity diagram for patient registration

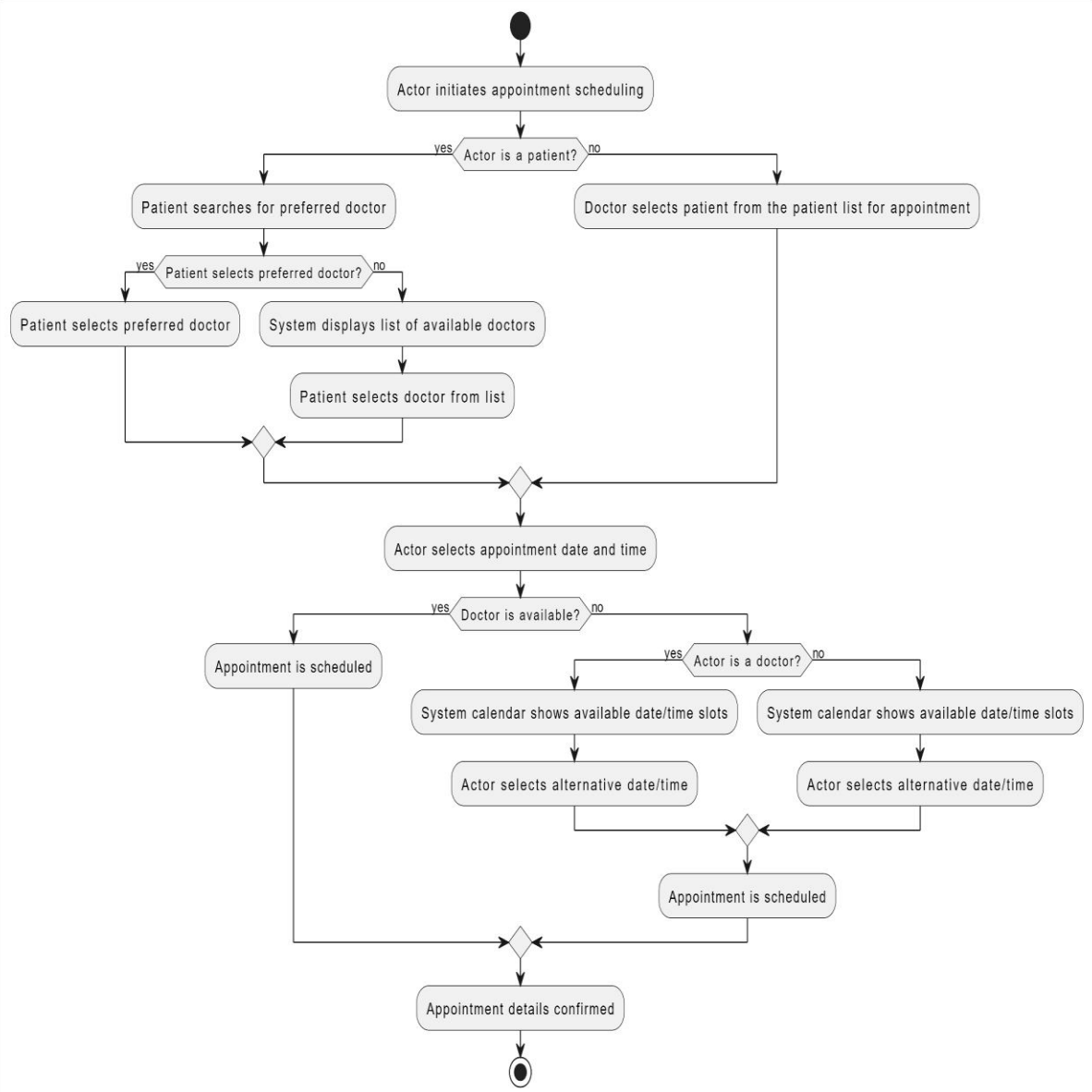


Figure 3.10: Activity diagram for appointment scheduling

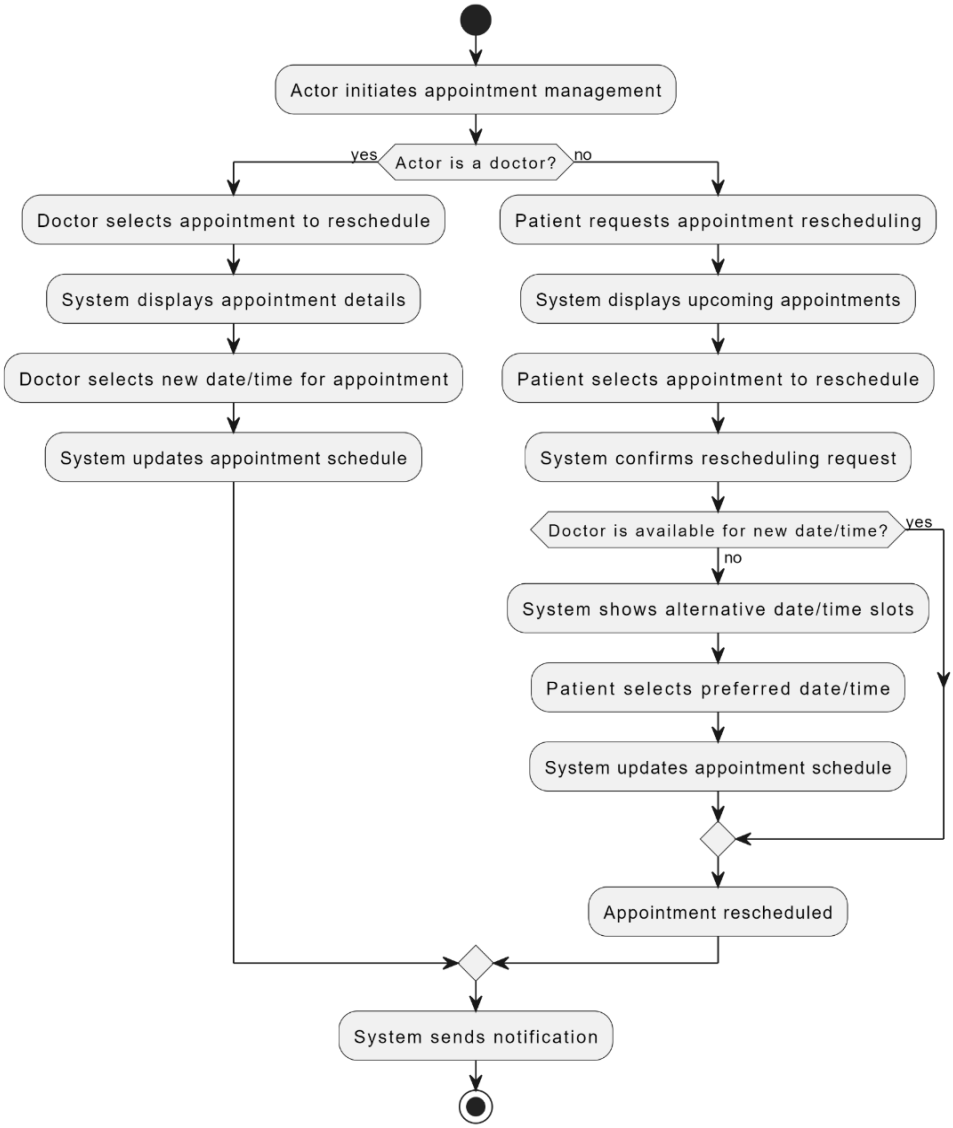


Figure 3.11: Activity diagram for appointment management

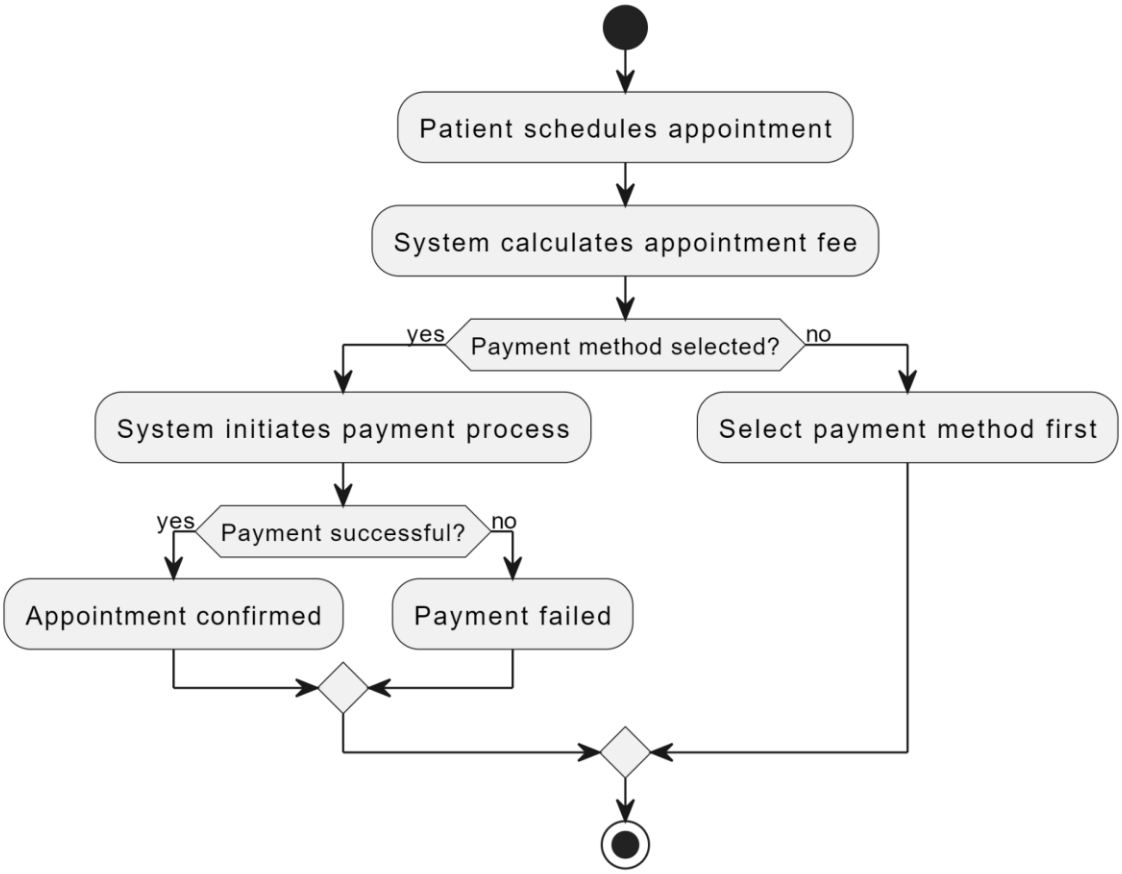


Figure 3.12: Activity diagram for payment processing

3.2.5 Analysis level class diagram (conceptual modeling)

A class diagram is a type of Unified Modeling Language (UML) diagram that provides a visual representation of the structure and relationships within a system or software application. It is a static diagram that focuses on the classes, their attributes, operations (methods), and the relationships between classes.

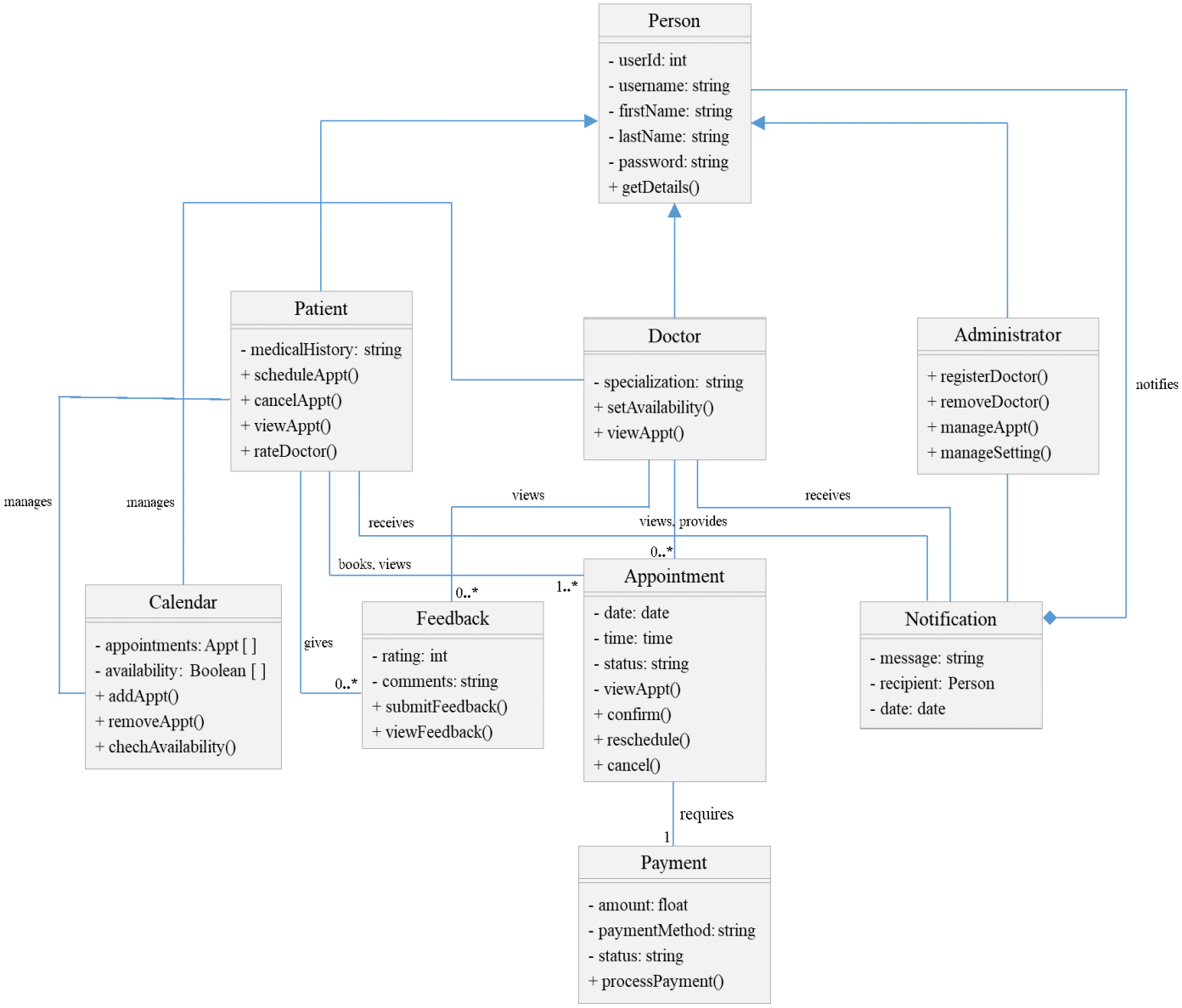
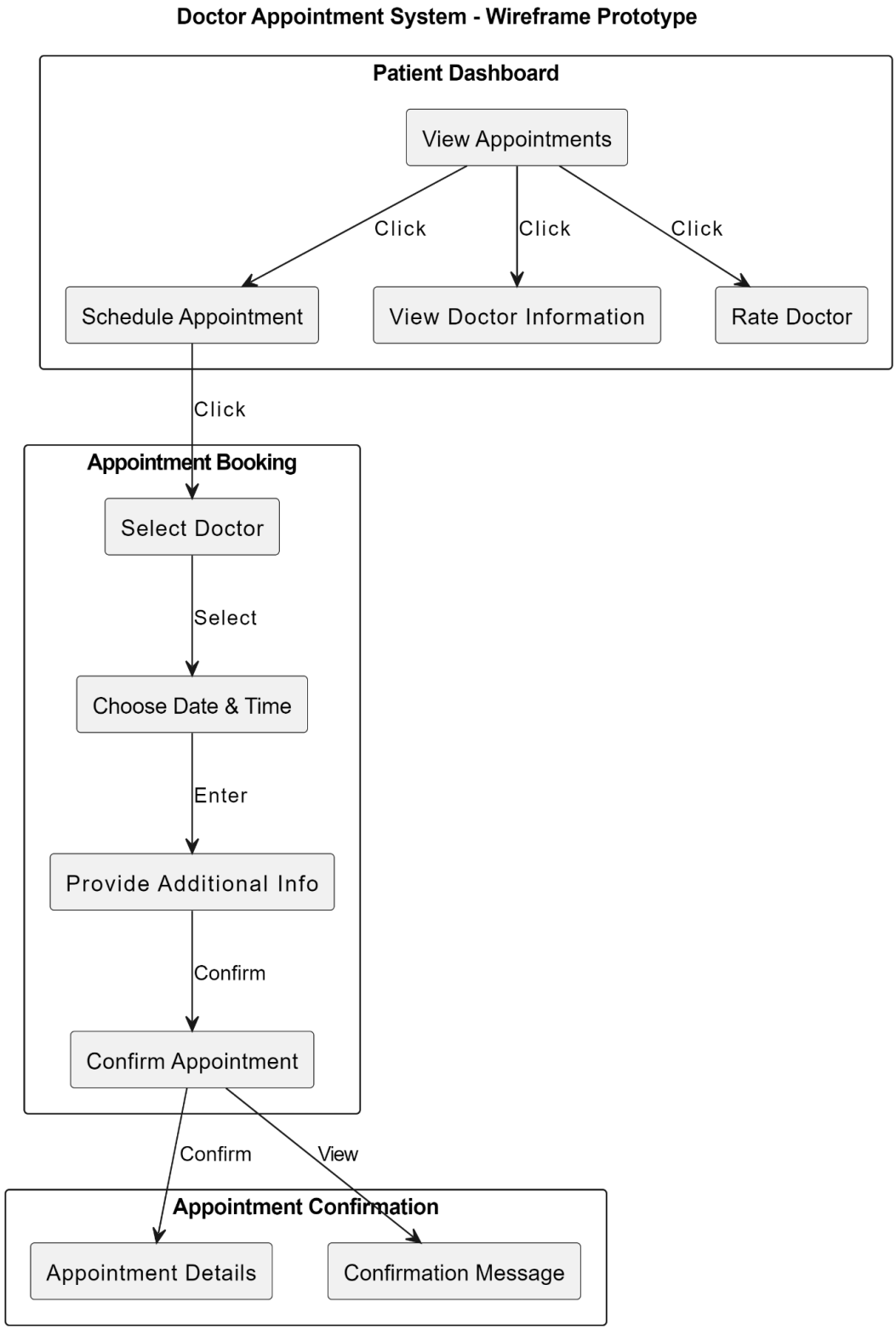
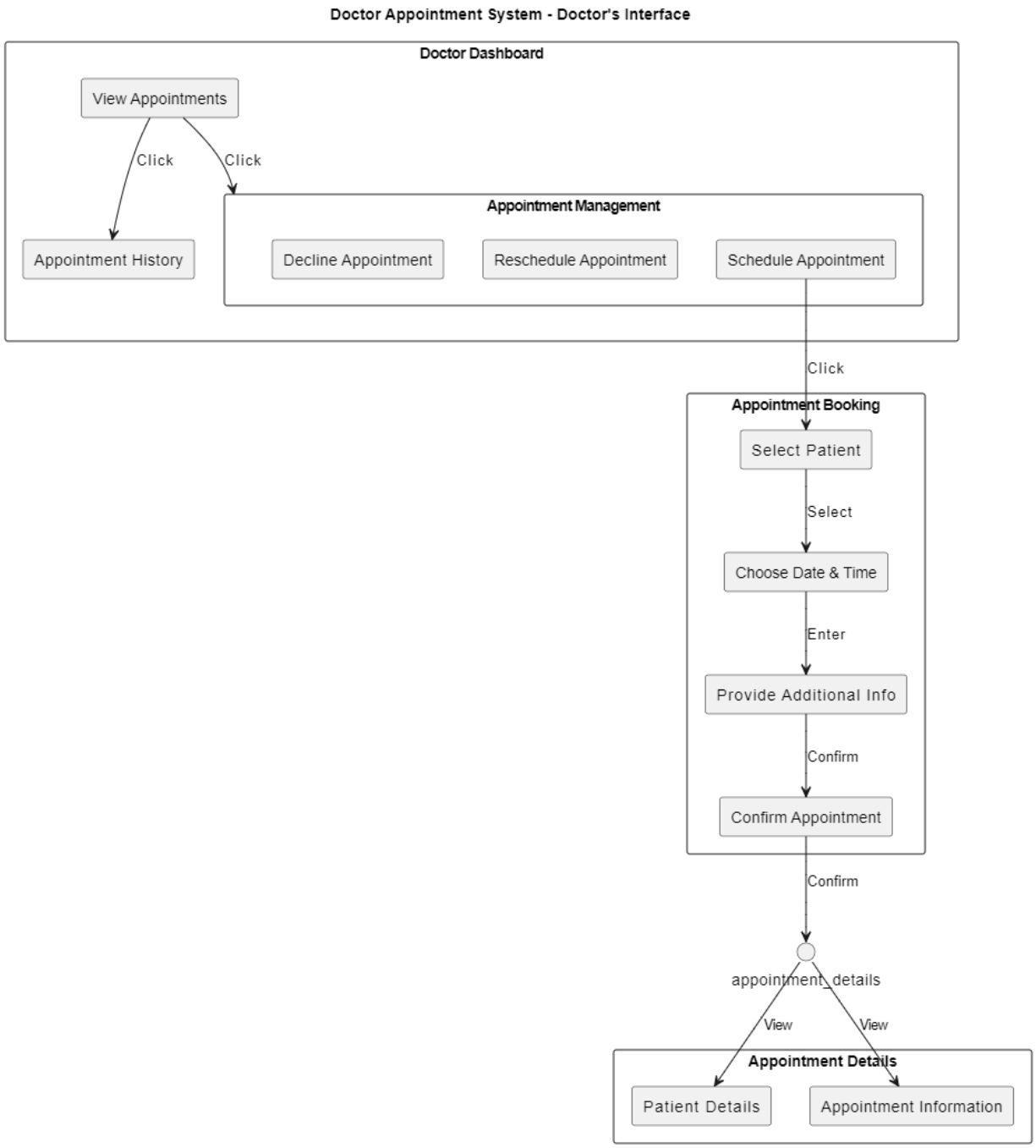


Figure 3.13: Analysis level class diagram

3.2.6 User Interface Prototyping




Doctor Appointment System - Doctor's Interface
Doctor Dashboard
View Appointments
Click
Click
Appointment History
Appointment Management
Decline Appointment
Reschedule Appointment
Schedule Appointment
Click
Appointment Booking
Select Patient
Select
Choose Date & Time
Enter
Provide Additional Info
Confirm
Confirm Appointment
Confirm
appointment_details
View
View
Appointment Details
Patient Details
Appointment Information

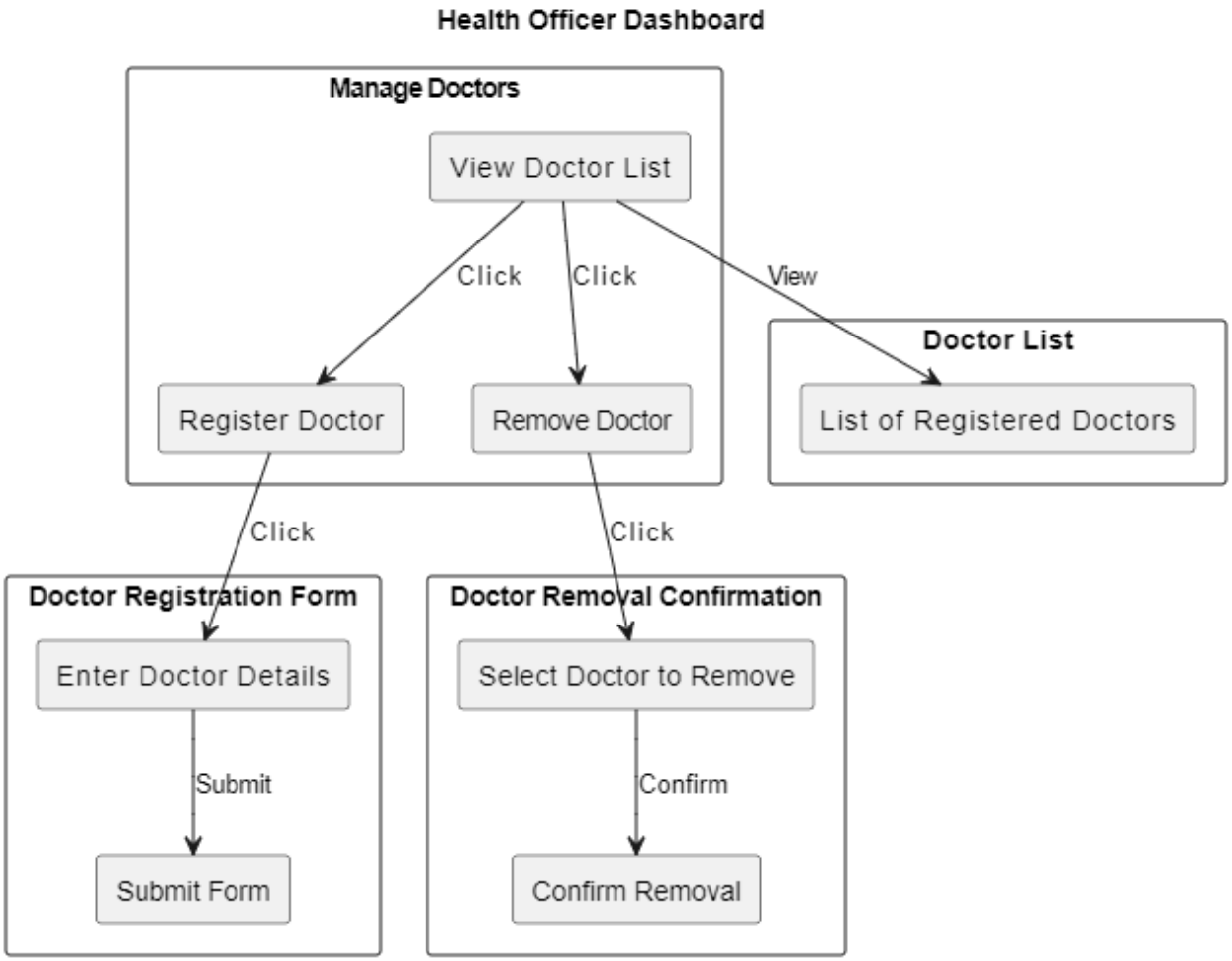


Figure 3.14: User Interface Prototyping

3.2.7 Supplementary specifications

Compliance with local, national, and international healthcare laws and standards is mandatory.

Comprehensive training materials, user guides, and documentation should be provided to help users understand how to use the system effectively.

The system should respect patient privacy and confidentiality, ensuring that only authorized individuals have access to patient data.

CHAPTER FOUR

SYSTEM DESIGN

4.1 Introduction

This chapter provides a comprehensive overview of the system design for the Doctor Appointment System. The focus is on defining the architecture, modeling different aspects of the system, and detailing the user interface design. This structured approach ensures the system is robust, scalable, and meets the specified requirements. The chapter covers various architectural layers, including the user interface, controller/process, business/domain, persistence, and system layers. It also delves into different modeling techniques such as class modeling, state chart modeling, collaboration modeling, component modeling, deployment modeling, and persistence modeling. Finally, it outlines the user interface design principles, wireframes, and mockups.

4.2 Class-type architecture

Class-type architecture is a common architectural strategy that is used to layer the architecture of a system into several layers or strata. In this section, we describe the layered architecture employed in the Doctor Appointment System. This architecture ensures separation of concerns, modularity, and maintainability by dividing the system into distinct layers. The layers include the User Interface Layer, Controller/Process Layer, Business/Domain Layer, Persistence Layer, and System Layer.

- **User Interface layer**: The user interface layer includes all the visual elements and interactions that users have with the system. This layer handles all interactions with the user. It consists of the screens and forms that the users interact with. The UI Layer includes both web and mobile interfaces to cater to different user preferences and accessibility needs. Each type of user (patient, doctor, health officer) has specific dashboards and pages tailored to their needs.
- **Domain layer**: This layer encapsulates the core business logic and rules of the Doctor Appointment System focusing on the data aspects of the business objects, plus behaviors specific to individual objects. It includes services and models that represent the system's entities and their interactions. This layer is crucial for implementing the functionalities that

meet the business requirements. Key elements include: Services, Business Logic, and Entities/Models.

- **Process layer**: The Process Layer handles the flow of data between the UI Layer and the Business/Domain Layer. It processes user inputs, performs validation, and coordinates responses. This layer acts as an intermediary that interprets user actions and invokes the necessary operations in the business layer. Key components in this layer include: Controllers, Middleware, and Routing.
- **Persistence layer**: Persistence layers encapsulate the capability to store, receive and delete Objects/data permanently without revealing details of the system.
- **System layer**: System classes provide operating-system-specific functionality for the applications, isolating the software from the operating system (OS) by wrapping OS-specific features, increasing the portability of the application. It includes configuration management, security, and system integration components.

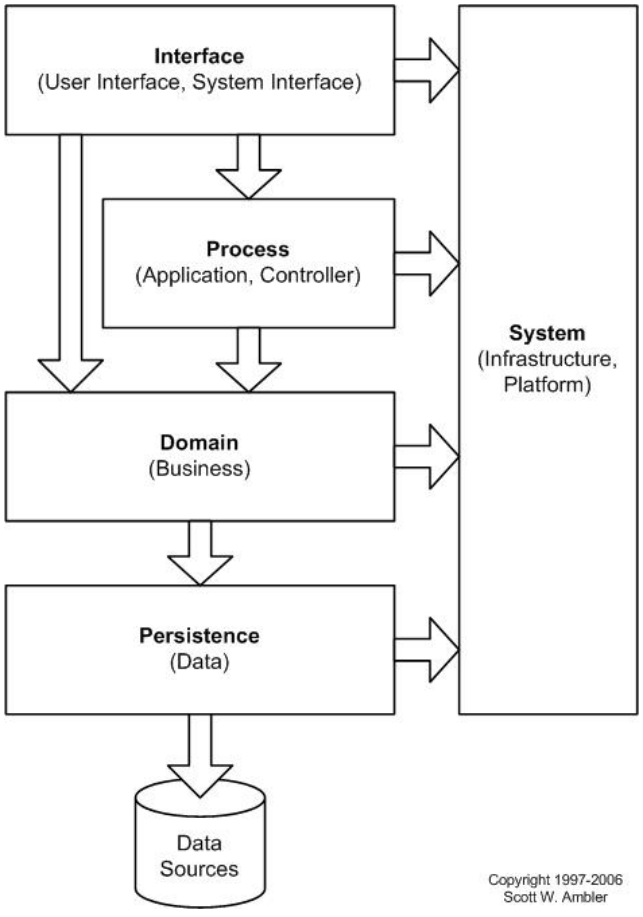


Figure 4.1: System layer

4.3 Class modeling

The class diagram is a type of static structure diagram that describes the structure of a system by showing the system's class, their attributes, operations (or methods), and the relationships among objects. It involves identifying and defining the key classes (or objects) that represent the major components of the system. In the context of our Doctor Appointment System, class modeling helps us understand the structure and relationships between different parts of the system, ensuring a clear and organized design.

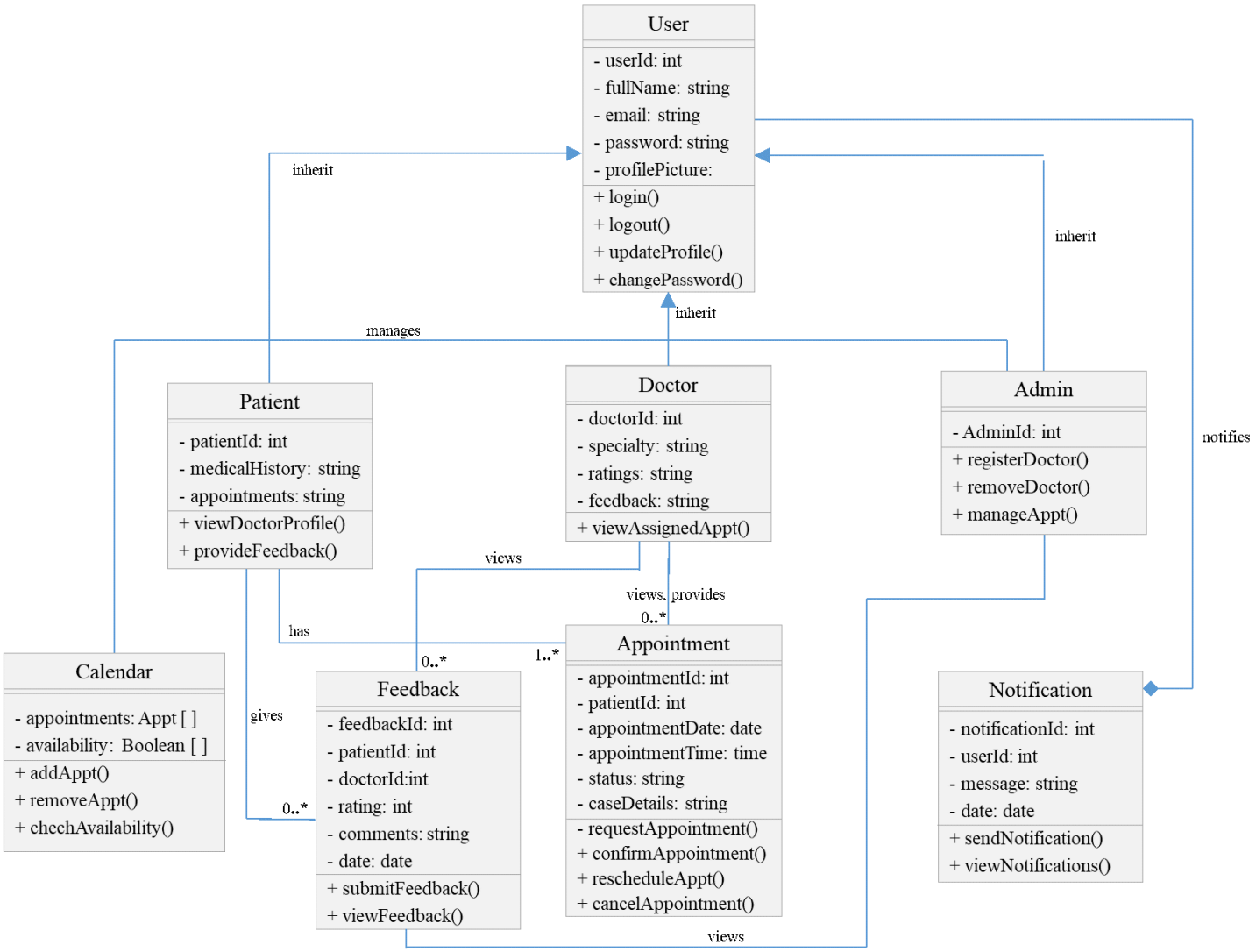


Figure 4.2: Class modeling

4.4 State chart modeling

State chart diagram shows the change of an object through time from one state to the other state. State chart modeling is used to show the sequence of states that an object goes through, the events that cause the transition from one state to the other and the actions that result from a state change. States are defined as a condition in which an object exists and it changes when some event is triggered. This section uses state charts to represent the dynamic behavior of the system. It shows the states of various objects and the transitions between these states based on events. For example, an `Appointment` might transition from "Requested" to "Confirmed" to "Completed". The following figures shows the state of the object:

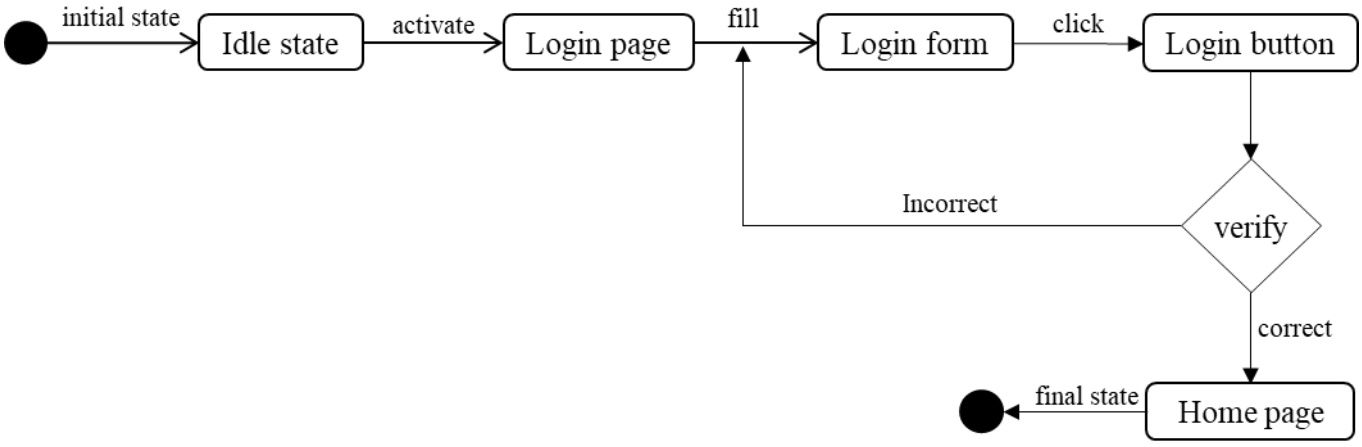


Figure 4.3: State chart diagram for login

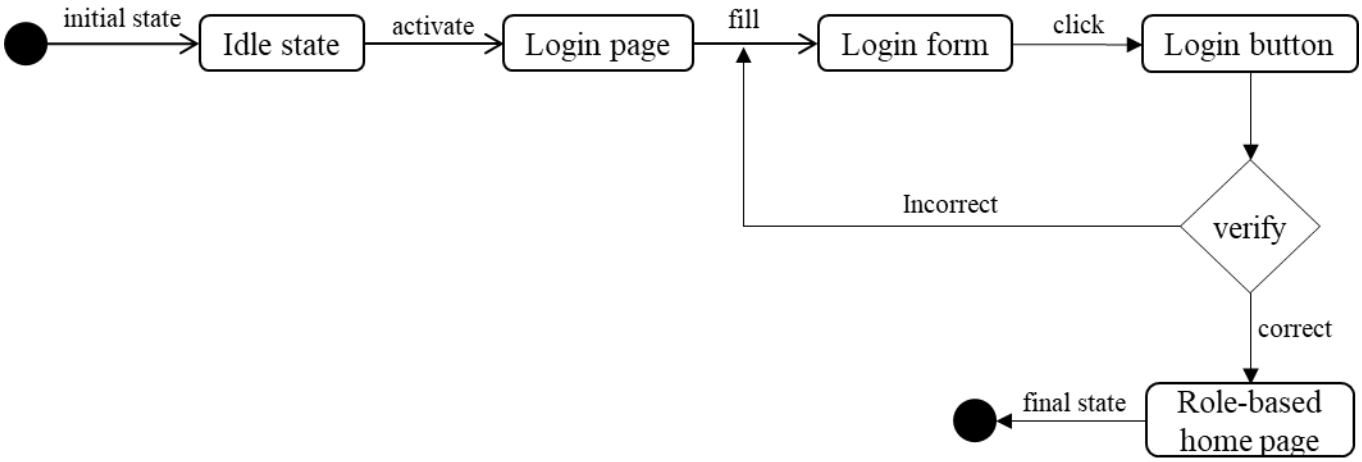


Figure 4.4: State chart diagram for admin-doctor login

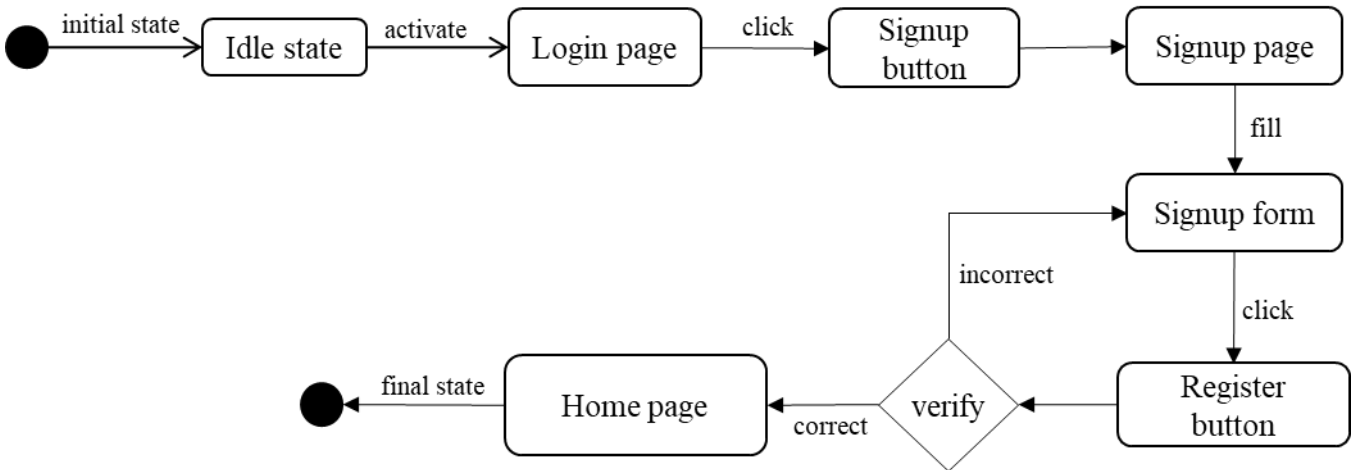


Figure 4.6: State chart diagram for patient registration

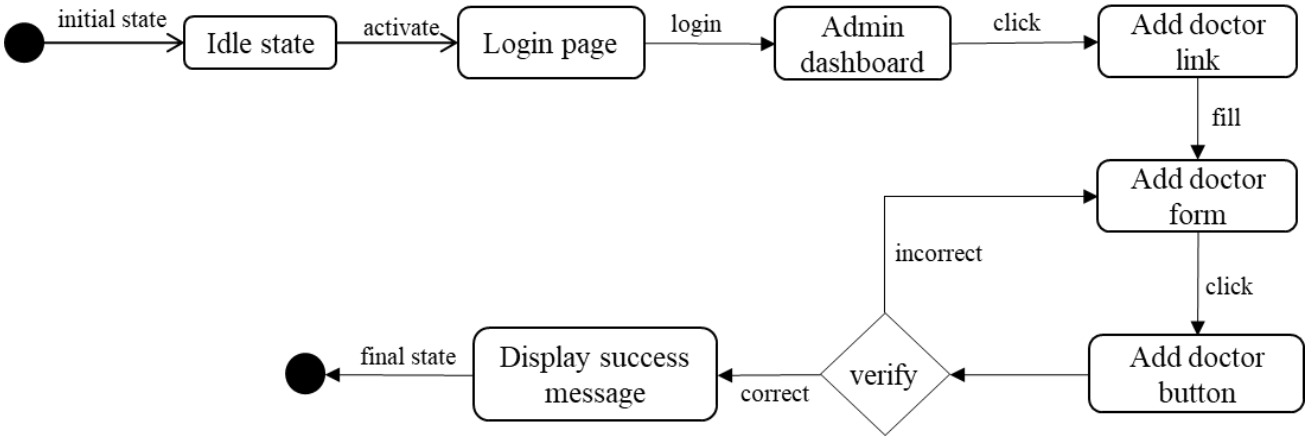


Figure 4.7: State chart diagram for adding new doctor

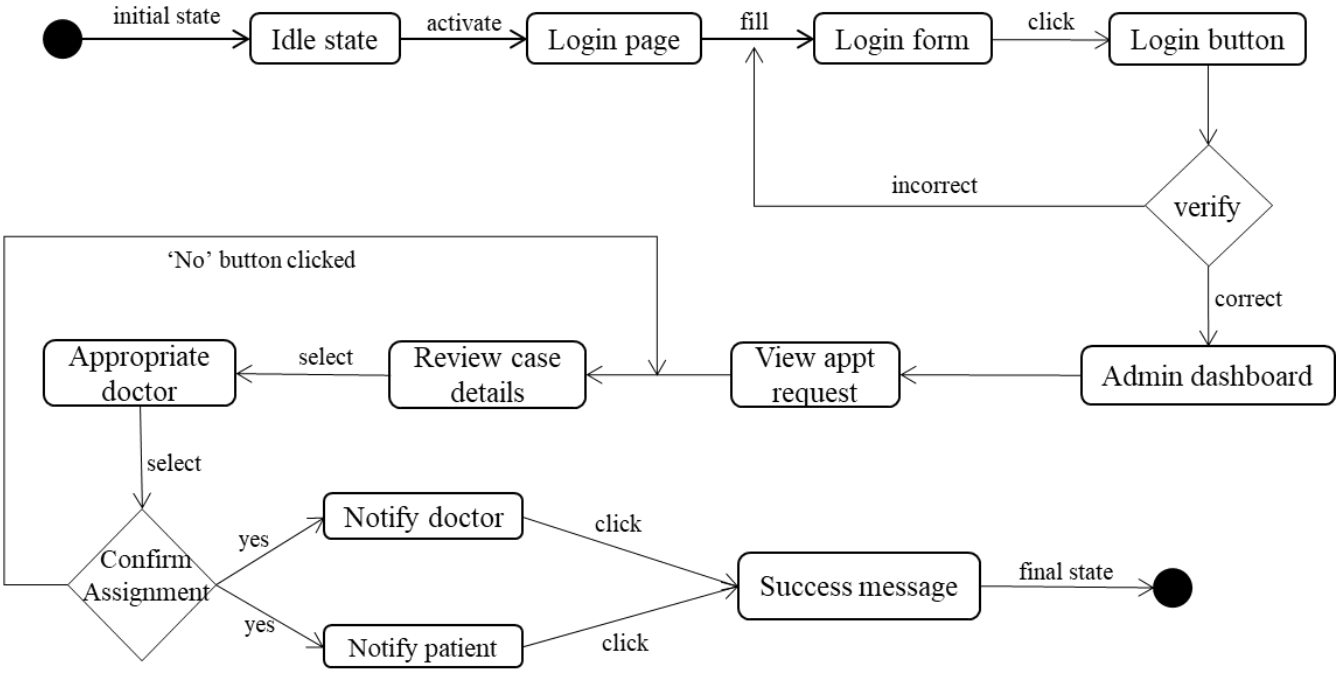


Figure 4.5: State chart diagram for assigning appointment

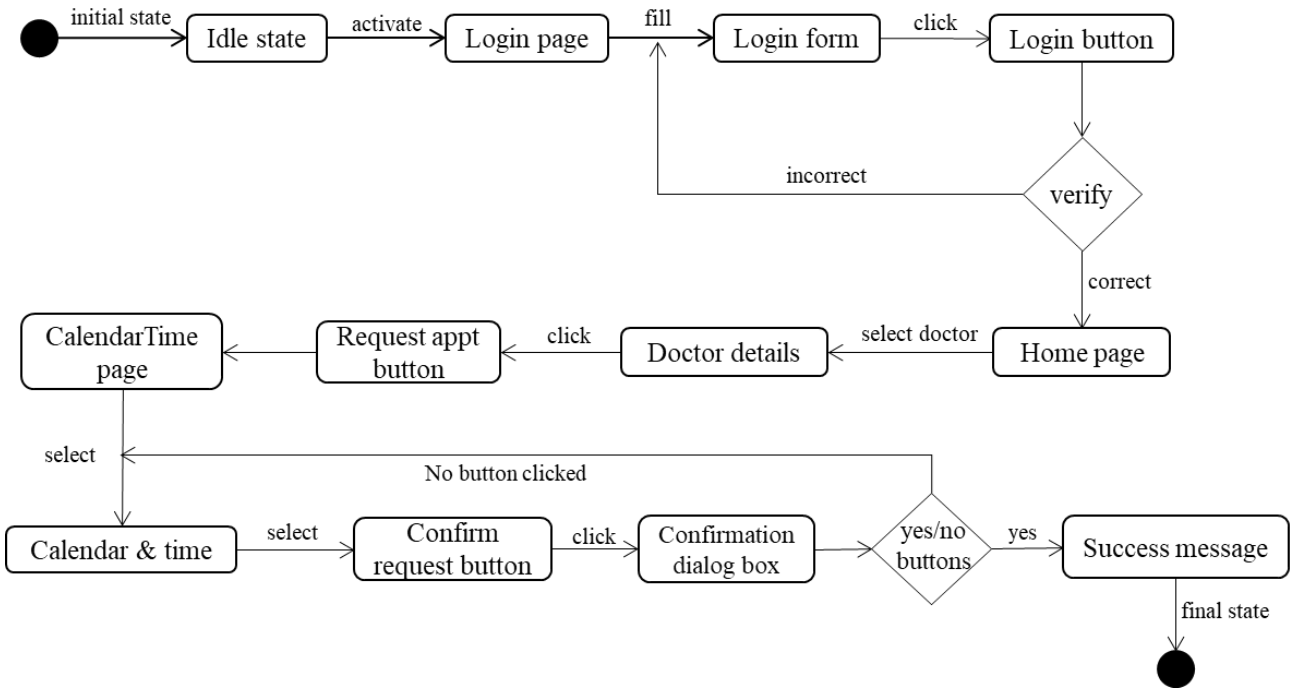


Figure 4.8: State chart diagram for request appointment

4.5 Collaboration Modeling

It represents the structural organization of a system and the messages sent/received. Structural organization consists of objects and links. This section involves creating collaboration diagrams to illustrate how objects interact to perform specific tasks. It focuses on the flow of messages between objects and the roles they play in these interactions, such as booking an appointment or processing a payment.

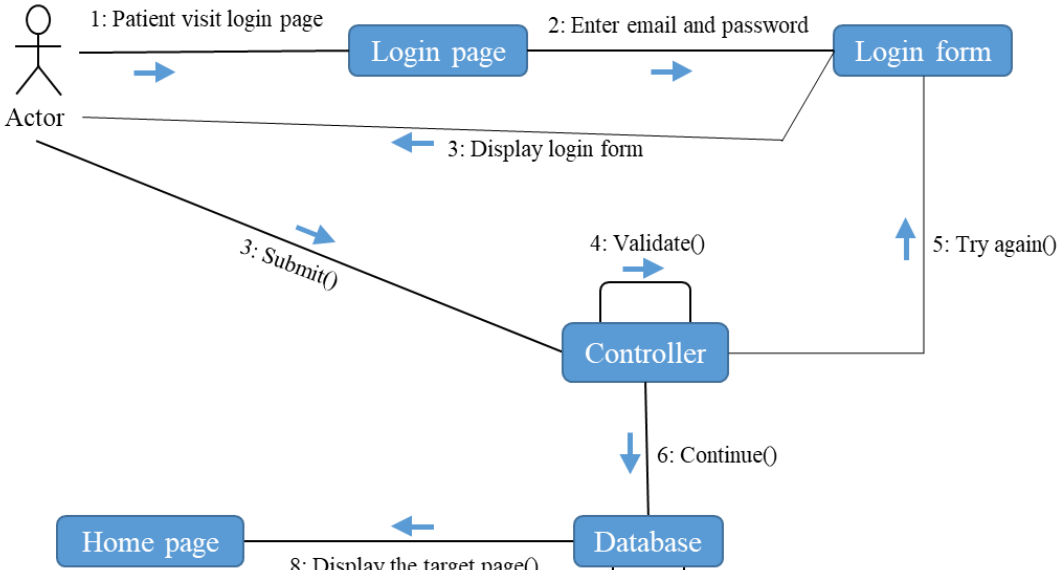


Figure 4.9: Collaboration diagram for login

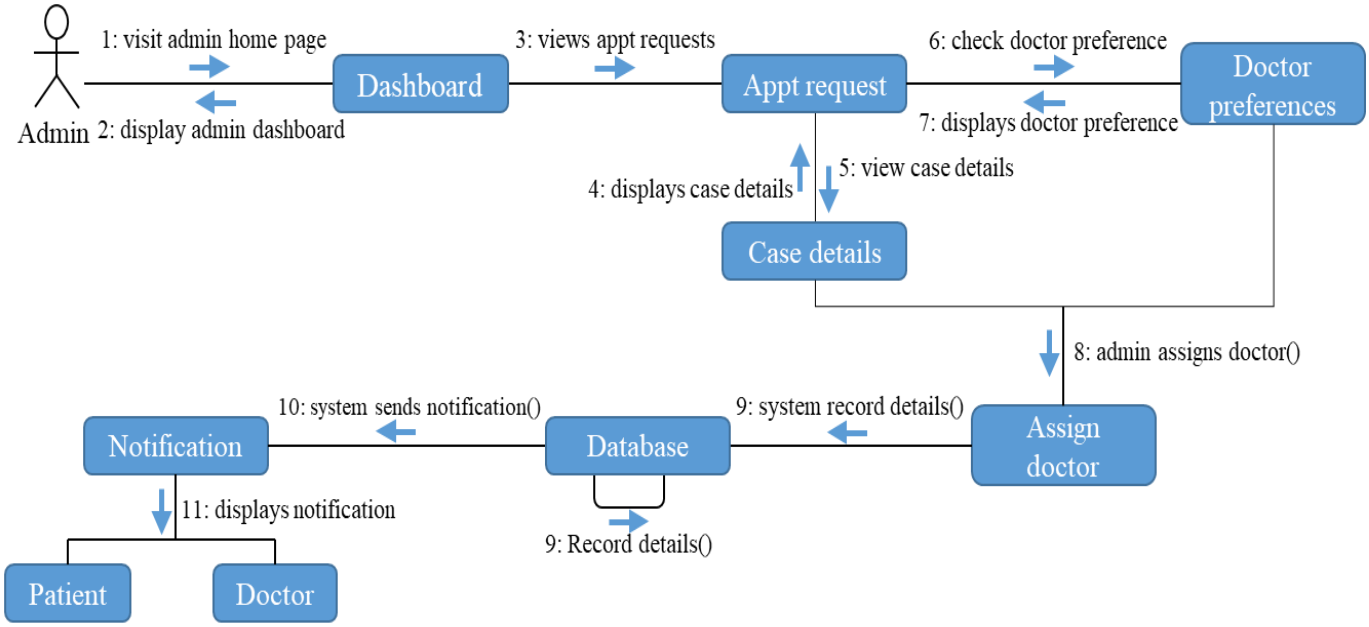


Figure 4.10: Collaboration diagram for admin assigning appt.

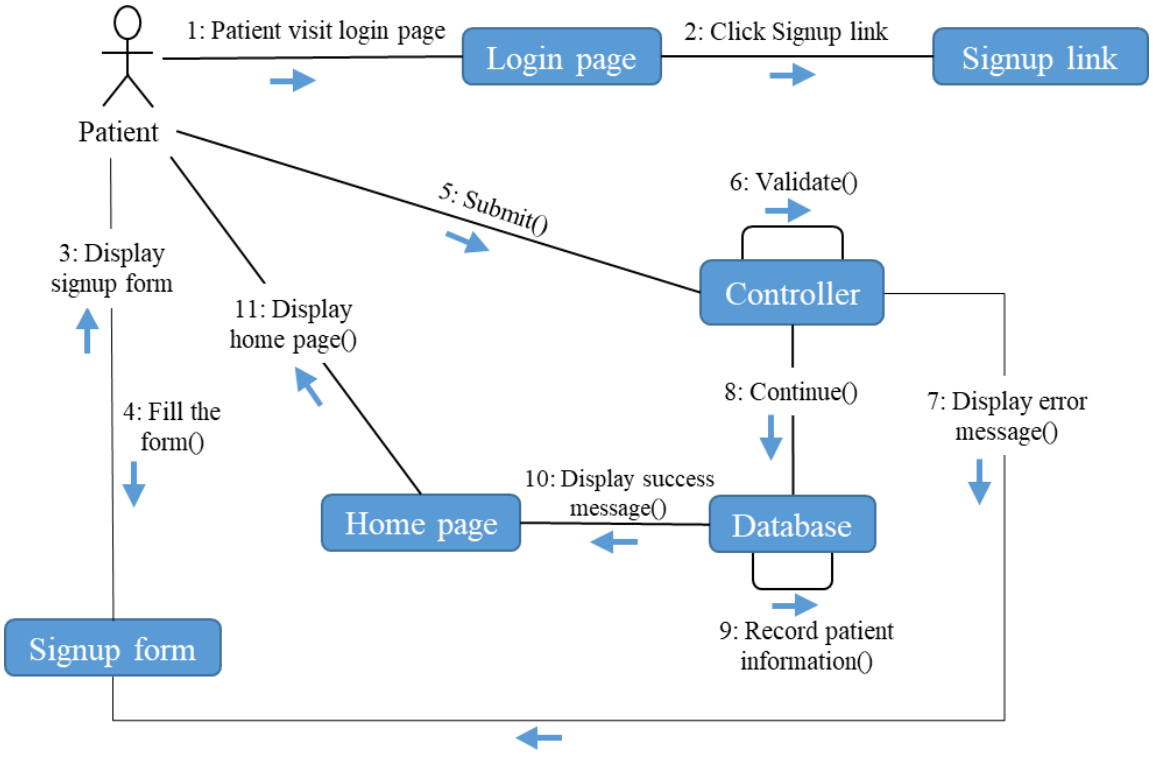


Figure 4.11: Collaboration diagram for patient registration

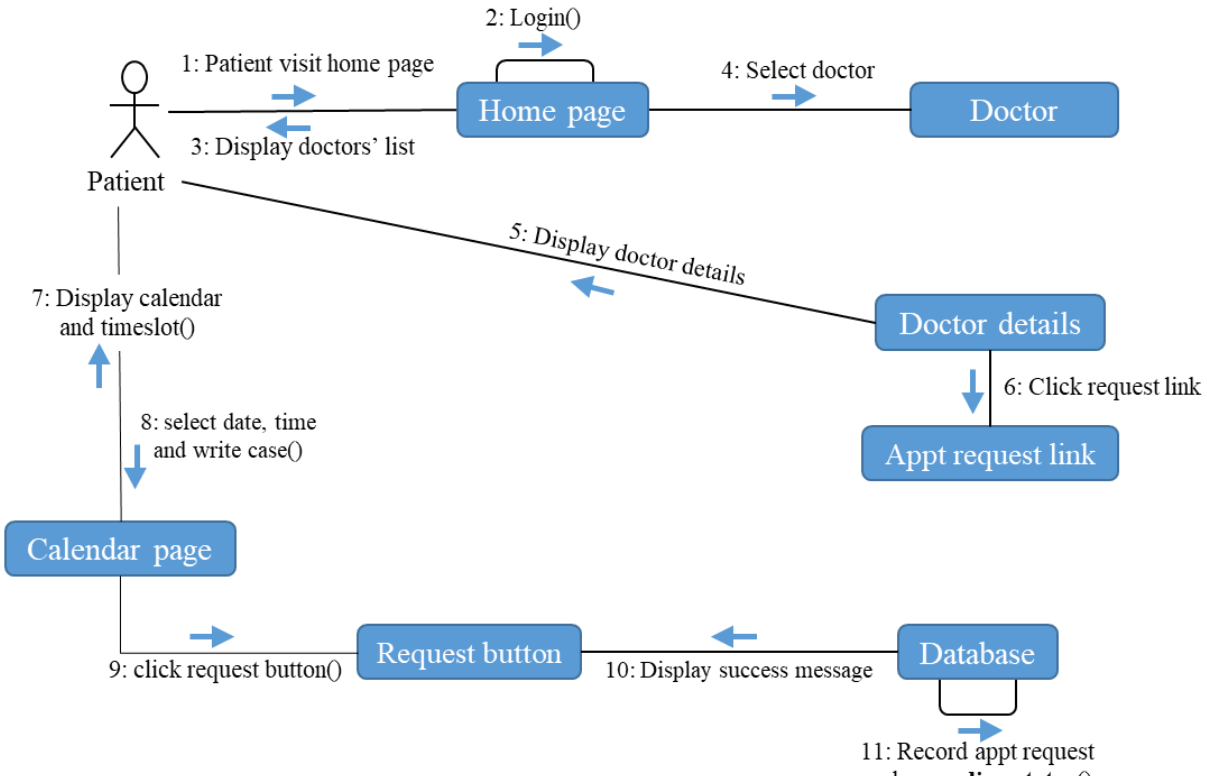


Figure 4.12: Collaboration diagram for appt. request

4.6 Component Modeling

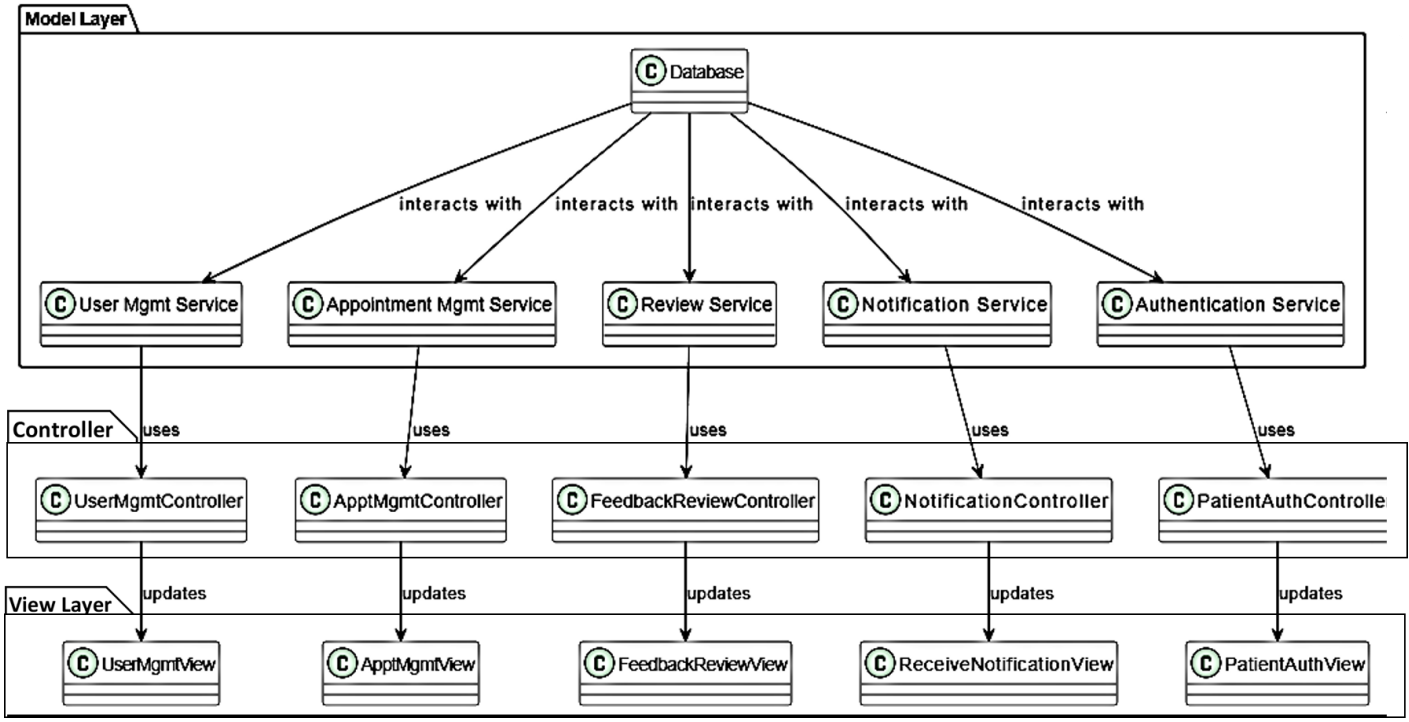


Figure 4.13: Component diagram

Component diagram is used to visualizes the software's parts, how are they organized, and how they depend on each other. This gives a high-level look at the parts of a system. This section describes the system's physical components and their interactions. It involves creating component diagrams to represent the high-level structure of the system. Components might include the database, web server, mobile app, and external services like payment gateways.

4.7 Deployment modeling

Deployment modeling is used to show the hardware of the system, the software that is installed in the hardware and also the middleware that is used to connect the disparate machines to one and other. It also shows how the software and the hardware components work together. It includes details on how software components are deployed across servers, workstations, and mobile devices. The diagram is simulated below.

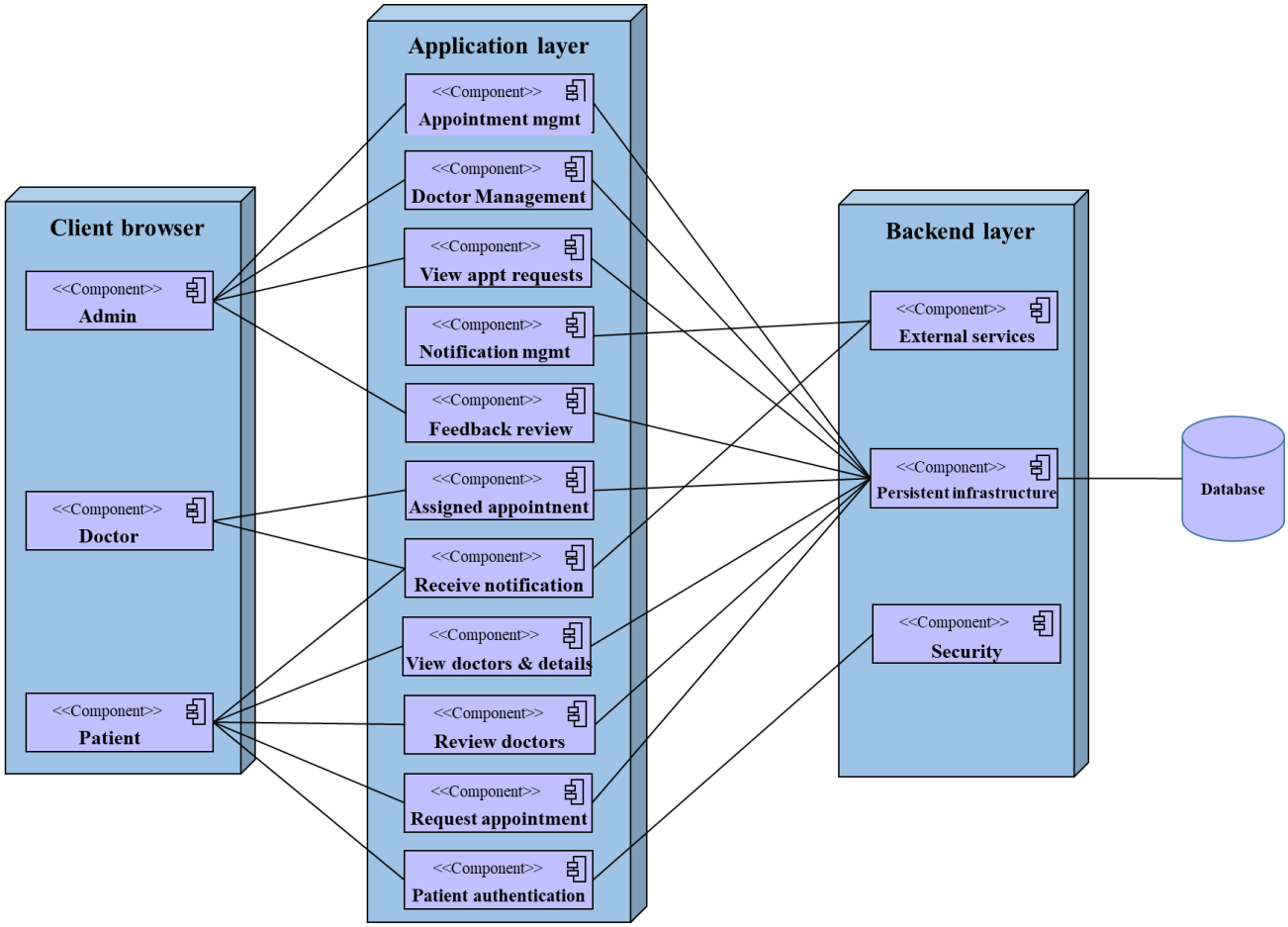


Figure 4.14: Deployment diagram

4.8 Persistence modeling

Persistence models are used to design the schema of our database. This section details how data is stored and managed within the system. It includes data models and schemas that represent the database structure, relationships between data entities, and data access patterns. We typically need to draw a persistence model whenever we are using a relational database to store our objects in. The strength of persistence models is that data entities are conceptually the same

as to the tables of a relational database and that attributes are the same as to the table columns. The following diagram shows the persistence diagram of our system.

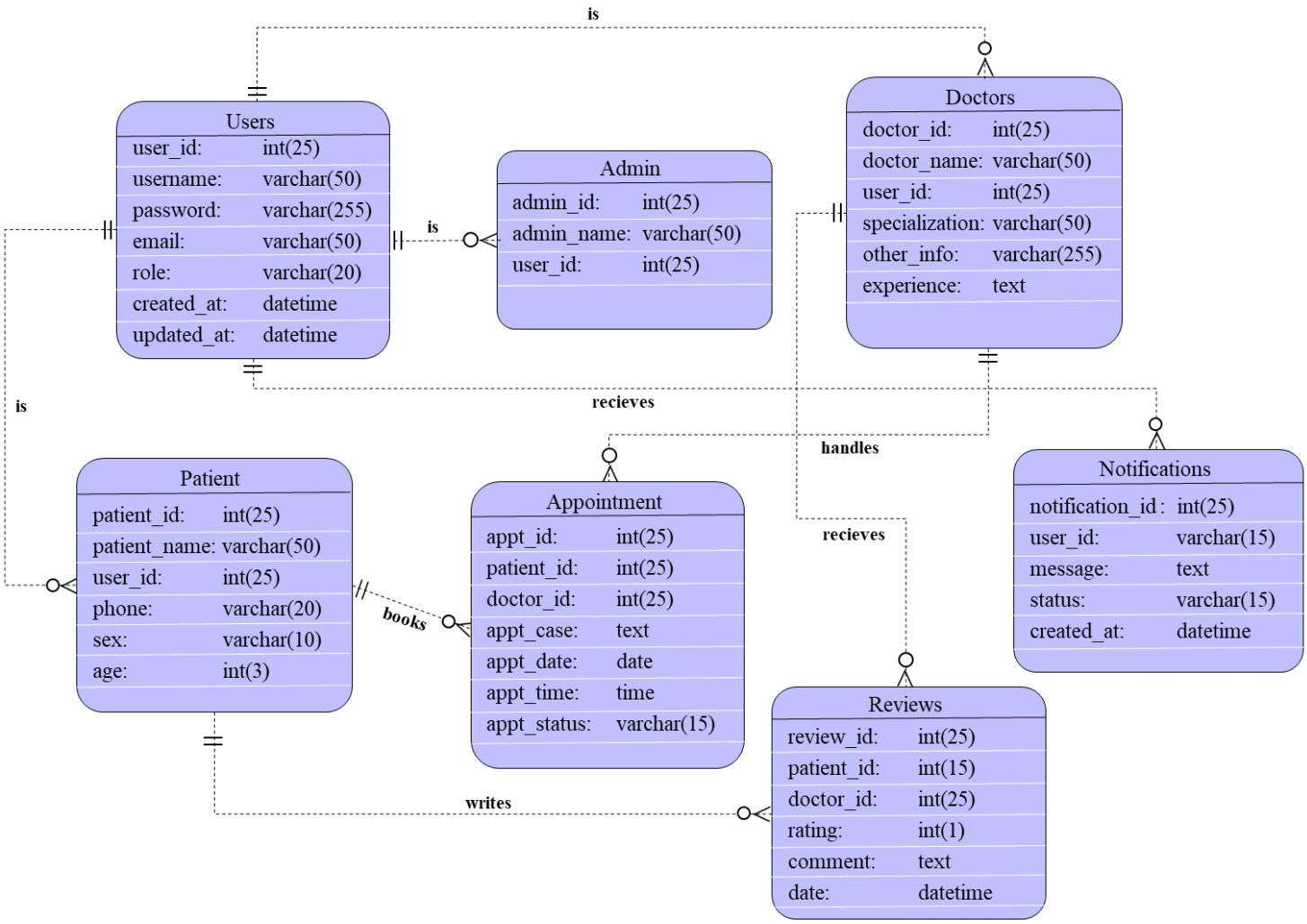


Figure 4.15: Persistence diagram modeling

4.9 User Interface design

This section provides detailed designs of the user interfaces for different user roles (patients, doctors, health officers). It includes wireframes, mockups, and design specifications to ensure a user-friendly experience.

Log in page: This page is displayed when the user opens either the mobile or web app. This page comes with a form that takes the email and password of the users. The web app has a landing page that has a login button link.

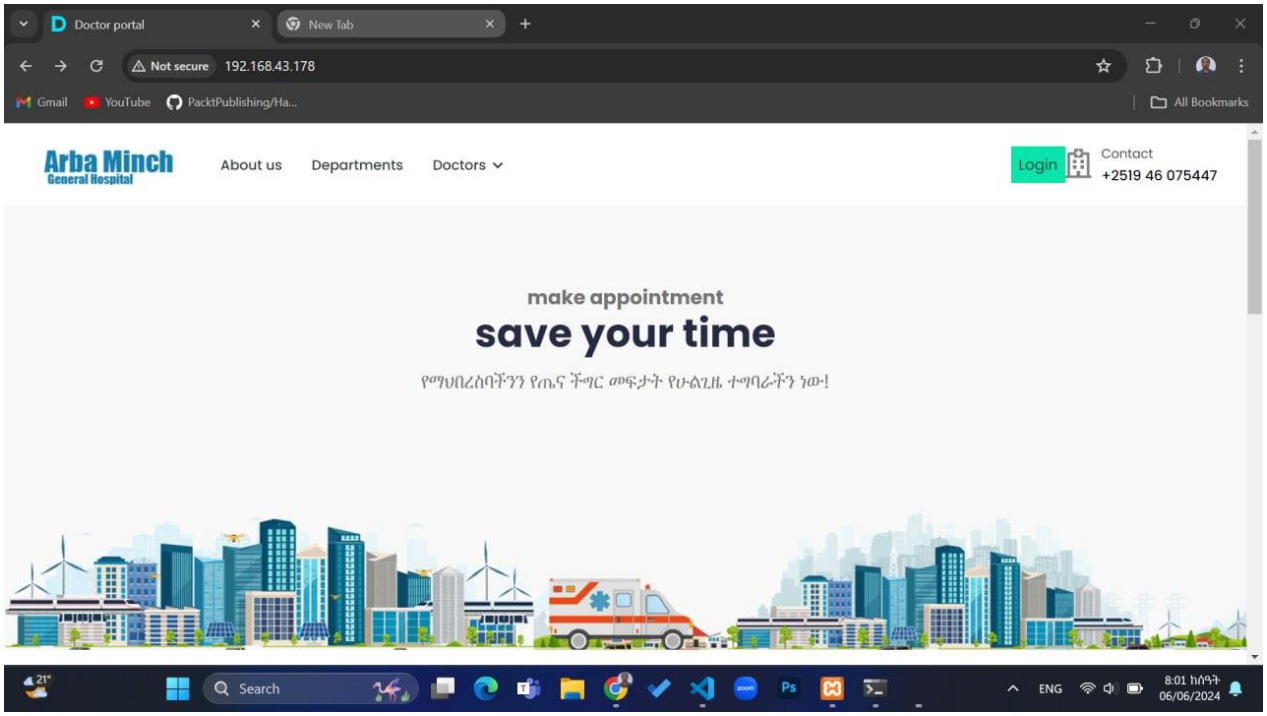


Figure 4.16: Doctor and admin landing web page

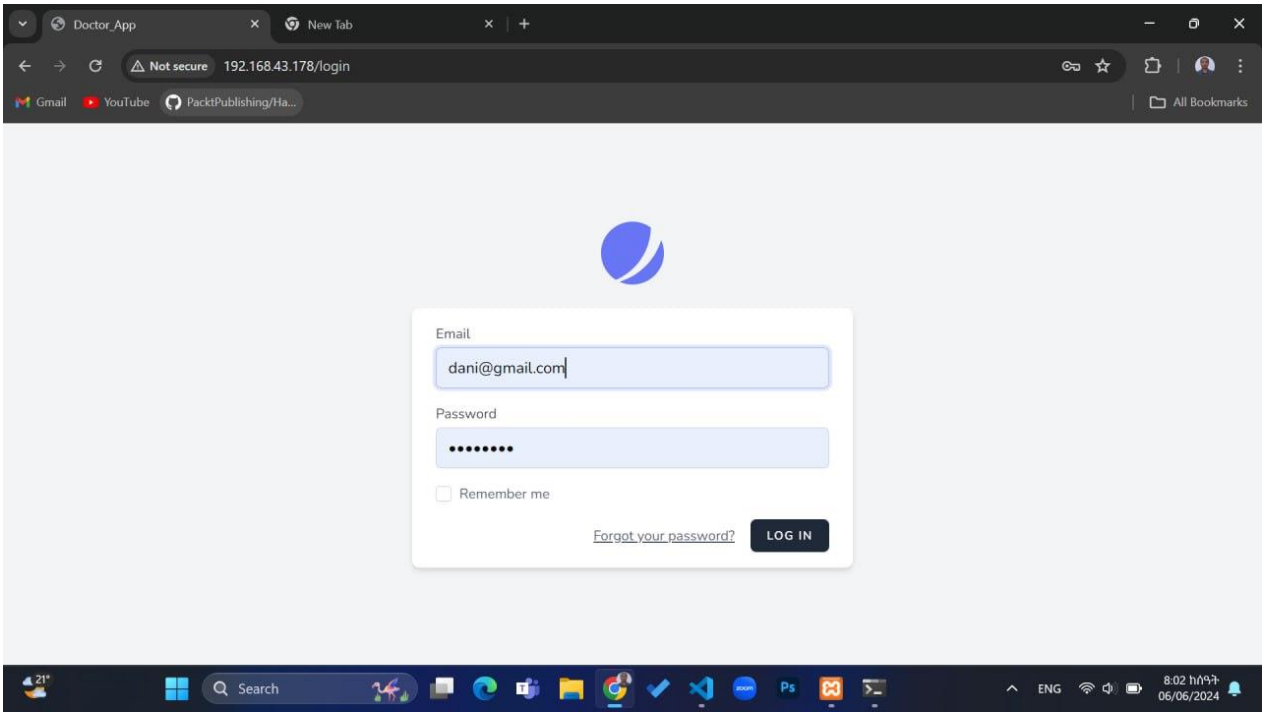


Figure 4.17: Login page for admin and doctor

Home page: On the mobile app, home page which patient access has doctor categories and doctors cards. The web-based home page for doctors displays appointments and patients assigned to the doctor. The other is for admin.

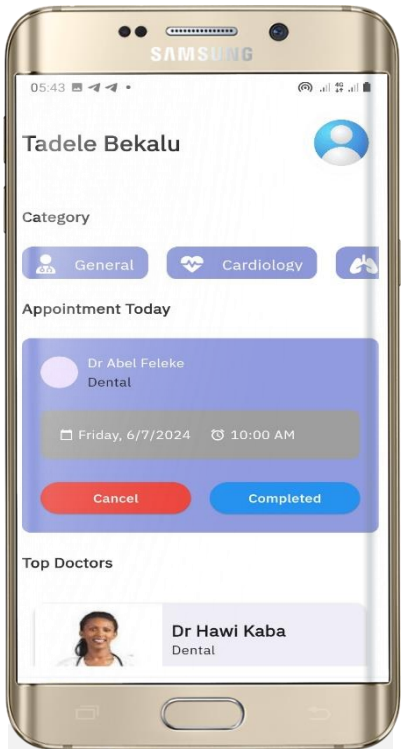


Figure 4.19: Patient home page

Doctor details page

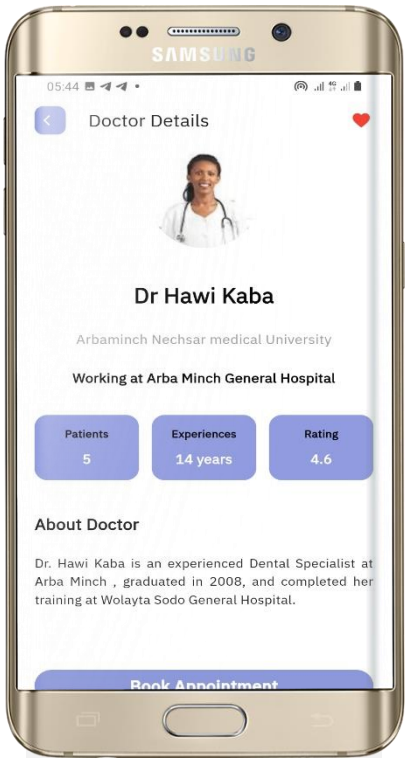


Figure 4.18: Doctor details page

Appointment request page

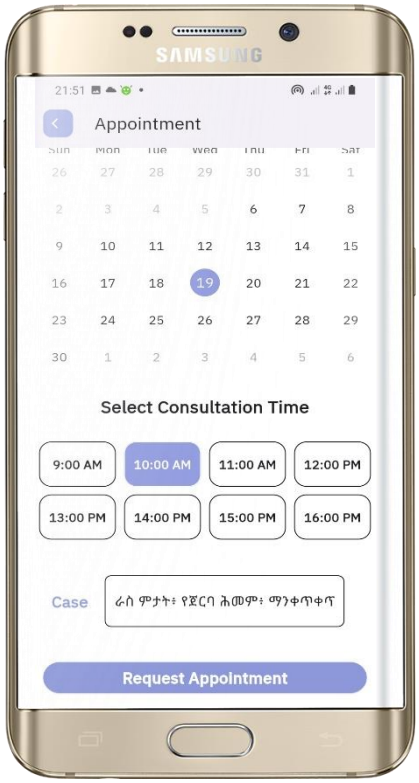


Figure 4.21: Appointment request page

Patient Sign up page

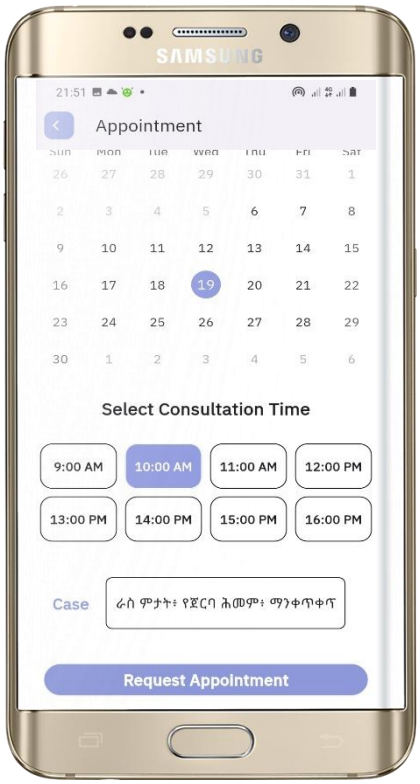


Figure 4.20: Patient Sign up page

Admin dashboard

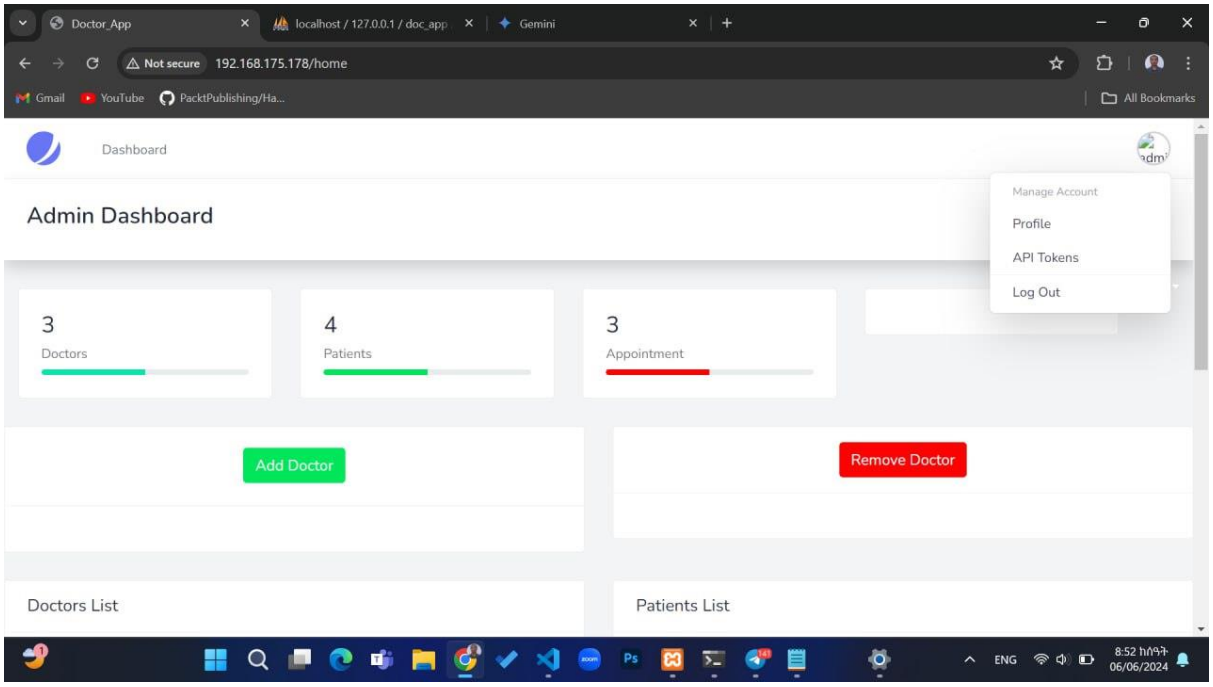


Figure 4.22: Admin dashboard

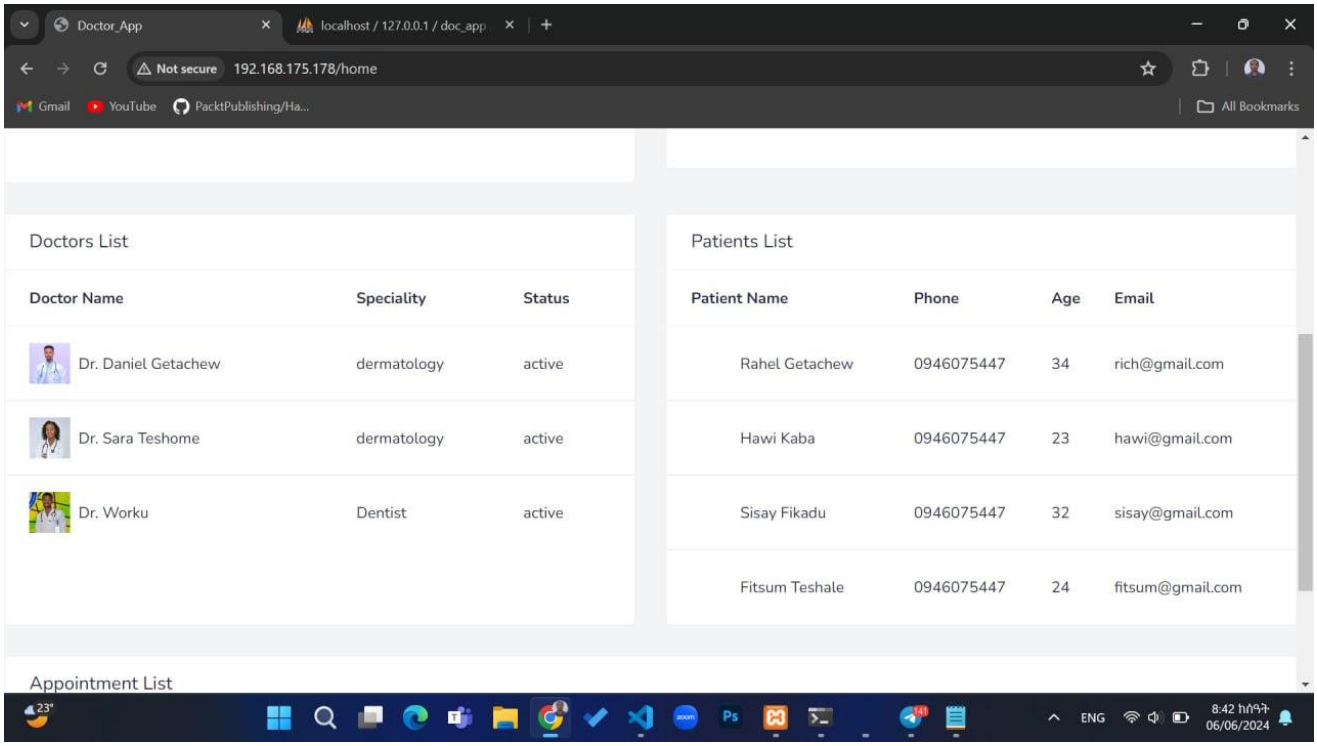


Figure 4.23: Admin dashboard-2

Pending appointments

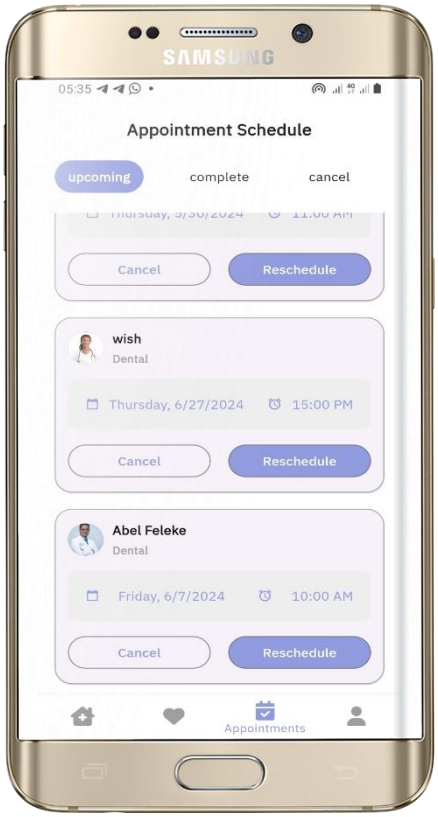


Figure 4.25: Upcoming appointments

Approved appointments

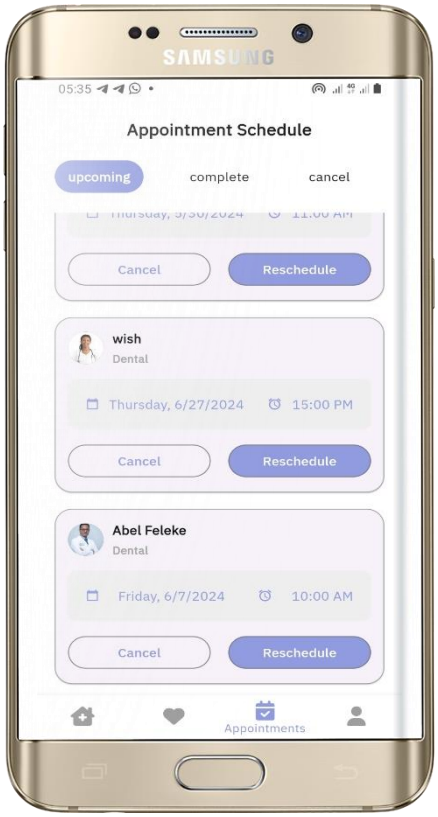


Figure 4.24: Approved appointments

CHAPTER FIVE

IMPLEMENTATION AND TESTING

5.1 Introduction

In this chapter, we mainly focus on the implementation and testing aspects of the Doctor Appointment System. This includes the type of material (hardware and software required), techniques to develop the system, sample codes of the system, data preparation, installation procedures, testing techniques, and the startup strategy for the newly installed system. These elements are briefly described in this part of the documentation to ensure a comprehensive understanding of how the system is brought to life and made operational for all user roles, including patients, doctors, and health officers.

5.2 Final Testing of the system

Testing is an improvement phase. This phase involves testing of developed system using different form of data. Testing is the process of executing a program with the explicit intention of finding errors that is making the program fail. Testing is one of the validation techniques of the system with acceptable standard inputs from the different testing types. The following are some of them.

5.2.1 Unit Testing

Unit testing is where individual components or units of a software application are tested in isolation to ensure that the code behaves as intended and meets the design specifications. It involves writing test cases for each unit, executing them to validate the output, and analyzing the results to identify any defects or failures. Some of the test cases that we write, execute and analyzes are listed in the following tables:

Table 5.1: Test Case for Login Page

Test Case ID: TestCase1			
Unit to Test: Login Page			
Item Name	**Test Description**	**Expected Results**	**Actual Results**
email	Leave it blank.	An error message that informs the user to enter email.	Error message: “Enter email”
	Type wrong email	Error message tells users that email is invalid.	Error message: “Enter valid Email”

	Type Correct Username	No error was expected	No Error
Password	Leave it blank.	An error message that informs the user to enter password.	Error message: “Enter password”
	Type Short Password	An error message that tells the user that password should not be less than 6 characters.	Error message: “Password length should be more than 6 characters.”
	Type Wrong Password	An error message tells users that password is incorrect.	Error message: “Incorrect password”
	Type Correct Username	No error was expected	No Error
Login Button	Click on the login button	If password and email is correct, user successfully logged into the system.	User logged in successfully with correct email and password.

Table 5.2: Test Case for patient registration page

Test Case ID: TestCase2			
Unit to Test: Patient signup page			
Item Name	**Test Description**	**Expected Results**	**Actual Results**
Full name	Leave it blank.	An error message that informs the user to enter full name.	Error message: “Enter full name”
	Type full name with a number or special character.	An error message that tells the user that full name must be a letter.	Error message: “full name must be a letter”
	Type full name with a letter only.	No error was expected	No Error
Email	Leave it blank.	An error message that informs the user to enter email.	Error message: “Enter email”
	Type email without @ and domain	An Error message tells user that input is not an email.	Error message: “Input is not an email”
	Type email with @ and domain	No error will be expected	No Error
Age	Leave it blank.	An error message that informs the user to enter age.	Error message: “Enter your age”
	Type age with a letter or special character	An error message that tells the user that age must be number.	Error message: “Age must be a number”
	Type Correct Age	No error will be expected	No Error
Password	Leave it blank.	An error message that tells the user to enter password.	Error message: “Enter password”

	Type Short Password	An error message that tells the user that password should not be less than 6 characters.	Error message: "Password should not be less than 6 characters"
	Type Correct Password	No error will be expected	No Error
Signup Button	Click on the signup button	If all fields are correct, user successfully register as a patient in the system.	Patient account created successfully with correct fields.

Table 5.3: Test Case for appointment request page

Test Case ID: TestCase3			
Unit to Test: appointment request page			
Item Name	**Test Description**	**Expected Results**	**Actual Results**
Past date selection	When past dates are to be selected	The dates should not be clicked.	A date is not clickable if it is already a past date.
Proper date selection	When the coming dates are selected	The system should show the selected date making it different.	System highlights the date a patient selects.
Time selection	When the coming dates are selected	The system should display timeslots.	Timeslots are displayed and if one is selected, it is highlighted.
Already booked Timeslot	When a patient requests appt on a time already appt is booked.	The system should prevent selecting an already booked timeslot on the same date.	Error message: "The date and time you select is already booked."
Request button	When proper calendar & time is not selected	The button should not respond nothing.	The button is not clickable.
Request button	When proper calendar, time is selected & patient clicks it	A patient can successfully request appointment.	The button is enabled and the patient successfully request appointment.

Table 5.4: Test case for assigning doctor & approving appointment

Test Case ID: TestCase4			
Unit to Test: appointment assigning and approving			
Item Name	**Test Description**	**Expected Results**	**Actual Results**
Assigning doctor	Verifying admin can verify or reassign a doctor to an appt.	The appt. should be kept as it was or be updated with the reassigned doctor ID.	The appt. is updated with the assigned doctor ID or kept as it was.

Approving appt.	Verifying admin can approve pending appointments	The appointment status should be updated to "Approved"	The appointment status is updated to "Approved".
Notification on approval	Verifying that a notification is sent to the patient & doctor upon approval	Notifications should be sent to both patient & doctor indicating the appointment has been approved.	Notifications are sent to both patient & doctor.

Table 4.5: Test Case for Notification system

Test Case ID: TestCase5			
Unit to Test: Notification system			
Item Name	**Test Description**	**Expected Results**	**Actual Results**
Sending notification	Verifying notification on doctor approval	System should sent notification to the patient about the accepted appointment request.	Message: "Your appointment request is accepted"
Sending notification	Verifying notification on appt. request	System should sent notification to admin as a patient request and appointment.	
Rejecting appointment	Verifying notification on appt. rejection	Notification should be sent to the patient about the appointment rejection.	
Upcoming appointment reminder	Verifying notification for upcoming appt.	Reminders should be sent to both the patient and the doctor as a set time of the appointment approaches.	
Notification description	Verify notification content accuracy	Notification content should accurately reflect the action taken	

Table 5.6: Test Case for doctor management

Test Case ID: TestCase6			
Unit to Test: Doctor management			
Item Name	**Test Description**	**Expected Results**	**Actual Results**
Adding doctor	Admin adding new doctor	The system should register the doctor and add his/her information to doctor database table.	The new doctor is added to the database under doctor table.
Removing doctor	Admin removing registered doctor	The system should drop the doctor from the doctor database table.	The doctor is removed from database doctor table.
Updating doctor	Admin updates doctor details	System should retrieve existing information and update on the DB.	The information modified are updated on the database

Table 5.7: Test Case for feedback and review

Test Case ID: TestCase7			
Unit to Test: feedback and review			
Item Name	**Test Description**	**Expected Results**	**Actual Results**
Feedback submission	Submitting empty feedback form	Error message indicating feedback cannot be empty	
	Verifying patient can submit feedback	Feedback is successfully submitted by patient and saved in the system	
Viewing feedback	Empty feedback	Message indicating no feedback available	Message: "No available feedback yet"
	Verifying doctors & admin can view feedback	List of feedback should be displayed if exist	List of feedbacks are displayed.
Feedback mgmt.	Admin delete feedback	System should delete the feedback.	Message: "Feedback is successfully deleted"
	Admin replies for feedback	Feedback is successfully approved	Message: "Feedback is successfully approved"

Table 5.8: Test Case for security issues

Test Case ID: TestCase8			
Unit to Test: security issues			
Item Name	**Test Description**	**Expected Results**	**Actual Results**
Session management	Verifying session expire after a certain time of inactivity	After a specified time, the system should logout the user.	
Authorization & access control	Verifying users has only access pages & actions based on their roles	System should allow access based on user's role.	
Data encryption	Verifying sensitive data is encrypted	Password should be stored as encrypted (hashed).	

Table 5.9: Test Case for Appointment management

Test Case ID: TestCase9			
Unit to Test: Appointment management			
Item Name	**Test Description**	**Expected Results**	**Actual Results**
Appt. rescheduling	Verifying admins can reschedule appt.	Appointment should be rescheduled and confirmation shown.	

Appt. cancelling	Verifying appointments can be canceled by patients, or admins	Appointment should be cancelled and confirmation shown.	
	Cancellation Without Confirmation	Appointment should remain as scheduled.	
Appt. reminder	Verifying reminder notifications are sent to patients & doctors	User should receive a reminder notification.	

Table 5.10: Test Case for mobile app-specific

Test Case ID: TestCase10			
Unit to Test: Mobile app-specific			
Item Name	**Test Description**	**Expected Results**	**Actual Results**
Navigation	Verifying navigation within the mobile app	App should navigate effectively between pages	
Device compatibility	App on different screen sizes	App should display correctly on different screen sizes	
	App on different OS versions	App should perform similarly	
Offline access	Verifying the app handles offline scenarios	App should open and display cached content	
	Trying to perform actions offline	App should notify user that actions require internet	

5.2.2 Integration Testing

Integration testing is a type of software testing conducted to verify the compatibility and communication between different software modules or components that have been integrated into a larger system. We will verify that different modules (patient interface, doctor interface, health officer interface) work together seamlessly. This involves testing the interactions between these modules to ensure data flows correctly and processes are executed as expected. It ensures that each module interacts as intended, and all the modules together produce the desired output. Therefore, we have tried to test our doctor appointment system for its integration that justifies the major component of the system are working almost as per the requirements gathered.

5.2.3 System Testing

System testing is a type of software testing where the entire system is tested as a whole to ensure that it meets the functional and non-functional requirements. The objective of this phase

is to identify any defects or discrepancies with the system, so that they can be fixed before releasing it to the end-users. This phase will test the entire system as a whole to ensure it meets the specified requirements. We will simulate real-world scenarios to identify any defects or issues. In this testing level, we try to compile the whole system starting from the initial phase and proceed testing the whole system to check out for the errors and flow control of the system.

5.3 Hardware software acquisitions

In our project, acquiring the right hardware and software is essential for the system's performance and scalability. For our project implementation, we used the following hardware and software.

Hardware: We will procure servers for hosting the web application, secure storage solutions for patient data, and mobile devices for testing the mobile application. For our system's implementation the following hardware are used:

- Computers/Laptops
- Wi-fi
- Network connection.
- Web server
- Database server
- Printer to print hard copy.
- Flash
- Hard disk for backup

Software: We will use Flutter for mobile app development and Laravel for web development. Additionally, we will implement secure payment gateways and a structured database management system (MySQL) for efficient data handling. For the system effective implementation the following software are used:

- Bootstrap Framework for front-end development.
- Laravel PHP framework for back-end development
- SQL for handling a database operation.
- Microsoft Office word 2019
- PowerPoint for preparing presentation
- Window 10 operating system
- Visual studio code.
- Android Studio

- XAMP server
- Draw.io website for designing UML diagrams

5.4 User manual preparation

In our project, preparing a comprehensive user manual is critical for ensuring all users can effectively navigate and utilize the Doctor Appointment System. The manual is structured to provide clear instructions tailored to the specific roles within the system: Admins, Patients, Doctors, and Health Officers. Each section of the manual is designed to empower users with the knowledge necessary to maximize their interaction with the system.

- For Admins:

  The user manual section dedicated to admins will comprehensively cover essential tasks such as logging in securely, managing user accounts, registering and managing doctor profiles, handling appointment requests, assigning or approving doctors based on case details, monitoring patient feedback, performing database operations, updating profiles, and changing passwords. The manual will guide them on managing doctor profiles, overseeing appointment requests, assigning doctors, observing statuses and monitoring system performance.

- For Patients:

  Patients will find detailed instructions in their section of the manual on logging into the system securely, viewing doctor profiles, scheduling appointments, providing feedback and ratings, accessing health tools and educational content, updating their profiles, and managing their passwords. The manual will include instructions on how to view doctor profiles, schedule appointments, provide feedback, and access health tools.

- For Doctors:

  The manual for doctors will provide clear guidance on logging into the system, managing their schedules, handling appointment requests, viewing patient feedback and ratings, receiving notifications, updating their profiles, and changing passwords to ensure efficient and effective utilization of the system. It will cover how to manage schedules, view patient feedback, receive notifications, and update their profiles.

The user manual will also include general guidelines covering system navigation, visual aids such as screenshots and diagrams to illustrate key processes, a troubleshooting section with common issues and solutions, and contact information for support and assistance.

5.5 Training

Training is the process of teaching or learning a skill or job. Training is needed for two reasons:

- If users are not adequately trained, they will not operate the system correctly or efficiently.
- If users fill that they are being asked to perform tasks that are outside their capabilities, they may become demoralized and separated.

Training plays a crucial role in ensuring that all users, including health officers who serve as admins, are proficient in utilizing the Doctor Appointment System effectively. The training program will be designed to cater to the specific needs and roles of each user group—Admins, Patients, Doctors—to maximize their efficiency and confidence in using the system. Each user group's training content focuses on specific functionalities and tasks pertinent to their roles within the Doctor Appointment System.

We try to simplify the learning curve by automating and digitizing existing manual workflows. The manual way workflow is kept as it is to easily adapt to with the system. This transition ensures that users can seamlessly adapt from traditional methods to the new automated processes facilitated by the system.

5.6 Installation Process

The installation process describes the steps required to successfully deploy and make the Doctor Appointment System operational for its intended users. In the context of your Doctor Appointment System, where the web application is primarily for admin/health officers and doctors, and the mobile app is exclusively for patients, the term "installation" may not apply in the traditional sense of installing software on a specific machine. Instead, here's a clearer perspective:

Web Application (Laravel)**:**

The web application will be hosted remotely on free IaaS cloud services.

- For admin/health officers and doctors, accessing the web application involves navigating to a specific URL using a web browser. This doesn't typically require installation on individual machines but rather access through a web portal and can be accessible through mobile or PC web browsers. Users authenticate and access the system via their browsers, interacting with the application hosted on a server. Upon accessing the URL, admins and

doctors authenticate using their credentials and upon their role, they access their respective dashboards and tasks in the system.

Mobile Application (Flutter)**:**

- Unlike admins and doctors, patients download and install the mobile application from app stores like Google Play Store (Android) or Apple App Store (iOS), after successfully hosted there, or doctors can also assist patients in getting the app. This is a straightforward process where users install the app on their mobile devices to access the system's functionalities designed specifically for patients.

Mobile Application (Flutter):

- Patients can download the Doctor Appointment mobile app from the Google Play Store (Android) or Apple App Store (iOS) after successfully hosted there, or doctors can also assist patients in getting the app. After installation, patients launch the app on their mobile devices. Patients authenticate themselves through the app using their credentials provided during registration.

5.7 Start-up strategy

As we told earlier our system has two parts: web app for admin and doctors, and mobile app for patients. The web app will start as soon as the admin or doctor enters the url of doctor appointment system website on his/her web browser. For further interaction and to able to access different features of the system, they need to login into the system by providing their email and password. After the user successfully logged into the system, they able to access the features provided by the system based up on their role.

The mobile app will start as soon as the patient opens the mobile app of doctor appointment system on his/her phone. For further interaction and to able to access different features of the mobile app, he/she needs to login into the system by providing email and password. After the user successfully logged into the app, they able to access the features provided by the system such as rating doctors, making appointment, seeing appointment statuses, receiving notifications and etc.

CHAPTER SIX

CONCLUSION AND RECOMMENDATION

6.1 Conclusions

The Doctor Appointment System represents a pivotal advancement in healthcare management, streamlining the process of scheduling medical appointments and enhancing patient-doctor interactions. This integrated platform addresses key challenges faced in traditional appointment booking systems by providing efficient scheduling, secure communication channels, and comprehensive feedback mechanisms.

Our system is designed to:

- Facilitate seamless appointment scheduling for patients, improving accessibility and convenience.
- Enable doctors to manage their schedules efficiently, receive timely notifications, and enhance patient care through better communication.
- Empower health officers (admins) to oversee the appointment system, ensuring optimal doctor-patient matches and operational efficiency.

Working on the project was an excellent experience. It helped us to understand the importance of planning, designing and implementation so far we have learnt in our theoretical education. It helped us to discover our creativity while working in a team. It also realized the importance of team working, communication as a part of this project. The Doctor Appointment System project, while not fully completed, has undergone significant development involving a lot of efforts and numerous work hours. Finally, we like to conclude that we put all our efforts throughout the development of our project and tried to fulfill most of the requirements of the user.

6.2 Recommendations

In light of our project's implementation and observations, we recommend the following enhancements and considerations for future developments to the respective bodies:

- **Integration with External Systems**:

Since our system currently does not support secure payment transactions, we recommend exploring opportunities to integrate with real-time banking systems to streamline financial transactions securely and efficiently.

- **Expansion of Language Support**:

  To broaden accessibility and user reach, future developers could consider adding support for additional languages like Gamotto and Amharic especially within the mobile app, accommodating diverse linguistic preferences and improving user engagement.

- **Continuous Improvement**:

  This system has some limitations because of lack of time, overloaded time schedules and connection at our university, so we highly recommend that the university fulfill the material in the learning and teaching process and also manage project schedules with class schedules and exams.

REFERENCES

1. https://link.springer.com/chapter/10.1007/978-3-031-08782-0_25
2. https://bmchealthservres.biomedcentral.com/articles/10.1186/s12913-022-08635-6
3. https://www.mdpi.com/2227-7390/10/22/4317
4. https://link.springer.com/article/10.1007/s12553-021-00547-5
5. https://effectivehealthcare.ahrq.gov/system/files/docs/topic-brief-scheduling.pdf