College of Computer and Information Sciences

Group 18 | 46475

#	Student Name	ID
1	Sara Alaiban	443200572
2	Lama Almazyad	443200594
3	Rafeef Alowayed	443203089
4	Munirah Al Obaid	443201024
5	Najlaa Alhazani	443200848
6	Ruba Alrabeia	443200453

Supervised By:
Dr.Abir Najjar & Rawan Alabdulrahman

Table of Contents

Revision Control History

Version Number	Date	Description
1	16 Feb 2024	Adjusted any inconsistencies, requirements correction
	17 Feb 2024	Update table of contents, format document
2	29 Feb 2024	Applied Changes to phase 1 based on the feedback
	27 Mar 2024	Architectural Style, use case and use case description review and corrections.
3	1 May 2024	Phase 3 diagrams
	12 May 2024	Final revision

Introduction

MedUp a revolutionary healthcare platform designed to disentangle and improve post-procedure checking for patients and healthcare providers. Patients regularly confront challenges to viably communicate their symptoms after a medical procedure, whereas healthcare providers struggle to evaluate and oversee the growing amount of patient information within the healthcare industry. **MedUp** points to address this challenge by enabling patients and making stride healthcare.

MedUp employments Al-based side effect severity assessment and image processing to provide patients with a user-friendly interface to record and capture symptoms. The platform guarantees fast communication by cautioning patients to high-severity symptoms and encouraging consistent associations with healthcare providers Through real-time chat or call features. For healthcare experts, **MedUp** gives a centralized system to track patient symptoms, give clear feedback on image assessments, and get real-time notifications for high-severity cases.

Moreover, the platform simplifies appointment scheduling, gives medication reminders and comprehensive information on prescribed medications, advances a caring approach centering on patient information, and puts the patient first in healthcare.

We want to create a difference within the lives of both patients and healthcare providers. By making a platform that improves patient-healthcare providers communication, ensure exact severity assessments and advances a more consistent healthcare experience.

Problem domain analysis

The problem domain of the app revolves around addressing the challenges of post-procedure care that negatively impact the patients. These challenges include a lack of immediate access to healthcare providers which can cause anxiety for the patient regarding their health status. Delayed symptom reporting in emergent situations could potentially lead to worse conditions. Additionally, patients may forget their medication schedule, which can result in missed doses or inconsistencies in their treatment plan impacting their health outcomes.

Solution Description

Medup is designed to enhance the post-procedural care process by enabling patients to track their symptoms with ease and help them manage their medications. It consists of the following parts:

1- Symptom Entry

Patients can enter their symptoms through an AI chatbot. Additionally, patients can upload images of visible symptoms such as surgical sites to ensure a detailed symptom report.

2- Symptom Assessment

Medup employs an AI chatbot and an Advanced wound imaging API to assess symptom severity based on the symptom report. It utilizes Natural Language Processing (NLP) for text inputs to analyze the symptoms description, and image processing through the Advanced wound imaging API to analyze the uploaded images to identify infections and other complications.

3- Communication with Healthcare Providers

Medup allows patients to communicate with healthcare providers if a symptom is severe via an API for messaging or calling for fast intervention. The app will also provide secure sharing of the symptom report with healthcare providers.

4- Medication management

Healthcare providers or patients can enter the medication schedule into the app, and the app will send them reminders to ensure they follow the treatment plan. The app will also provide information about each medication such as dose administration, drug interactions, and possible side effects.

System Glossary

Abbreviation	Description
Natural Language Processing (NLP)	"refers to the branch of computer science—and more specifically, the branch of artificial intelligence or AI—concerned with giving computers the ability to understand text and spoken words in much the same way human beings can" [1]
Image Processing	"the use of algorithms and mathematical models to process and analyze digital images. The goal of digital image processing is to enhance the quality of images, extract meaningful information from images, and automate image- based tasks" [2]
AI	"Artificial intelligence refers to the ability of technology, particularly computer systems, to simulate human intelligence processes. Expert systems, machine learning, speech recognition, and natural language processing are a few specific uses of AI" [4]
MVC	Model-View-Controller Architecture.

System Context View

(Scenario #1)

After surgery monitoring

A patient just had open heart surgery, the surgery went well, and the patient is feeling good so he/she asked to be released early, the healthcare provider agreed since the patient's condition was stable, and there was no significant danger on their life according to their postoperative symptoms in the first few days, yet the healthcare provider wants to monitor the patient case just to make sure that they're safe and healthy, the healthcare provider enters the patient's medical records into the system then the healthcare provider asks the patient to record their postoperative symptoms at least once a day for two weeks. The system well help both the healthcare provider and the patient. The system will decide if the postoperative symptoms are dangerous or normal. When the patient records some postoperative symptoms that could be dangerous, the system will immediately notify the healthcare provider and tells the patient to go to the hospital immediately. This will save time for both the healthcare provider and the patient; the healthcare provider won't waste their time and the hospital beds on people who are not in need. The patient is saved from significant financial overhead by avoiding hospital monitoring.

(Scenario #2)

Wound image processing

A user of the system had a surgery three days ago, unfortunately the wound got infected, but the user is not sure if this look and pain is normal or not, so the user decides to use the system first so he/she can see if they need to go to the hospital or not. The users asked to assign the pain level and a photo of the wound, the system then processes this photo and decides whether it looks normal or not. The wound seems to be infected. The system will ask the user if they can go to the hospital or need to order an ambulance. The user's life is saved by knowing that there was an infection early before it spreads and become harder to manage which may lead to bigger problems. If the user had a normal looking wound and they knew that without going to the hospital, the hospitals will have fewer patients and the healthcare provider can focus on patients who are in need of help.

Requirements

Functional Requirements

1.1 The patient shall be able to create an account by specifying (Name, phone number, national ID and password) .

1.2 The patient shall be able to login by entering the national ID and the password.

1.3 The patient shall be able to enter his/her medication schedule within the app.

1.4 The patient shall be able to delete his/her medication schedule.

1.5 The patient shall be able to book appointments with his/her healthcare provider immediately from the app.

1.6 The patient shall be able to chat the AI health assistant to discuss his/her symptoms.

1.7 The patient shall be able to capture and upload images of his/her wound to AI image processing API.

1.8 The patient shall be able to view previous symptoms assessments.

1.9 The patient shall be able to view their medical records.

1.10 The system shall facilitate real-time communication between patient and healthcare provider (chat or call).

1.11 The system shall provide medication reminders based on the patient medication schedule.

1.12 The system shall display information about each medication including dosage, instructions and potential side effects.

1.13 The system shall send appointment reminders to the patient via SMS.

1.14 The system shall ask clarifying questions to gather additional information about the symptoms.

1.15 The system shall provide clear and concise information about the assessed severity of each symptom.

1.16 The system shall offer appropriate self-care advice and educational resources for low-severity symptoms.

1.17 The system shall prompt the patient with high-severity symptoms to contact his/her healthcare provider.

1.18 The system shall provide feedback to the patient about the AI assessment of their wound image.

1.19 The system shall transmit symptom information and wound images to the healthcare provider.

1.20 The system shall notify the healthcare provider about any high-severity symptoms.

1.21 The system shall notify the healthcare provider about any infected wound images identified by the AI image processing API.

1.22 The healthcare provider shall be able to create an account by specifying (Name, phone number, national ID and password).

1.23 The healthcare provider shall be able to login by entering the national ID and the password.

1.24 The healthcare provider shall be able to view patient symptoms assessments.

1.25 The healthcare provider shall be able to add new patients.

1.26 The healthcare provider shall be able to link new patients to their medical records.

1.27 The healthcare provider shall be able to view the patient record.

1.28 The healthcare provider shall be able to view patient appointment requests.

Non-Functional Requirements

2.1. Scalability| The system shall be able to accommodate 60,000 concurrent users.

2.2. Performance| The system response time shall be less than 3 seconds.

2.3. Usability| The new users shall be able to learn how to use the system in less than 5 minutes.

2.4. Reliability| The system shall be available 99% of the time.

2.5. Flexibility| The system shall take no more than two weeks to add a new functionality.

2.6. Testability| All software components shall complete testing in one month.

2.7. Portability| The system developed for IOS should be compatible with android environment.

2.8. Security | The system shall be protected against unauthorized access and data breaches.

Challenges

It is only natural for any project to face many challenges which can cause major issues in the software, since we will be using an API and machine learning module, we expect them to be our main challenge. We may find challenges in integrating them with our system, ensuring they are compatible and upgradable within our systems, assuring that the AI is continuously updated with the latest medical knowledge and have high accuracy even with the ambiguity of the user's input, following regulations and standards of Healthcare Systems and insuring accuracy and reliability on both the API and the machine learning module. Additionally, another problem we may face is getting users trust and reliance since users may be skeptical of relying solely on an AI system for medical diagnose of their symptoms, especially in critical or com plex cases.

Projection

The objective of this project is to gain the knowledge and skills necessary to select the most suitable software architecture for our system, taking into account our priorities. By the end of this project, our aim is to develop a system that effectively fulfills the specified functionality requirements while maintaining a high level of quality.

What will be accomplished by the software?

- Benefiting healthcare providers by saving time and effort.
- Patient getting efficient and rapid response for their concerns.
- Managing patients appointments and medications
- Easy and efficient access to the healthcare providers for patients

Skills and Knowledge to be obtained

- Defining and analyzing problem domains.
- Enhance design, modeling, validating and testing skills.
- Utilizing software architecture fundamentals to avoid and solve software problems.
- Emphasizing collaboration among teammates hence enhancing interpersonal skills.

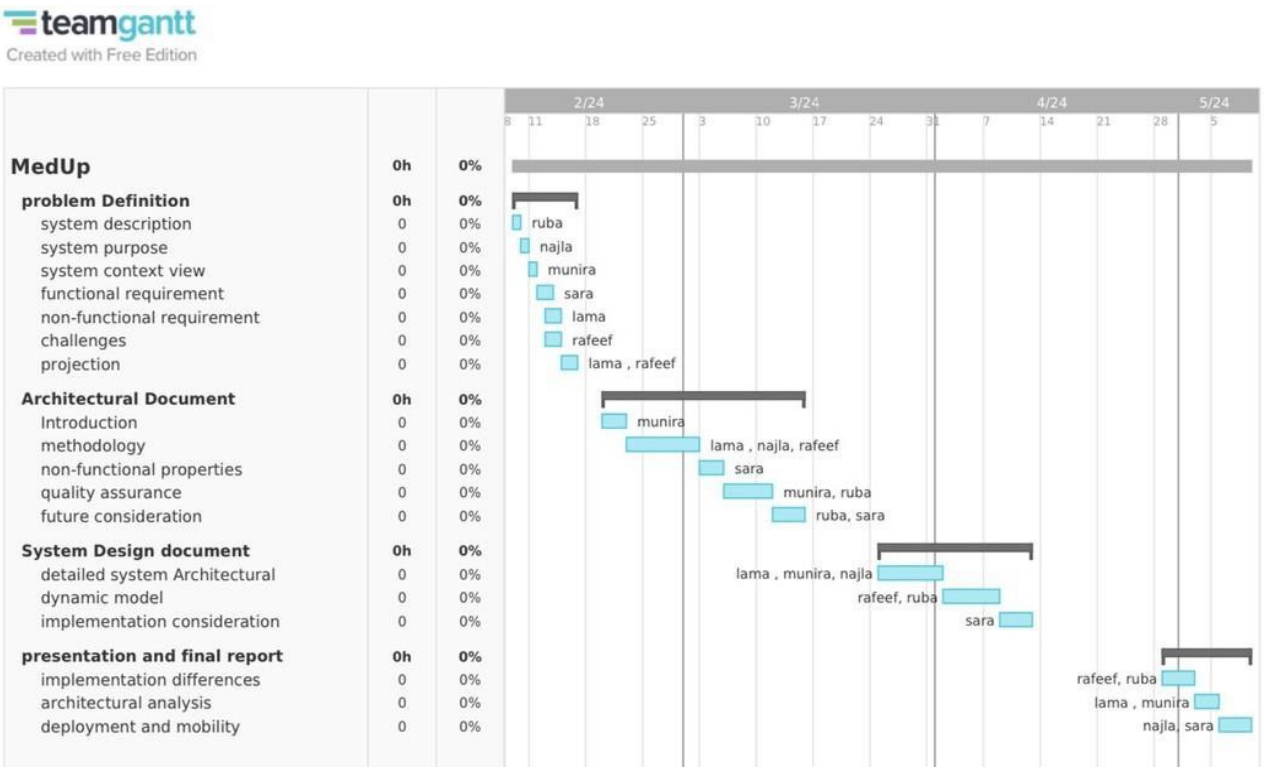

Figure 1: this figure shows Gantt chart execution plan for the medup system

The mentioned chart simplifies the project management process by outlining timelines, tasks with their start and end dates, dependencies between activities, scheduling, and deadlines. This aids in effectively monitoring and managing the project development.

Introduction

MedUp is a healthcare platform designed to streamline and enhance post-procedure monitoring for both patients and healthcare providers. Many patients struggle in effectively communicating their symptoms after a medical procedure, while healthcare providers struggle to manage the increasing volume of patient information in the healthcare industry.

MedUp utilizes AI for assessing the severity of symptoms and wound image processing, providing patients with a user-friendly interface to assess their symptoms after a procedure. The platform alerts patients to high-severity symptoms and allows interaction with healthcare providers through real-time chat or call features. For healthcare providers, MedUp offers a centralized system to track patient symptoms, provide clear feedback on image assessments, and receive real-time notifications for high severity cases.

In addition, the platform simplifies appointment scheduling, offers medication reminders, and provides comprehensive information on prescribed medications. MedUp promotes a patient-centered approach that prioritizes patient information and aims to improve the overall healthcare experience.

Methodology

We have determined that Agile is the most effective methodology for developing MedUp after carefully examining its requirements, the project, and the needs of the users. Agile guarantees that the development process is always improving while offering the flexibility needed to swiftly adjust to shifting needs of customers, which eventually improves the quality of the software.

We suggest dividing the project into several iterations, each focused on delivering certain features, in order to successfully apply Agile for MedUp. At first, the emphasis will be on creating core elements such as patient and healthcare provider profiles, login and logout, and registration. More complex features including communication, medication management, and appointment scheduling will be included in later versions. Subsequent versions will entail integrating services such as Agora, Yakeen, and SMS gateway, leaving the riskiest part for the last stage which is the AI chatbot. By prioritizing core aspects before going on to more advanced functionality, this staged strategy assures progressive development.

Table 1: This table shows distribution of tasks upon team members with the time duration.

iteration number	Module/Subsystem	Duration
1	Registration, Login, Logout, Patient and Healthcare provider profiles.	2 weeks
2	Appointment scheduling, Medication management and Communication.	1 month
3	Agora, yakeen and SMS gateway integration.	1 month
4	AI Chatbot	3 months

The Gantt chart below outlines the estimation plan for Medup development, which is based on the Agile Methodology. The quality assurance section includes the testing plan for MedUp.

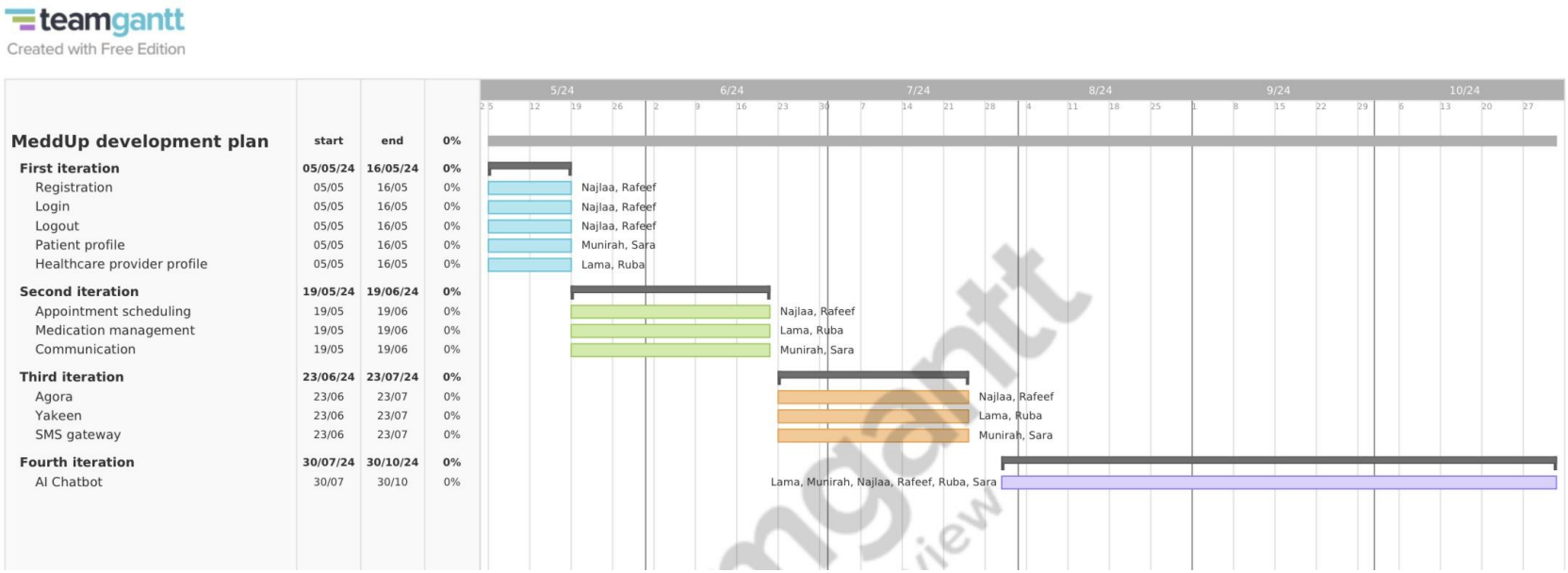

Figure 1: detailed plan for MedUp development using Gantt chart.

In the MedUp system as discussed in the challenges section implementing an AI chatbot is difficult due to its nature and the extensive training and testing required to validate its accuracy and suitability. To address this issue, we have scheduled it as the final stage allowing our team to gain knowledge and possibly seek guidance from experienced individuals to ensure its successful implementation, with some level of confidence.

Risk	Mitigation strategy
AI Chatbot reliability and accuracy	extensive training and testing using real-world data, consulting with experienced professionals.
Dependency on external API “EKare”	Create backup plans in case the API stops working or goes down. Additionally, we might think about using other APIs or even develop our own internal solutions. Perform routine maintenance and monitoring to guarantee that functionality continues.

System Architecture

A. Design Decisions:

Design Question 1: How will the Medup system authenticate users while logging in?

We have opted to use OTP (One-Time password) authentication to enhance the security of the MedUp system during the log in procedure. This decision is crucial for safeguarding users accounts and restricting unauthorized access. To verify ownership of the registered phone number, the system will prompt the users to input the correct OTP during the log in procedure. We will also be using newer technology later on the project such as OAuth(open authorization) for users to be able to sign in via a third-party services such as google or Nafath(نفاذ) and WebAuthn API for users to sign in with the any biometrics depending on their devices such as face-id, fingerprint or device pin without ever having to have a password for our system only the users ID.These password less methods are the most secure way to prevent against most security threats and unauthorized access.

Design Question 2: How will MedUp store users' data?

MedUp system will use Google Cloud Platform(GCP), as it works well with machine learning models. GCP also offers great data privacy and security features which are important for patients' sensitive information. In addition, GCP is known for having high-performance computing and fast response time, which is important for MedUp as it deals with patients who could be in a critical condition. We will be using two databases one contains all users' information, and the other is only specialized in the MLM with the blackboard.[6]

Design Question 3: How will we ensure the security of our user's data?

With secure coding practices like validating and sanitizing users input in the chatbot or any input field, we will be able to prevent cross site scripting(XSS) and injection attacks on our application and server, preventing injections to external databases. Additionally, when developing our machine learning module and integrating our API, we will be using secure HTTPS or SSL/TLS for data transfer to the external API protecting the user's data

Design Question 4: How will we deal with the challenges of training the machine learning model?

To overcome these challenges, it is essential that the training data is curated and preprocessed carefully. This will involve cleaning the data to remove any errors or inconsistencies as well as handling missing information plus ensuring that the data reflects the problem at hand. Feature engineering also plays a major role because it involves picking out and altering those attributes in each dataset which matter most for purposes of improving model's performance. In addition, choosing the right algorithm and hyperparameters along with regularization processes of models to prevent overfitting are crucial steps towards solving machine learning training issues.

Design Question 5: How can the user trust the information that our system provides?

The only source of information that will be used is research papers that are verified by global health institutions, and medical books, therefore we chose blackboard architecture since it is the most suitable for this form of
information, this ensures that the information provided is reliable and up to date. Additionally, assessments will be always monitored by healthcare providers.

Design Question 6: How will the MedUp system improve the accuracy of symptom severity assessment?

The MedUp system will use **eKARE** advanced image processing for comprehensive wound imaging capabilities to enhance the accuracy of symptom severity assessment. Through this strategic incorporation of image processing, we enable MedUp to gather and analyze high-quality visual data, for post-procedural symptoms. MedUp guarantees a more precise evaluation to create informed decisions based on precise data, contributing to a more successful post-procedural care**.**

B. Domain model:

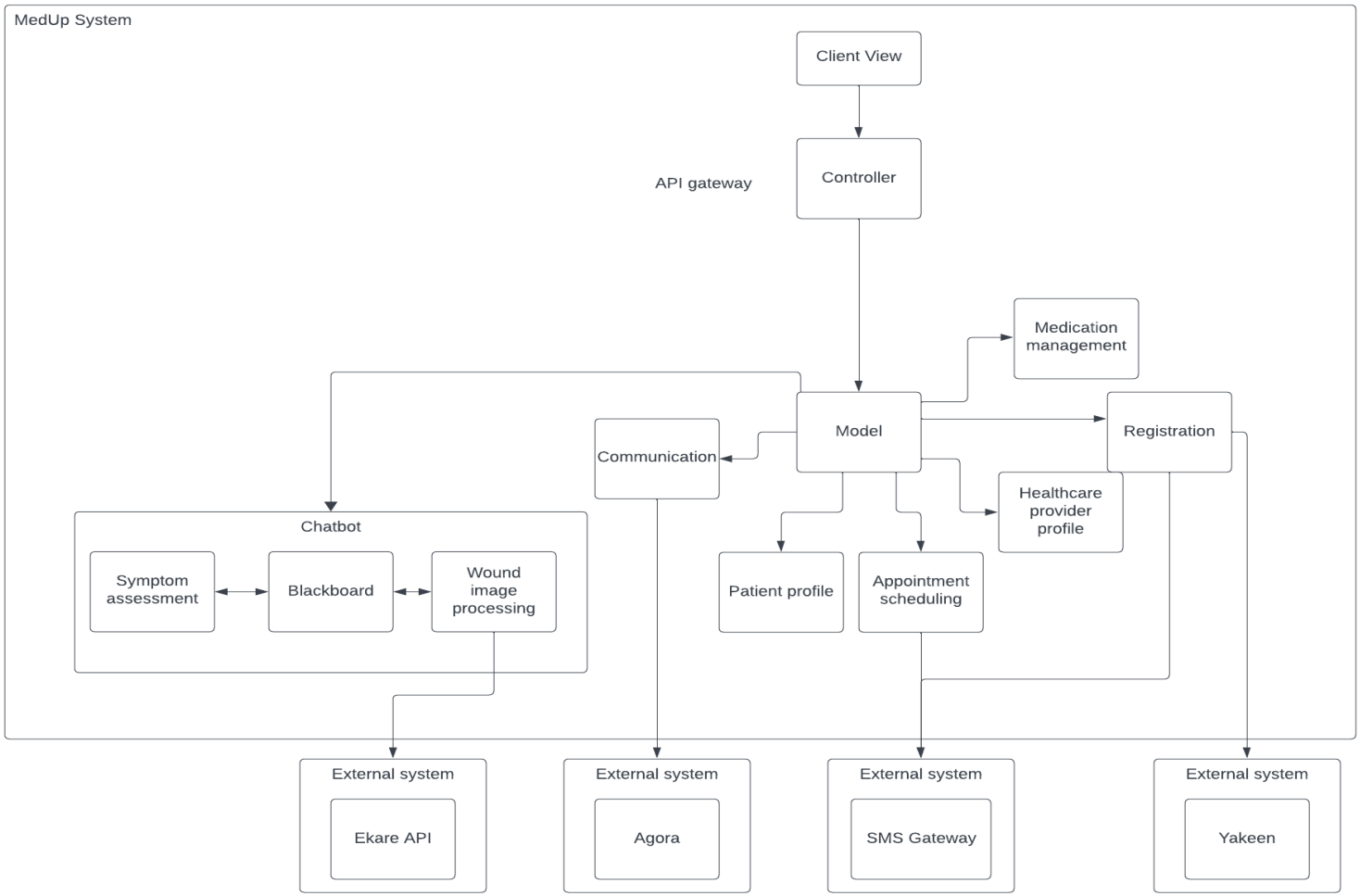

Figure 2:Domain Model for MedUp explaining the system architectural style.

MedUp Components:

Client: a registered user who has access to system functionalities based on their role as a healthcare provider or a patient.
Controller: represents the interface of all services put together.
Patient profile: Patient profile is used to manage comprehensive information about patients' accounts, representing the patient profile picture, name, records and app settings.
Healthcare provider profile: healthcare profile is used to manage comprehensive information about healthcare provider account, representing experience and education in addition to their profile picture, name and app settings.
Registration service: a component responsible for registration/ signing in patients and healthcare providers.
Medication management service: a service that notifies the patient when they need to take their medication and keeps track of their medications.
Communication service: a service that enables the patient to communicate with the health care provider via agora interface.
Agora: Agora is a platform used for real-time voice, video, and messaging features. It enables seamless communication and collaboration across various platforms and devices.[7]
Appointment scheduling service: a service that helps the patient to schedule appointments with a healthcare provider.
Wound image processing: sends the wound image that is provided by the patient through eKare API, gives the patient a description of the severity of their wounds.
Ekare API: wound image processing external system used to analyze and manage images of wounds.
Blackboard: will provide the knowledge source for connecting the Wound Image Processor and Symptom Assessment components, allowing data exchange.
Symptom assessment: a service that enables the patient to enter their symptoms and track them.
Yakeen: Electronic verification that used to send confirmation codes to the recipients' cell phone numbers enrolled through Absher at the National Information Center by connecting it directly to government systems databases following the most reliable security and protection measures worldwide.[8].
SMS Gateway: using the registered phone number to send reminders about upcoming appointments.

C. Architectural Style:

Choosing the most suitable architectural style for our system is crucial for a robust system, hence we decided to combine two architectural styles to get the most benefits out of each, since we are building a system that contains unique components with distinct relationships among them. For our system, we adopted Model-View- Controller (MVC) as our main style along with Blackboard.

MVC provides a structured way to organize and separate the overall system components into three main components: Model, View, and Controller. Dividing these components allows each component to have an independent development which enhances modularity, scalability, and reusability. The Model component represents the data and logic part of our system where our main services are going to be along with databases, also in the model we are going to be taking users input and managing data along with the heavy computations like the machine learning module, we are using in our AI assistant chatbot. Secondly, the second part of the MVC is the view which is the client side and responsible of displaying the data and services of the module in a simpler user-friendly interfaces and takes the users input and forward it to the last component of this structure which is the controller, it acts as an intermediate between the view and the model that is responsible for processing users input and update the model as well as updating the view based on the model changes.

Furthermore, within MedUp we have the AI assistant chatbot as our main service that contains the MLM(symptoms assessment) along with wound image processing API that acts as a functionality within the chatbot so this created a problem in which there is decreased modularity since we are using an external API which can backfire on our system in the future, to solve this, we decided to integrated Blackboard architectural style in which it enables collaboration among independent components and allows the integration of diverse sources and technologies hence promoting modularity and extensibility

Blackboard is an architecture that is commonly used in artificial intelligence systems, it takes the collection of independent components and allows them to share information in a common “blackboard”. Components interact with the blackboard and update the information stored there iteratively combining knowledge converging to a better outcome in computations without the components interfering among each other which is perfect for our AI assistant chatbot functionalities.

Initially, Layered architecture style was our first consideration. Though for various reasons we shifted away from it. Our main reason is because layered architecture promotes tight coupling with the many layers. Those many layers would greatly decrease the performance as our system would have to traverse through many back and forth to pass data and get a response, which eventually will cause a great overhead to our system.

High level Diagram:

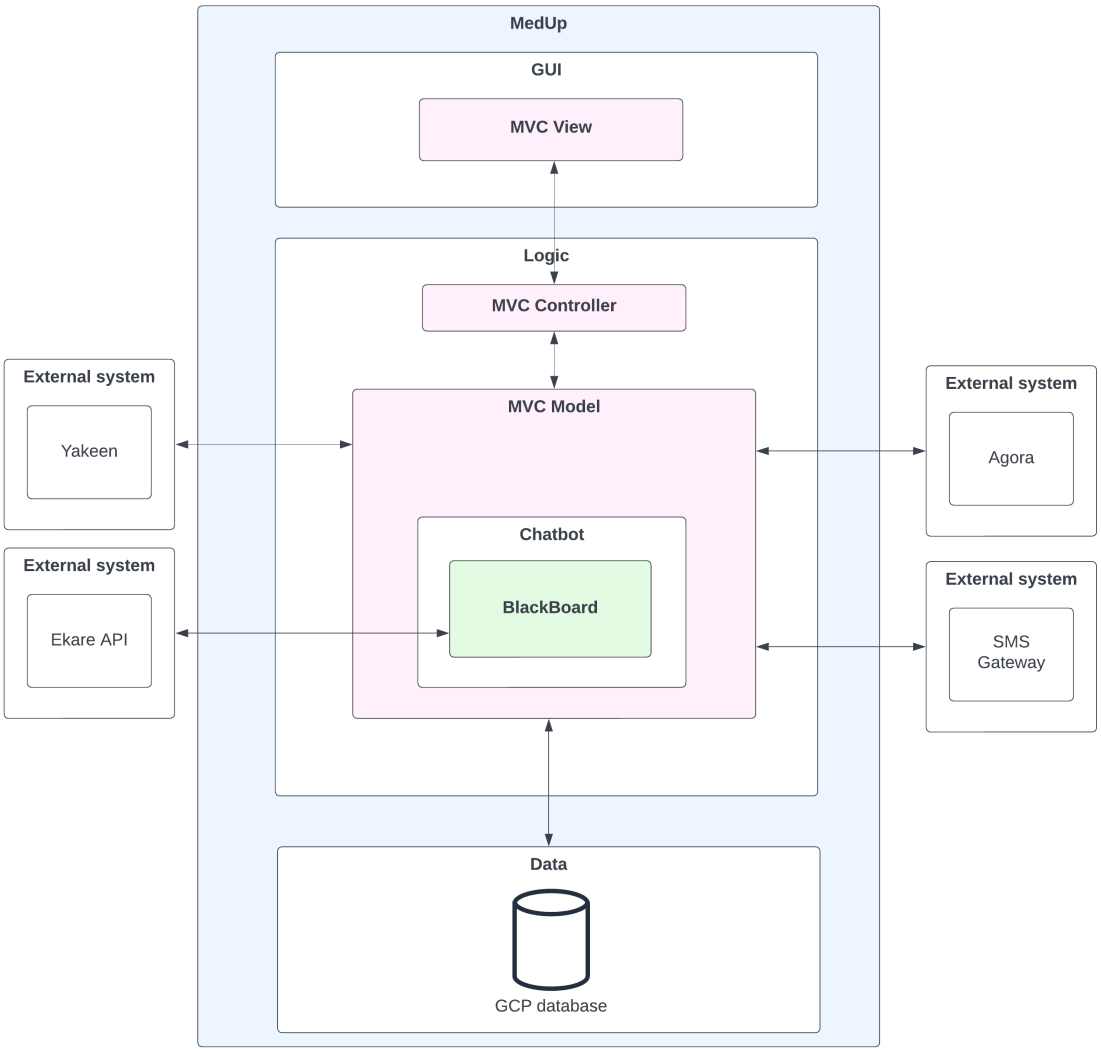

Figure 3:High level diagram for MedUp system

D. Structural Model

- **Component Diagram:**

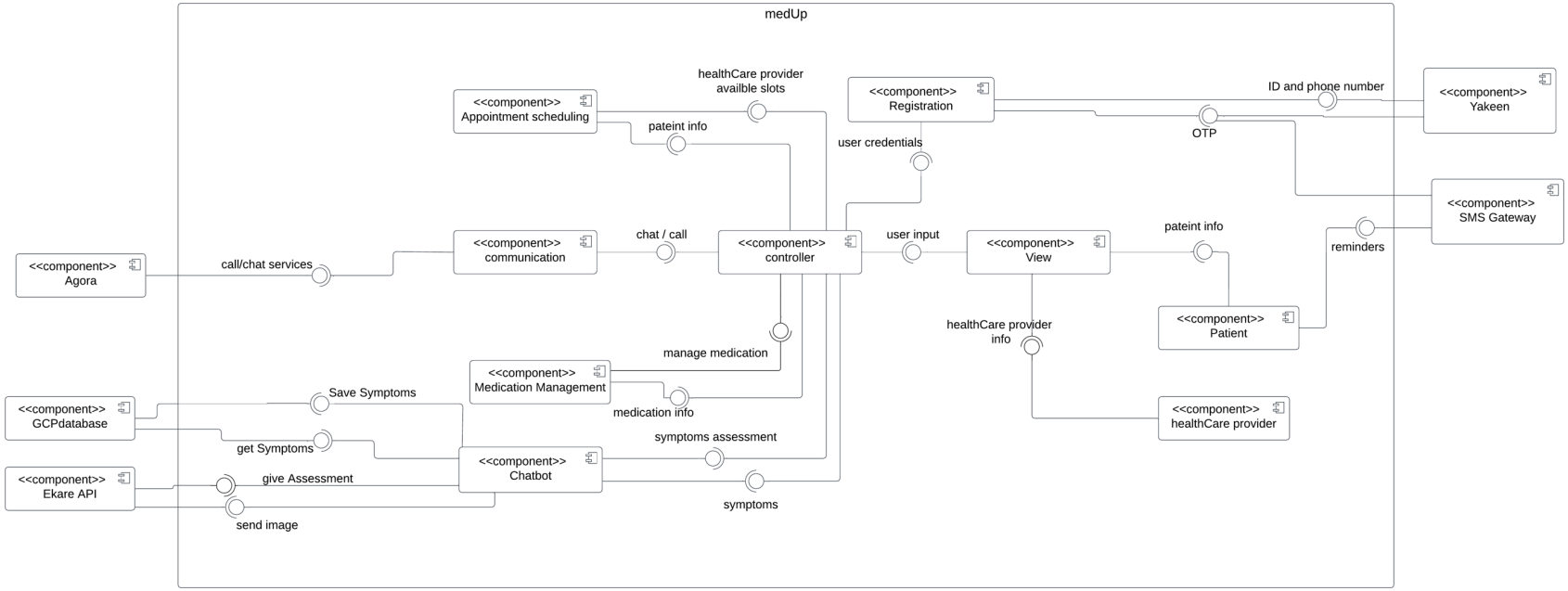

Figure 4:Component diagram for MedUp system.

Component Name	Patient
Description	This component consists of patient profile and functions.
Behavior/Functionality	Schedule Appointments, Manage Medication, Enter symptoms and communicate with healthcare provider.
Connectors and Interfaces	Provided interfaces: manage medications, enter symptoms and provide patient information. Required interfaces: remainders, medication information, symptoms assessment and chat/call.

Component Name	Healthcare provider
Description	This component consists of healthcare provider profile and functions.
Behavior/Functionality	Checks patients' information, communicates with patients and provides available slots for scheduling appointments.
Connectors and Interfaces	provided interfaces: healthcare provider information and appointments available slots. Required interfaces: chat/call.

Component Name	Registration
Description	A component that provides signup and login functions.
Behavior/Functionality	Allows users to login, register, and save their data securely.
Connectors and Interfaces	provided interface: phone number. Required interfaces: healthcare provider information, patient information and OTP.

Component Name	Appointment Scheduling
Description	A service that helps patients to schedule appointments with healthcare provider.
Behavior/Functionality	checks healthcare provider available slots and enables patients to schedule appointments with healthcare provider.
Connectors and Interfaces	Required interfaces: patient information and available slots.

Component Name	Communication
Description	A service that enables the patient to communicate with the health care provider via agora interface.
Behavior/Functionality	provides chat and call services.
Connectors and Interfaces	Provided interface: chat and call. Required interfaces: chat/call services.

Component Name	Medication Management
Description	A service that notifies the patient when they need to take their medication and keeps track of their medications.
Behavior/Functionality	Notifies the patient about their medication and enables the patient to manage their medications.
Connectors and Interfaces	Provided interface: medication information. Required interfaces: manage medications.

Component Name	SMS gateway
Description	An external component that sends reminders and OTP to patients.
Behavior/Functionality	Send appointment reminders.
Connectors and Interfaces	Provided interface: reminders.

Component Name	yakeen
Description	An external component that helps with registration and login functionalities.
Behavior/Functionality	Sends a message containing OTP to Authenticate users.
Connectors and Interfaces	Provided interface: OTP. Required interface: ID and phone number.

Component Name	GCPdatabase
Description	A component that is used to store and retrieve Symptoms.
Behavior/Functionality	Saves symptoms.
Connectors and Interfaces	Provided interface: get symptoms. Required interface: save symptoms.

Component Name	Chatbot
Description	A chatbot is a computer program that simulates conversation with human end users.[9]
Behavior/Functionality	Assess symptoms.
Connectors and Interfaces	Provided interfaces: symptoms assessment, save symptom and send image. Required interfaces: symptoms, get symptoms and give assessment.

Component Name	Agora
Description	An external component used for real-time voice, video, and messaging features.
Behavior/Functionality	Provides communication services.
Connectors and Interfaces	Provided interface: call and chat services.

Component Name	EkareAPI
Description	wound image processing external system used to analyze and manage images of wounds.
Behavior/Functionality	analyze wound images.
Connectors and Interfaces	Provided interface: give assessment. Required interface: send image.

- **Deployment diagram**

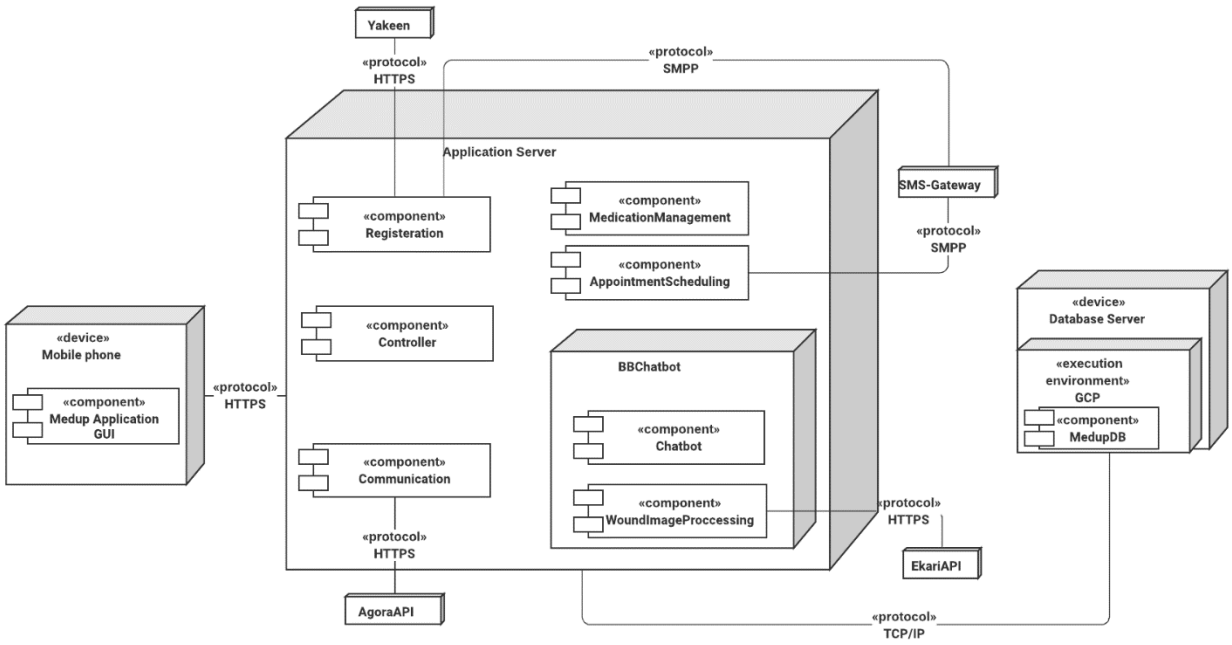

Figure 5: Deployment Diagram for MedUp system.

• Class diagram:

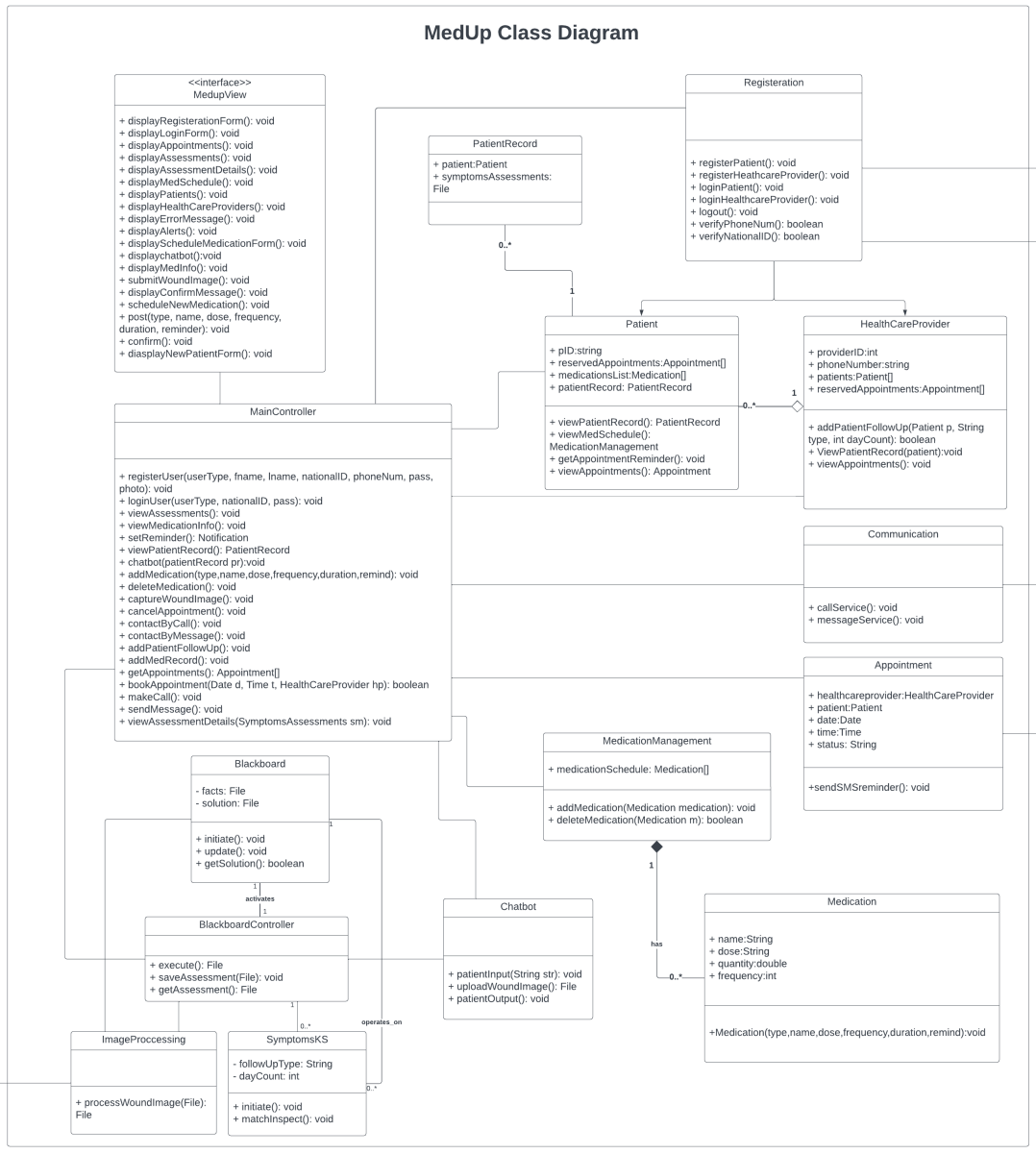

Figure 6:Class diagram for the MedUp system.

- **Class component mapping:**

Component name	associated classes names
View	MedUpView
Controller	MainController
Patient	Patient patientRecords
healthcareProvider	healthcareProvider
Registration	registration
Chatbot	Blackboard blackboardController chatbot imageProcessing symptomsKS
appointment scheduling	Appointment
medication management	medicationManagement Medication
communication	communication

E. Behavioral Model:

- ## Use Case Diagram:

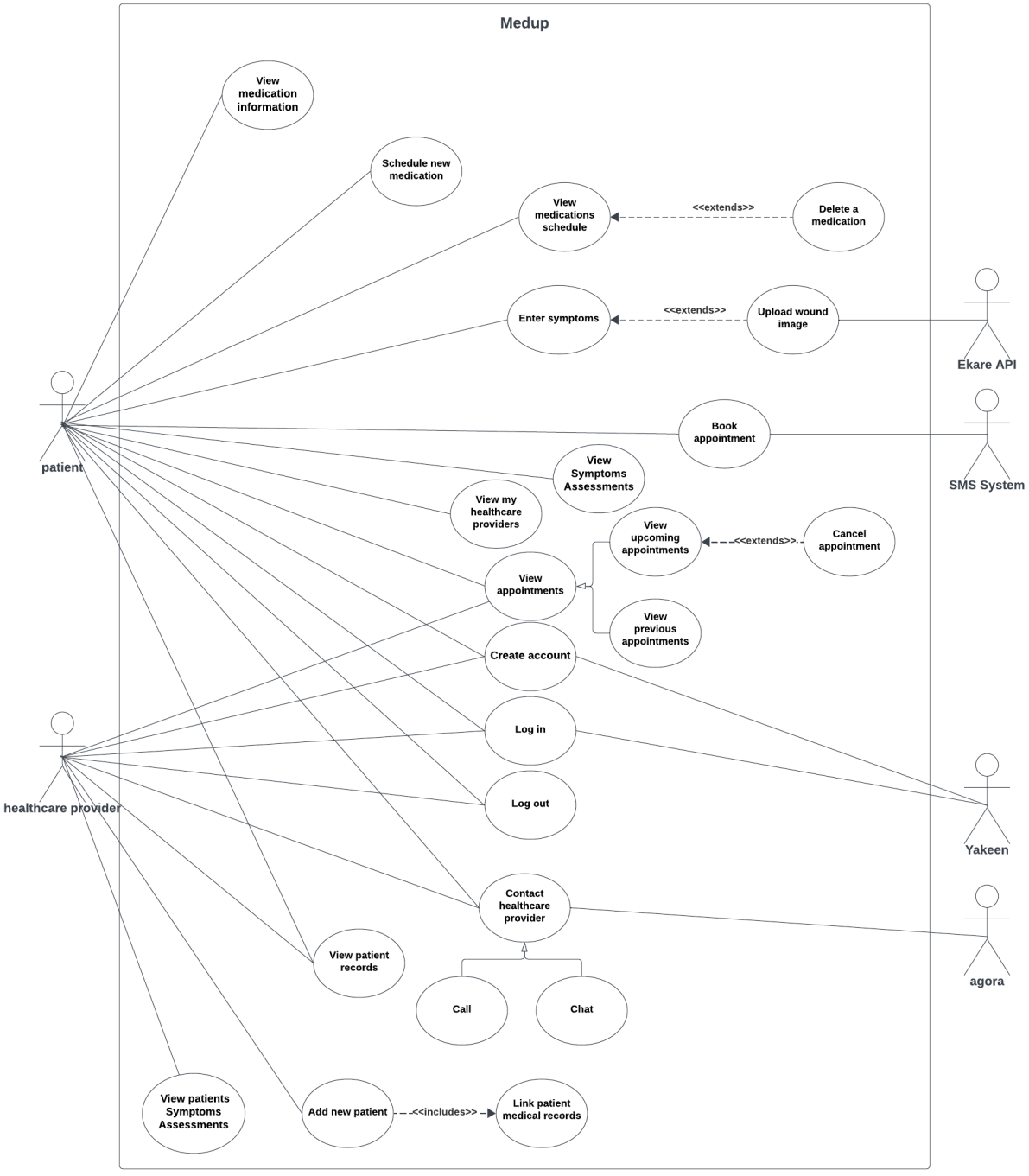

Figure 7:Use case diagram for MedUp system.

- **Use Case Description:**

Use Case Description	
System: MedUp	
Use Case Name: **Enter Symptoms**	
Primary actor(s): Patient	Secondary actor(s): -
Description: this use case describes how patients can record their symptoms	
Relationships Includes: none Extends: upload wound images.	
Pre-conditions: The patient must be logged into the system.	
Basic flow:	
Patient	System
1-Patient selects the "Assess Symptoms" button after entering symptoms. 3-Patient provides details about their symptoms.	2-The system asks the patient clarifying questions. 4-The system assesses the symptom severity and provides clear information about the assessed severity of each symptom. 5-The system provides appropriate self-care advice and educational resources. 6- The system securely transmits symptom information to the healthcare provider for monitoring.
Alternative and exceptional flows: **High-severity symptoms** If in Step 4 high-severity symptoms are detected, then: 4.1. The system prompts the patient to contact the healthcare provider. 4.2. The system notifies the healthcare provider about the high severity assessment.	
Post-conditions: Successful condition: The symptoms are successfully assessed . Failure condition: The user gets an unsuccessful symptom assessment.	

Figure 8:A use case description for Enter Symptoms use case.

Use Case Description		
System: MedUp		
Use Case Name: **upload wound images**		
Primary actor(s): Patient	Secondary actor(s): eKare	
Description: this use case describes how patients can check their wound by uploading an image of the wound.		
Relationships Includes: none Extends: none		
Pre-conditions: 1- The patient must be logged into the system. 2-The patient must be actively interacting with the chatbot.		
Basic flow:		
Patient	System	Ekare
1-Patient selects the "upload wound images" option. 3-The Patient uploads an image of their wound.	2- The system asks the patient to capture and upload an image of their wound. 4-the system sends the image to the eKare for processing. 7-the system displays a message that the wound appears normal.	5- eKare analyzes the uploaded image. 6-eKare sends confirmation indicating that no infection is detected to the system.
Alternative and exceptional flows: **Infected wound:** If in Step 5 The wound seems to be infected. , then: 6.1- The system prompts the patient to contact the healthcare provider.		
Post-conditions: Successful condition: The patient is informed about the wound's status. Failure condition:The patient is not informed about the status of the wound.		

Figure 9:A use case description for upload wound images use case.

Use Case Description	
System: MedUp	
Use Case Name: **Enter Medication Schedule**	
Primary actor(s): Patient	Secondary actor(s): -
Description: this use case describes how patients can Enter their Medication Schedule	
Relationships Includes: none Extends: none	
Pre-conditions: The patient must be logged into the system .	
Basic flow:	
Patient	System
1-Patient selects the “Schedule new Medication ” option. 3- The Patient enters the required information. 4- The Patient selects “Confirm”	2- The system asks the Patient to enter their medication schedule (medication type , medication name , dose, frequency, duration, reminders). 5- The System displays a message that the Medication Schedule was added successfully
Alternative and exceptional flows: **Edit the Medication Schedule:** If in Step 4 The Patient selects “Cancel”. , then: 4.1 The system takes the user back to the home page .	
Post-conditions: Successful condition: The medication schedule is successfully added to the system. Failure condition: The medication schedule fails to be added to the system.	

Figure 10: A use case description for Enter Medication Schedule use case.

- **Sequence Diagram:**

Sequence diagram for schedule new medication use case:

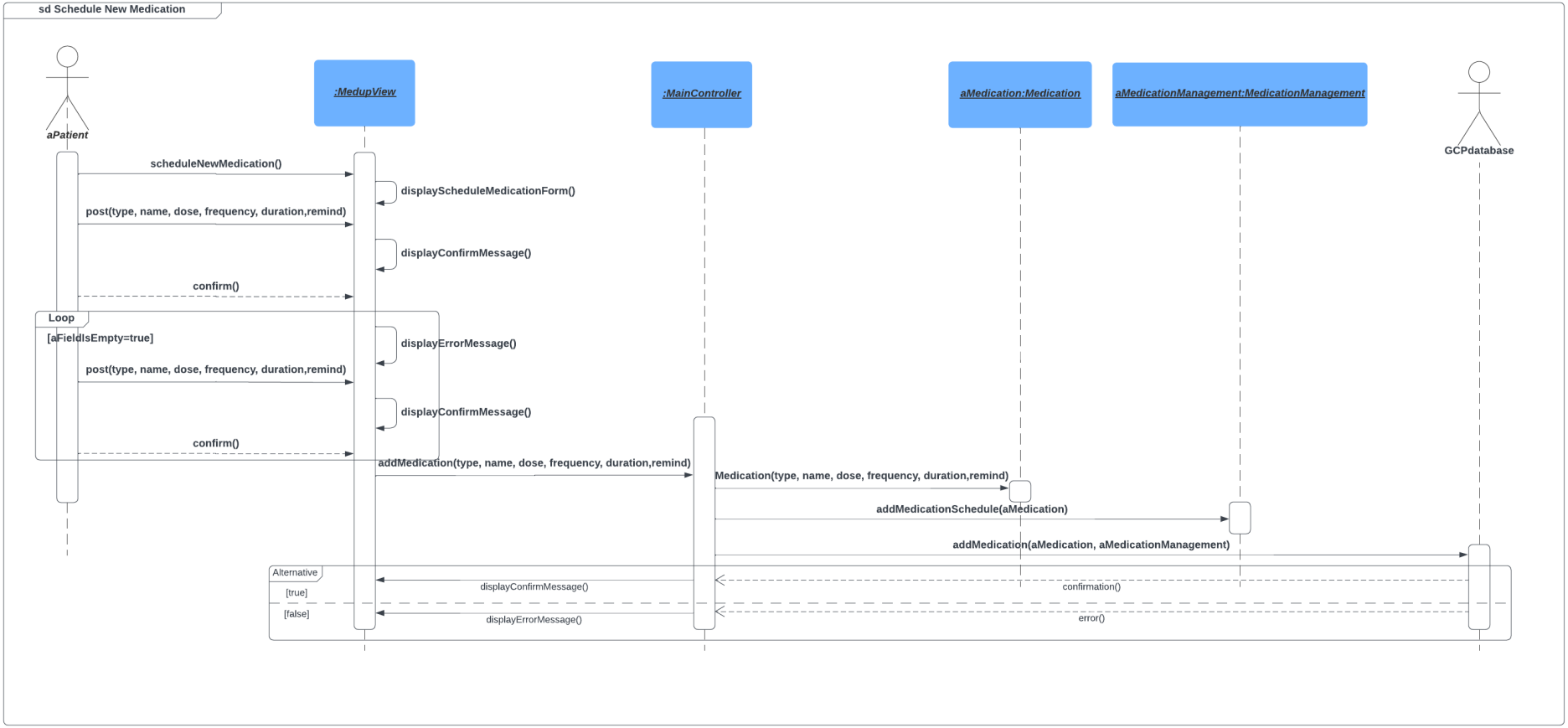

Figure 11:Sequence diagram for "schedule new medication" use case in MedUp system.

Sequence diagram for Enter Symptoms use case:

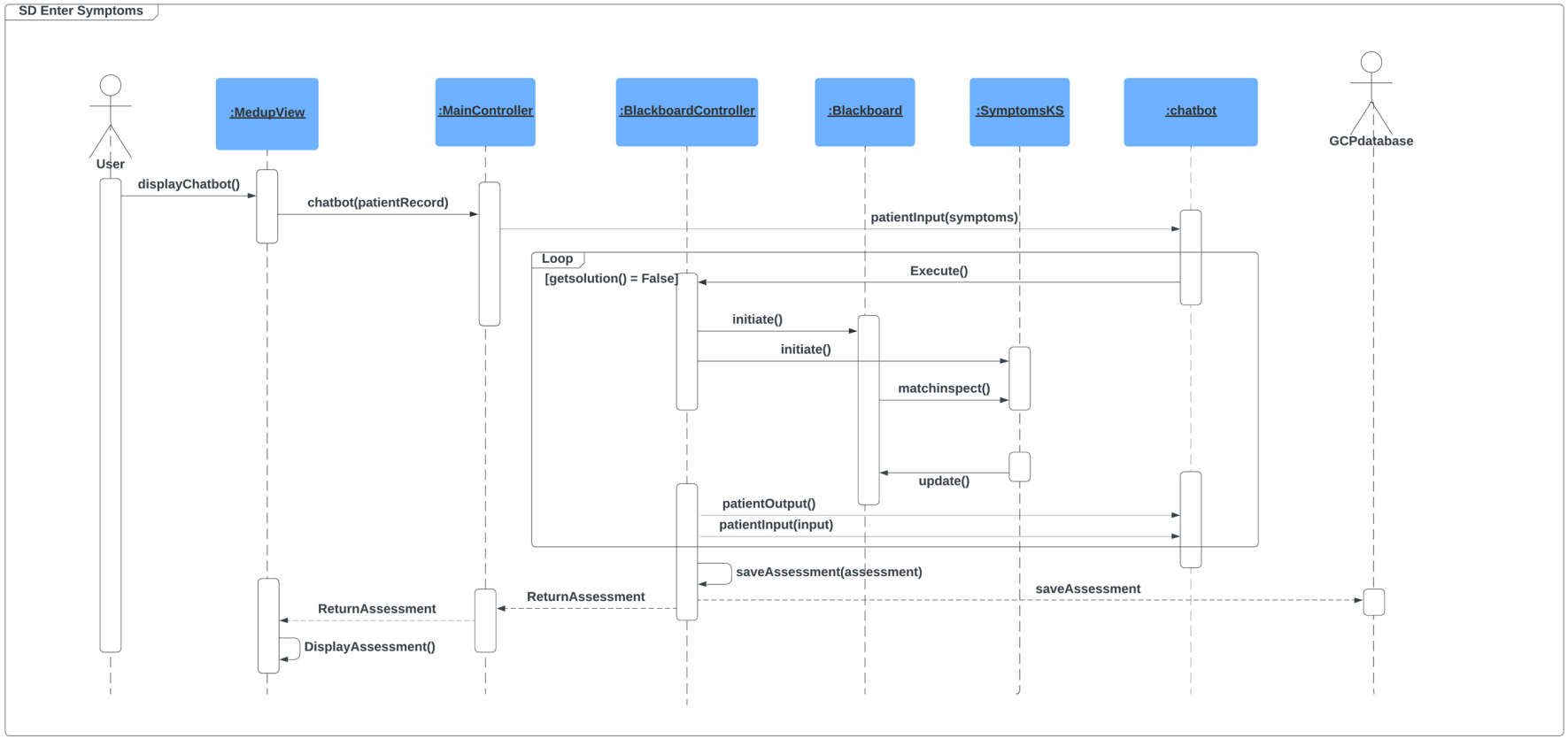

Figure 12:Sequence diagram for "Enter symptoms" use case in MedUp system

Sequence diagram for image processing use case:

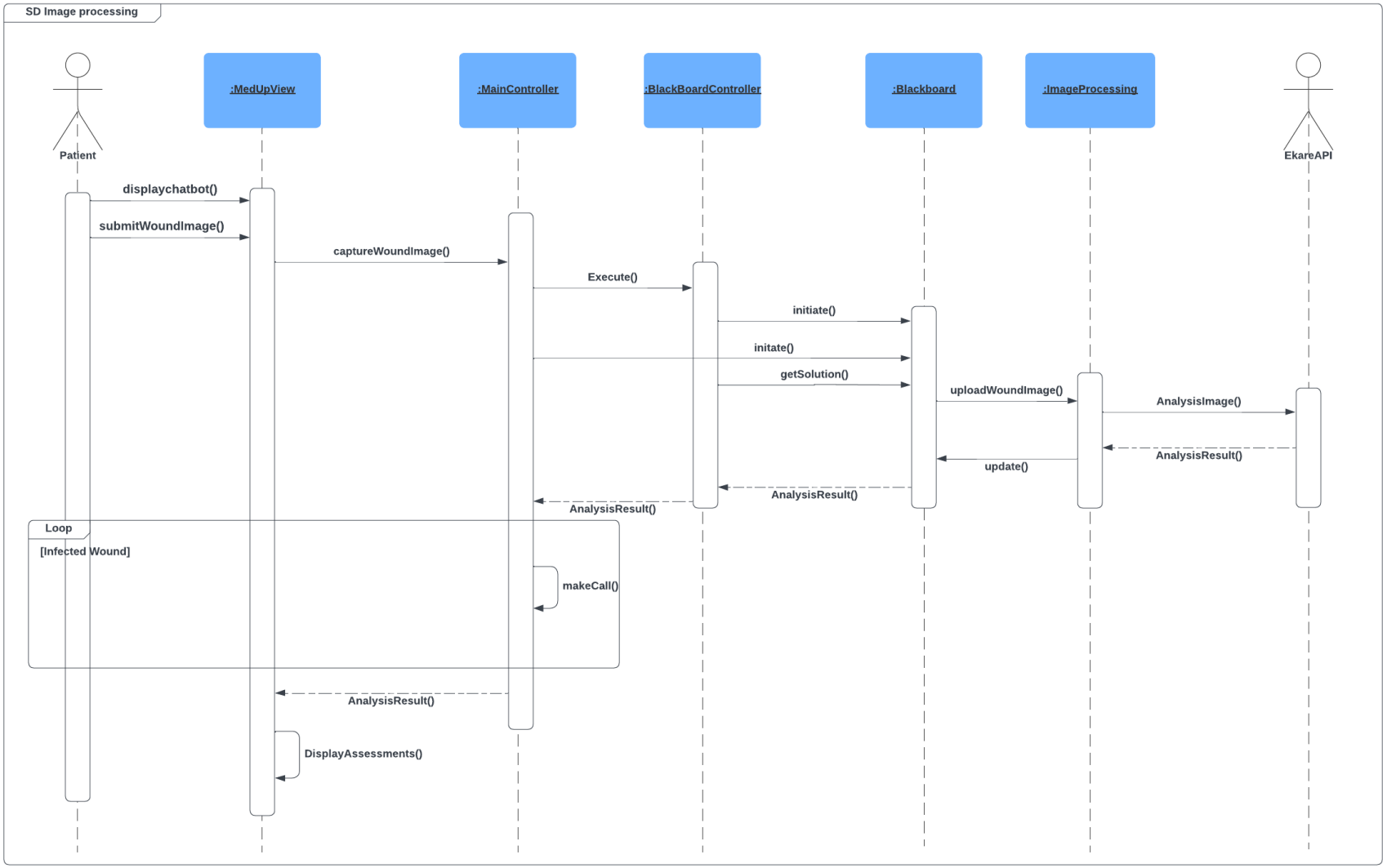

Figure 13:Sequence diagram for "Upload wound images" use case in MedUp system.

- **State diagram:**

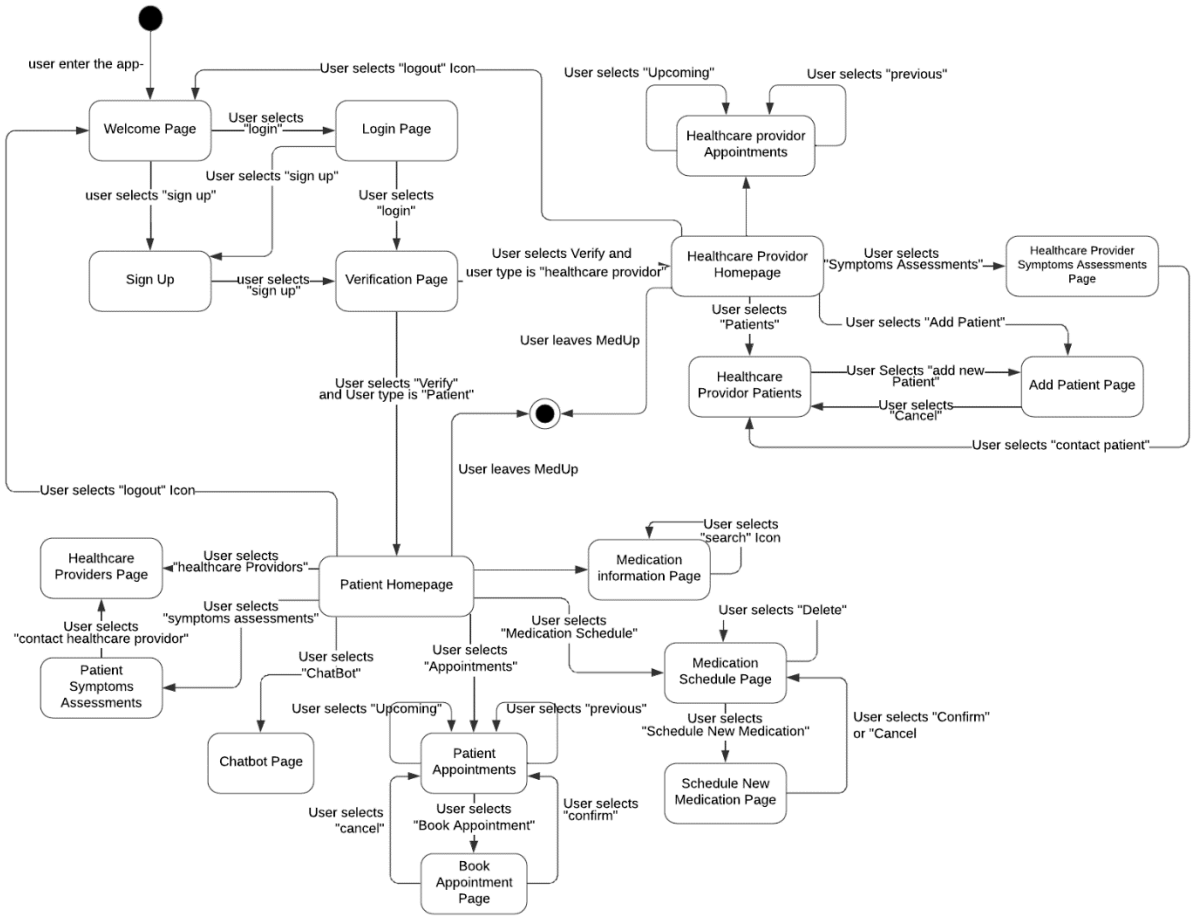

Figure 14:State Diagram for Medup.

F. User Interfaces:

- **Welcome Page:**

  This page will appear when the user first opens the application, and it will give the users two options: Login and Sign Up.

- **Login Page:**

  This page will appear when the user chooses the login option, and the user can login choosing the user type, and providing his/her national ID and password.

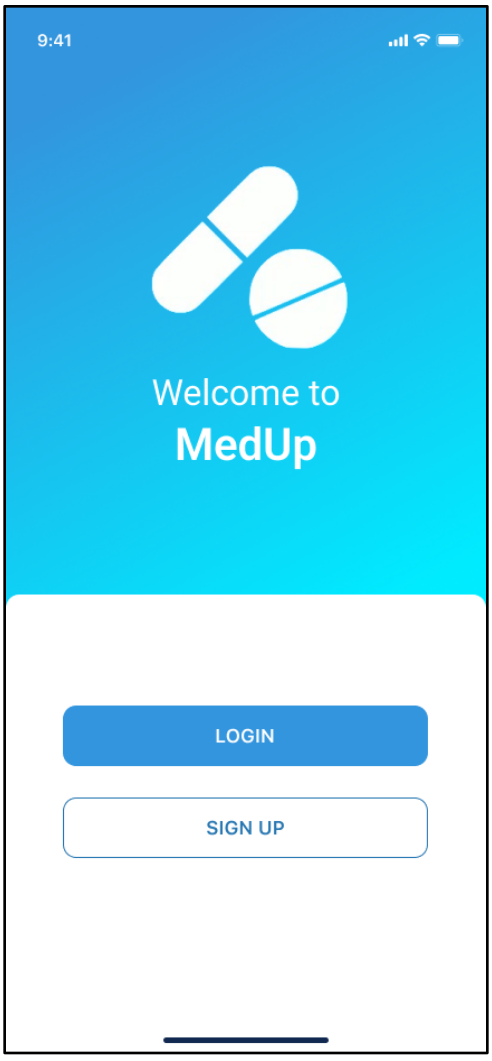

Figure 15: Welcome Page.

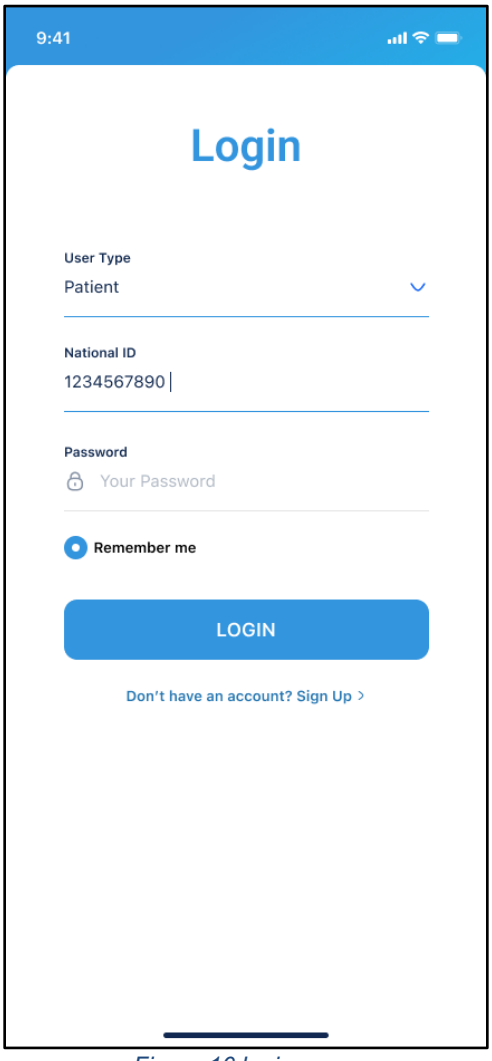

Figure 16:login page.

- **Sign-up Page:**

  This page will appear when the user chooses the sign-up option, and it will allow the user to create an account first choosing the user type, then providing the first and last name, national ID, phone number, password, and optionally profile photo. The user will not be able to create the account unless he/she agrees to the terms and conditions.

- **Verification Page:**

  This page will appear when the user attempts to login, and it will verify the phone number as an authentication method. Additionally, it will verify the phone number provided when the user attempts to sign-up to ensure that the phone number provided belongs to the user.

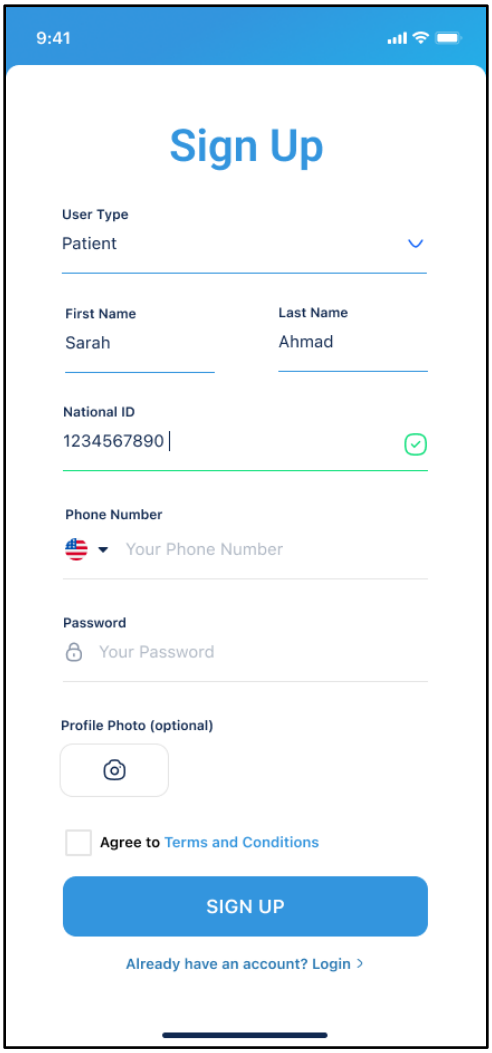

Figure 17:Sign Up Page.

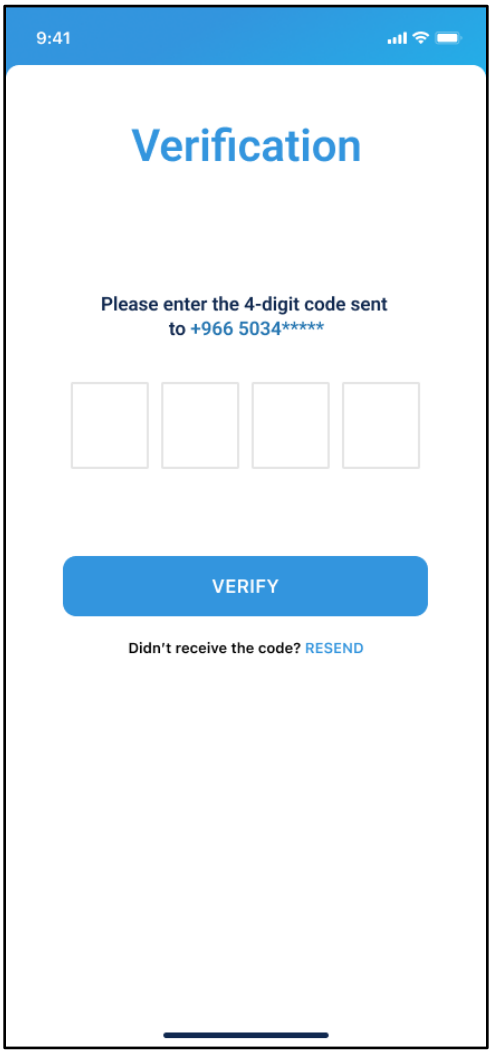

Figure 18:Verification Page.

- **Patient Homepage:**

  This page will appear when the patient logs in, and it shows the main functionalities of the application.

- **Chatbot Page:**

  This page will appear when the user selects Chatbot, and it will allow the user to chat with an AI chatbot to assess his/her symptoms for the recent visit/procedure.

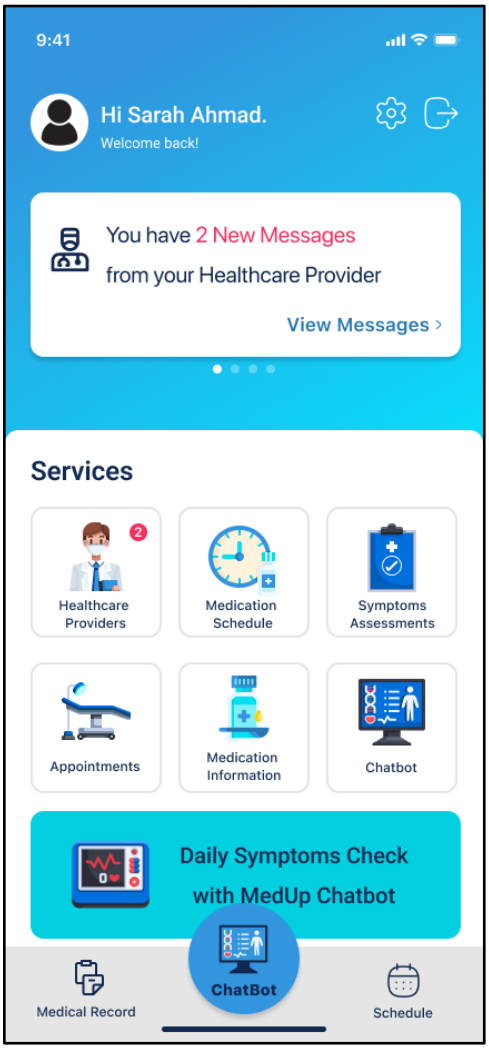

Figure 20:Patient Homepage.

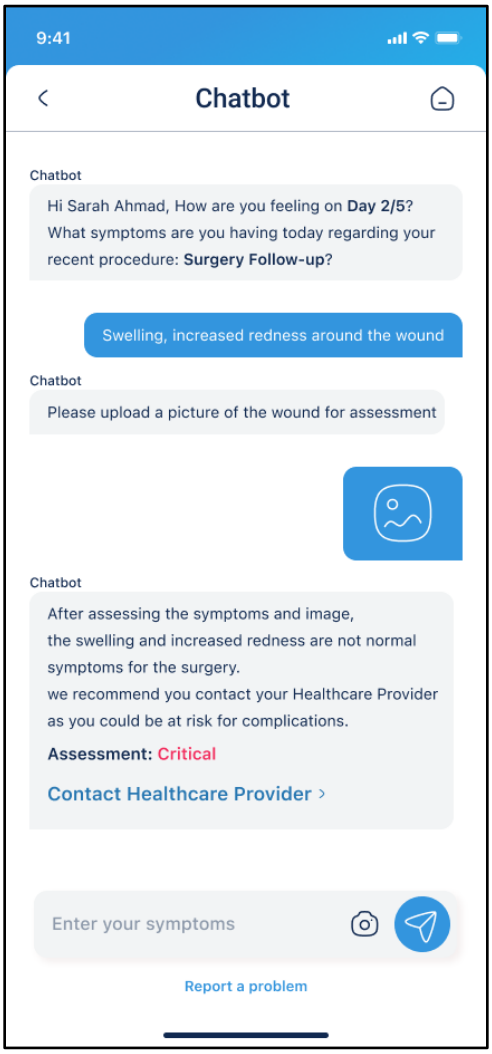

Figure 19: Chatbot Page.

- **Patient Symptoms Assessments page**

  This page will appear when the user selects symptoms assessments option from the home page as well as after finishing an assessment with the chatbot. It will allow the user to view all his/her previous assessments including the final assessment. Additionally, any assessments that were considered critical will be highlighted red and will have an option to contact the healthcare provider.

- **Healthcare Providers Page:**

  This page will appear when the user chooses the healthcare providers option, and it displays all of the patient's healthcare providers with the option to call or message the healthcare provider.

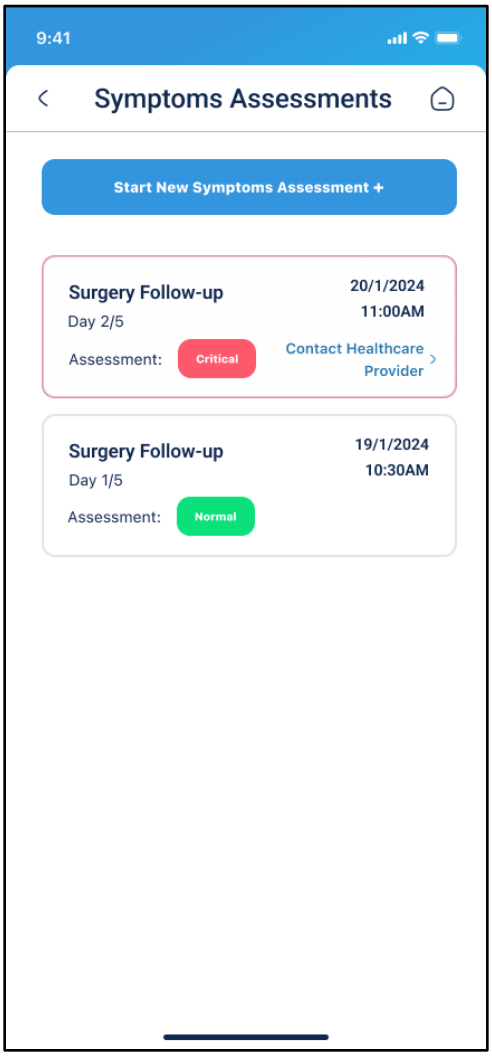

Figure 22:Patient Symptoms Assessments Page.

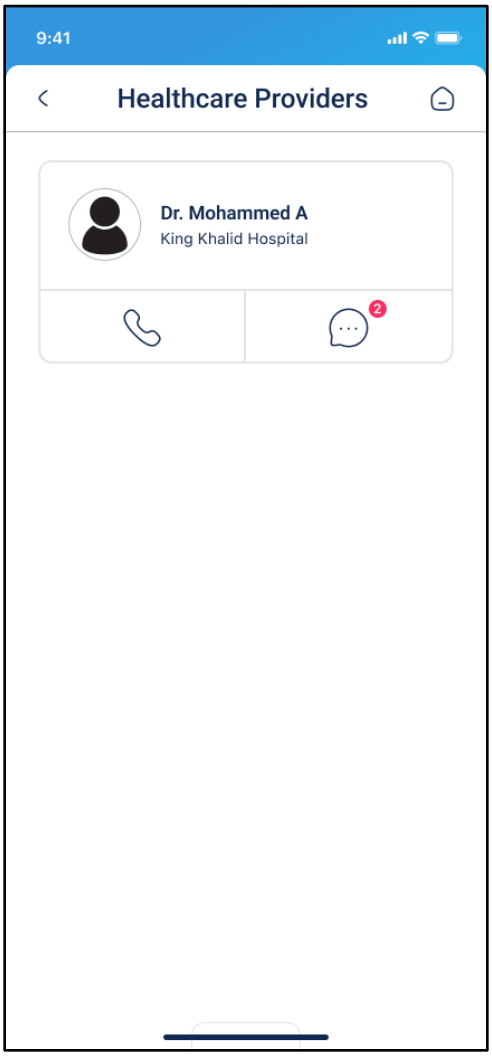

Figure 21:Patient Healthcare Providers Page

- **Patient Appointments Page:**

  This page will appear when the user chooses the appointments option, and it displays upcoming or previous appointments with their details. It will also allow the user to cancel upcoming appointments.

- **Book Appointment Page:**

  This page will appear when the user chooses the book appointment option. It asks the user to choose the appointment details.

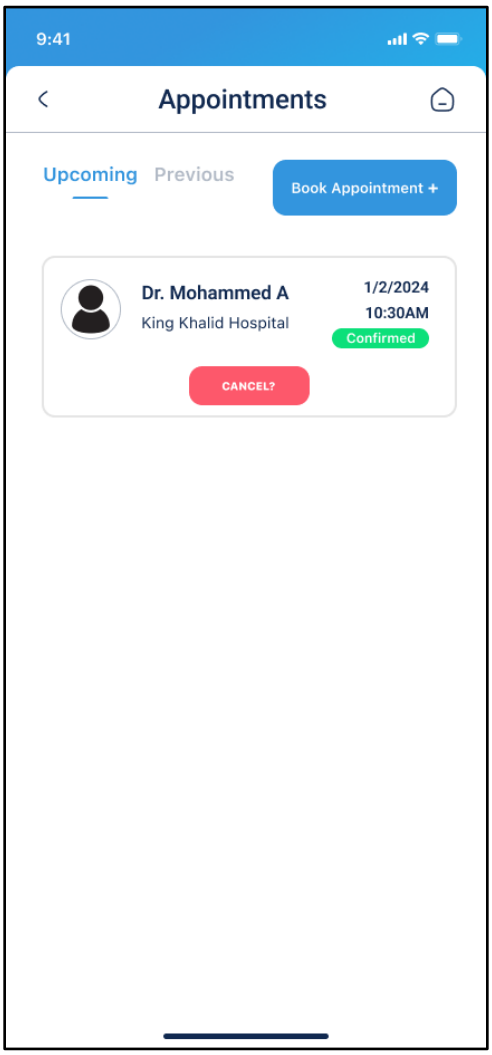

Figure 24:: Patient Appointments Page.

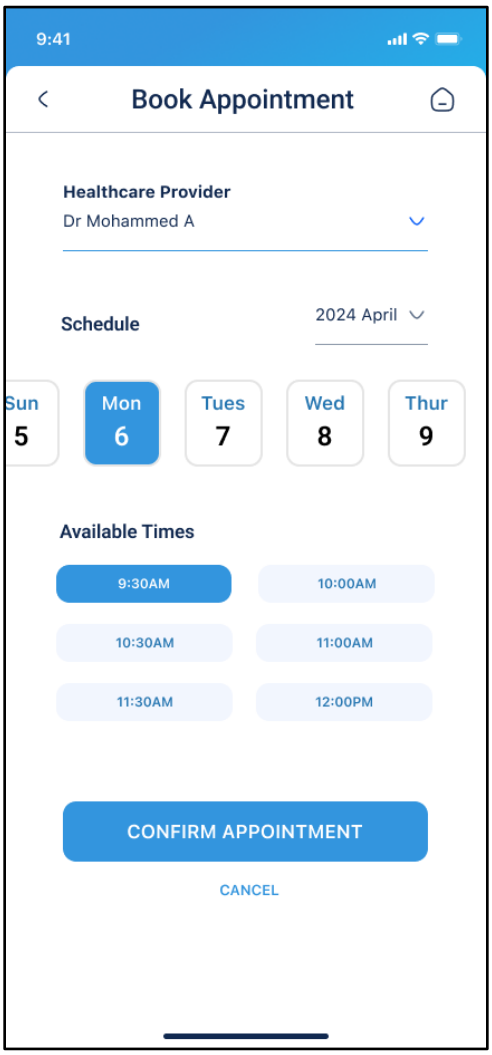

Figure 23:Patient Book Appointment Page.

- **Medication Schedule Page:**

  This page will appear when the user chooses the medication schedule option, and it shows all of the scheduled medication with the option to delete a medication. Additionally, it has the option to add a new medication to the schedule.

- **Schedule New Medication Page:**

  This page will appear when the user chooses the schedule new medication option. It prompts the user to enter the scheduling details.

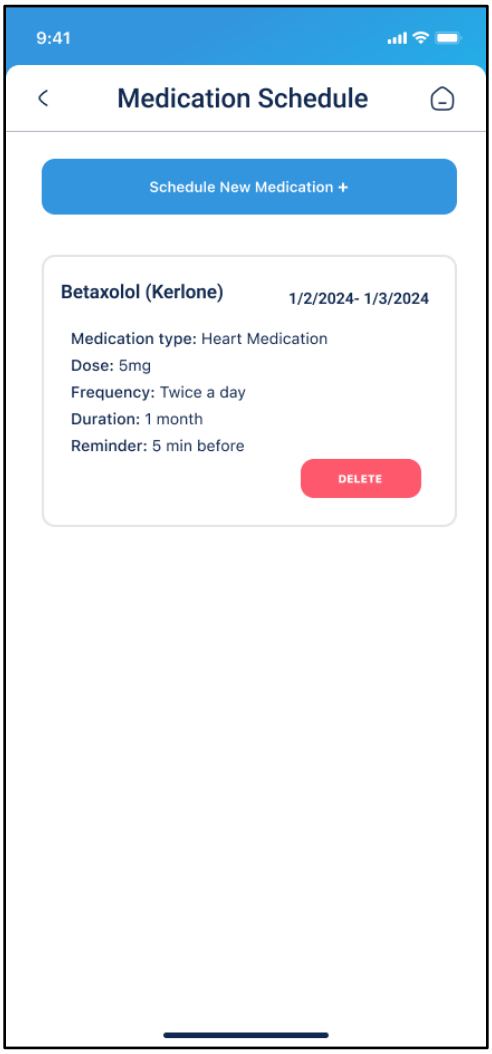

Figure 26:Medication Schedule Page.

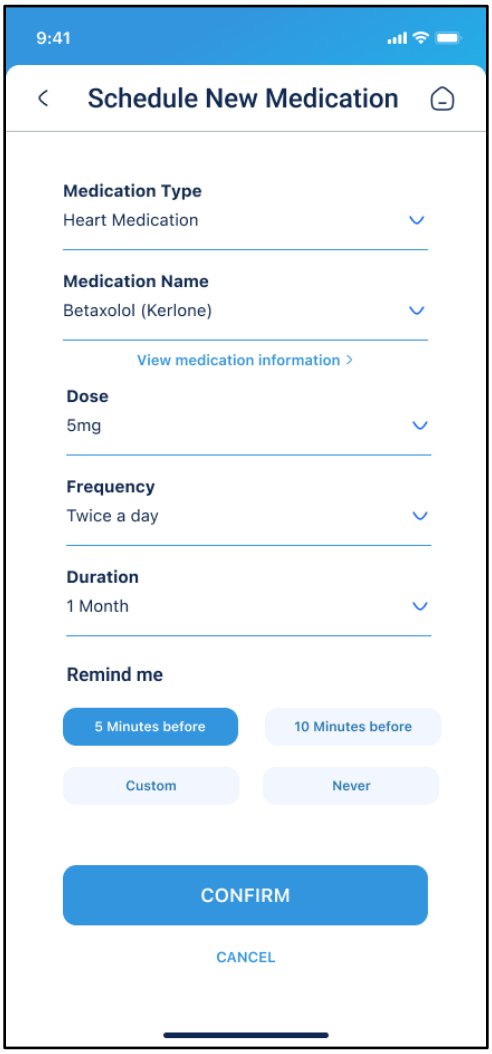

Figure 25:Schedule New Medication Page.

- **Medication Information Page:**

  This page will appear when the user chooses medication information option on the home page or when scheduling medications.

- **Healthcare Provider Homepage:**

  This page will appear when the healthcare provider logs in, and it includes the main functionalities of the application.

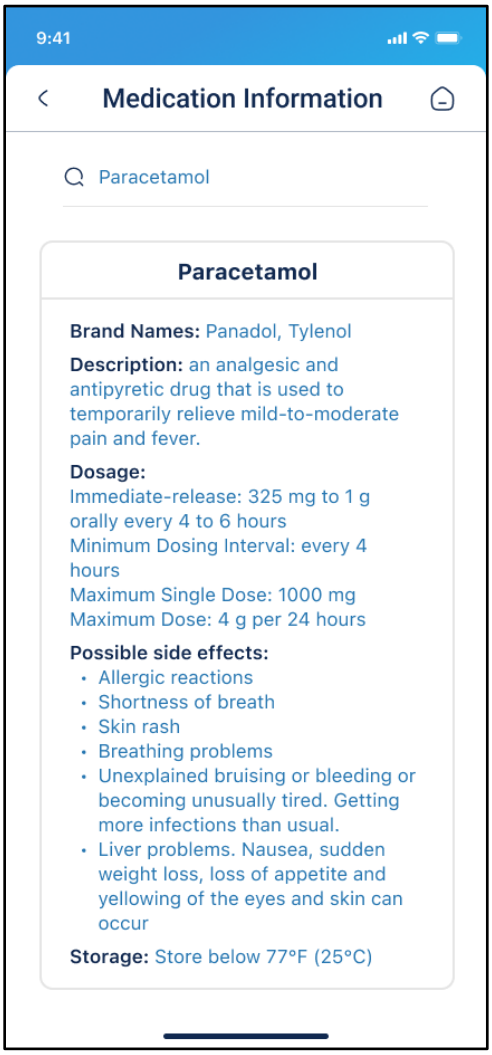

Figure 28:Medication Information Page.

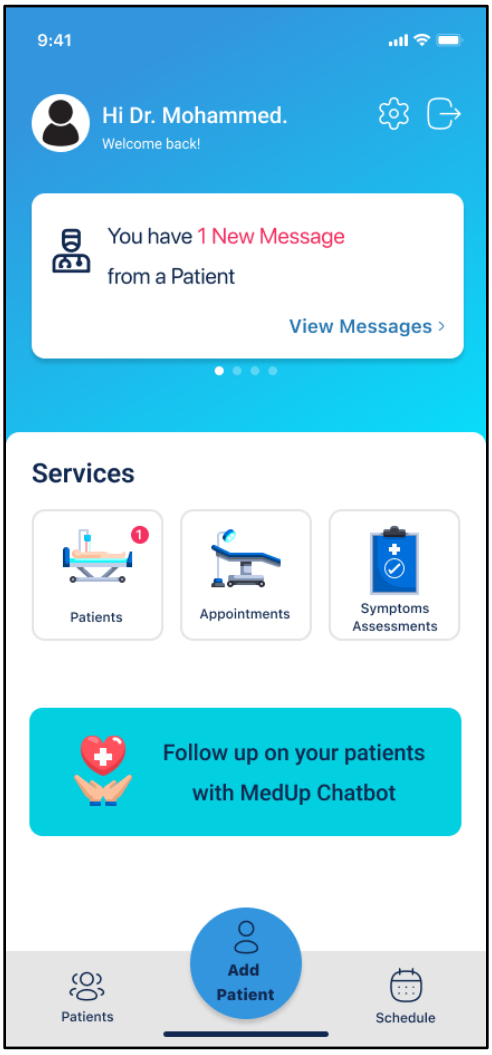

Figure 27:Healthcare Provider Homepage.

- **Healthcare Provider Patients Page:**

  This page will appear when the healthcare provider chooses the patients option, and it will show all the patients for the healthcare provider, and it will allow him/her to view the patient's medical record as well as contact the patient by either call or messaging. Additionally, an option for adding new patients is available.

- **Add Patient Page:**

  This page will appear when the healthcare provider chooses the add patient option. It allows his/her to search for a patient by national ID, and will give the option to link the medical record for the patient.

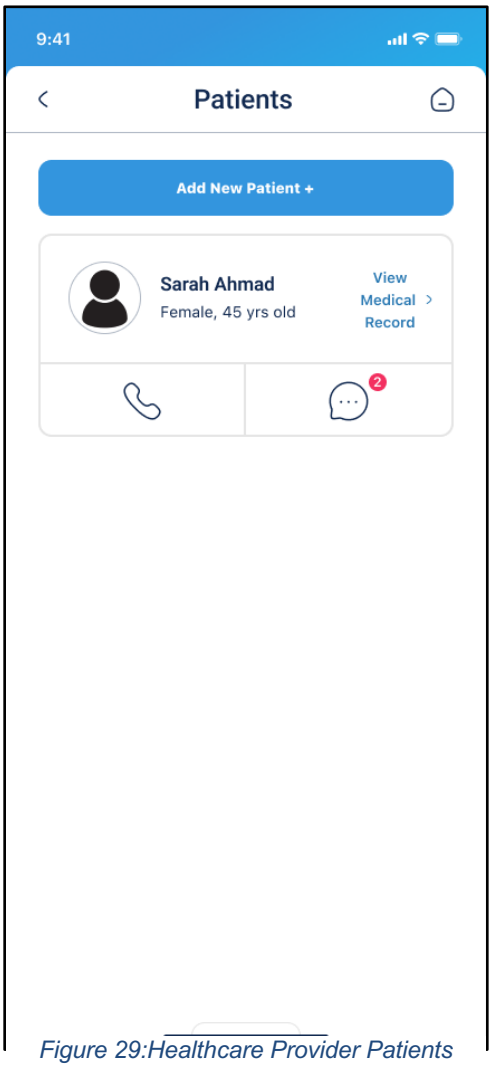

Figure 29:Healthcare Provider Patients Page.

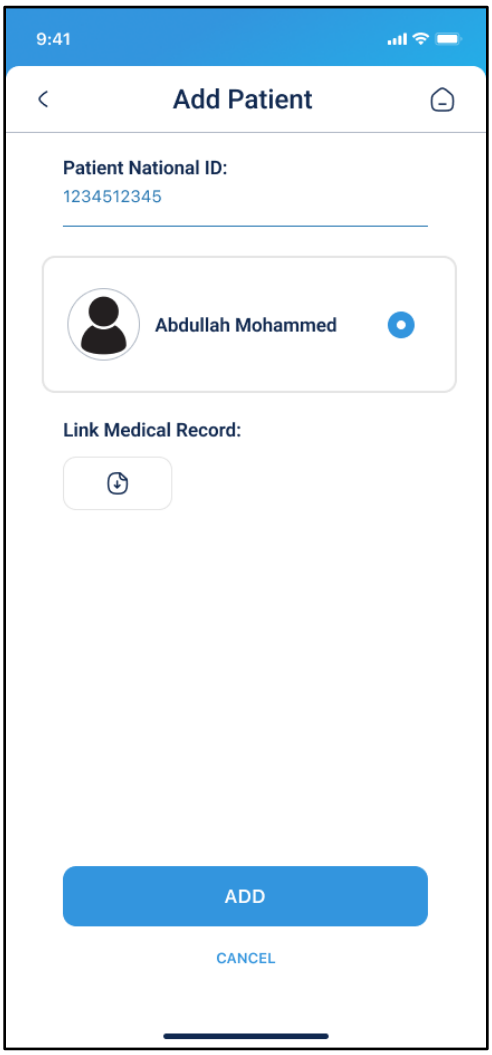

Figure 30:dd Patient Page.

- **Healthcare Provider Appointments Page:**

  This page will appear when the healthcare provider chooses the appointments option, and it displays upcoming or previous appointments with their details. It will also allow the healthcare provider to confirm or reject upcoming appointments.

- **Healthcare Provider Symptoms Assessments:**

  This page will appear when the user selects Symptoms Assessments from the home page. It will allow the user to view all assessments including the final assessment as well as the assessments' details. Additionally, any assessments that were considered critical will be highlighted red and will have an option to contact the patient.

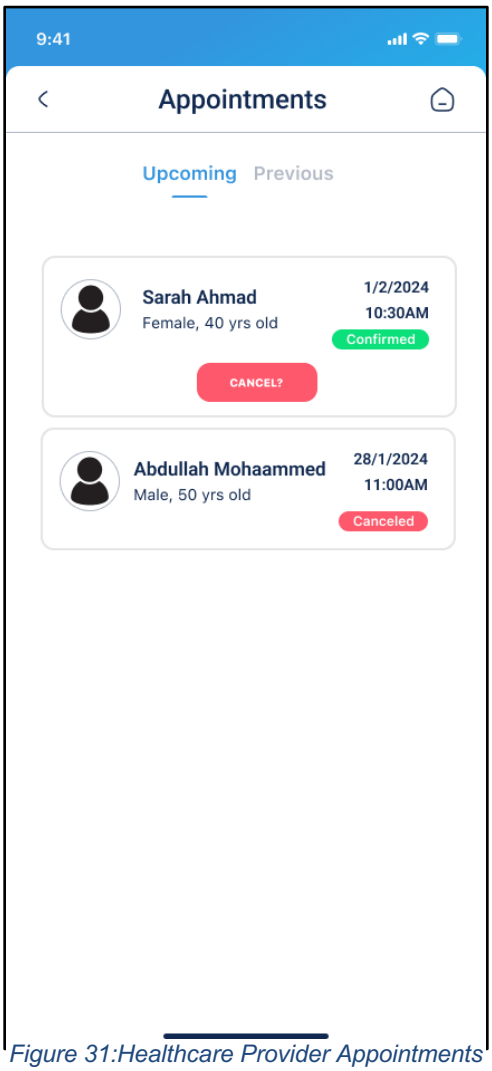

Figure 31:Healthcare Provider Appointments Page.

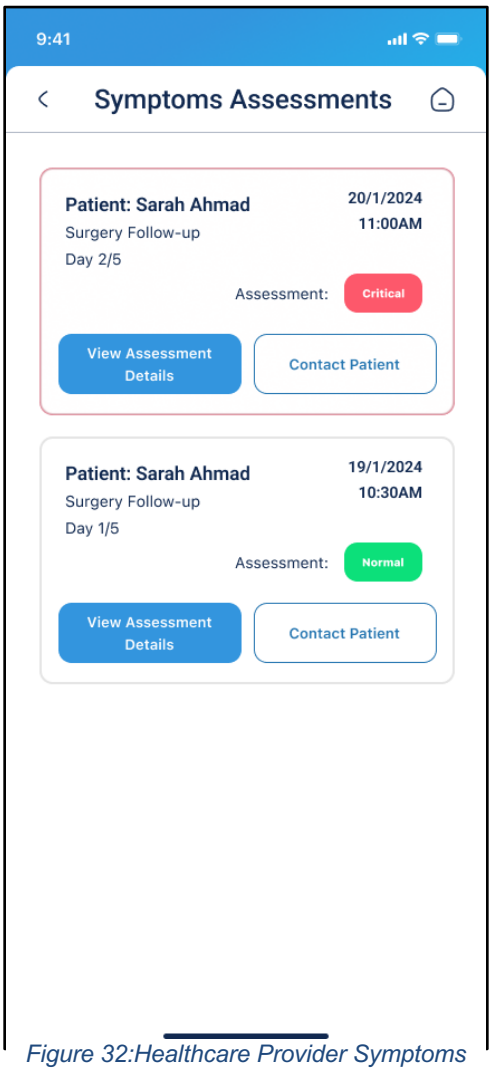

Figure 32:Healthcare Provider Symptoms Assessments Page.

Non-Functional Properties

1-Usability: MedUp system should ensure effective communication by providing a user-friendly interface to navigate it without any difficulty for patients and healthcare providers. Specially users who aren't familiar with technology should be able to learn how to use it in less than 5 minutes.

2- Performance: MedUp system should provide a seamless user experience and rapid system response time for various functionalities which are considered crucial for MedUp's effectiveness. Such as assessing symptom severity, facilitating real-time communication between patients and healthcare providers, processing wound images, or delivering medication reminders. Therefore, any delay of more than 3 seconds could put patient care at risk.

3- Scalability: MedUp system should be highly scalable, it should support simultaneous usage by up to 60,000 users, with the capability of handling a growing amount of patient data , and medical records . Blackboard architectural style supports scalability as we can add agents seamlessly enhancing scalability. Additionally, SOA enhances scalability since new services can be added to the system in the future.

4-Flexibility: MedUp system considers flexibility as a fundamental aspect to meet the patient and the healthcare provider needs effectively. Within our architecture style we have chosen, integration of new features should be seamless as individual modules in SOA are responsible for different functionalities.

5-Reliability: MedUp system should ensure effective patient care especially during emergencies to provide timely care, so any failure or system downtime could Potentially risk patient health and safety. Reliability is supported in our system by MVC architectural style since failures do not influence other components, reliability is crucial to maintaining the independence and isolation of the Model, View, and Controller components in the Model-View-Controller architecture.

Constraints:

1. The System shall run on all iOS platforms.
2. The Project budget shall not exceed 500,000 SAR.
3. The System shall be programmed using Swift programming language.
4. The System shall use Google Cloud Platform (GCP) for database management.
5. The Project shall be completed within 8 months.
6. The System shall be capable of handling up to 60,000 concurrent users.

Quality Assurance

Reviews:

Checklists: Since our system contains many various functionalities, we are introducing checklists to all necessary requirements, steps, tasks are completed. They are used in many stages through planning to maintenance. In development they play a guidance role as they are used to gather requirements, system design, testing. They are used at the beginning and end of the development stage, initially to set a guide for developers to follow and eventually to review the work done. Moreover, checklists support reviews as they act like a template to ensure standards are met and not overlooked.

Walkthroughs: They involve informal peer review sessions where team members of similar organizational levels systematically review and discuss a piece of software presented by a colleague. These sessions allow for questions to be asked and potential errors to be identified in a collaborative environment. They are Conducted periodically throughout the SDLC but the most in development. usually done by the Development team members, QA specialists and there is no need for preparation.

Verification:

For verifications, Inspections are a quality assurance technique, and it is the most formal type of reviews. It is done by a group of inspectors in higher management who are experienced along with some members from the development team to present and answer questions. They conduct a thorough inspection on all development documents to find defects and to give approval to go to the next phase if succeeded. Inspections are crucial in ensuring expected quality standards are met and they are done at the end of the development stage.

Validation:

We will do validation to make sure that the MedUp app meets the stakeholder's needs. Validation includes conducting real testing on the software product. It answers the question " Are we developing the right product? ". It also guarantees the identification of defects and flaws that may have been overlooked during the verification process.

Unit testing:

We will initiate the validation process for the system units, with each unit comprising a little and interconnected code. By doing separate tests on each unit, we can guarantee that each unit operates perfectly, thereby contributing to the overall functionality of the software.

- We will use Black box test. It involves testing a system without any knowledge of its internal structure. It's beneficial for assessing the overall functionality of the system and user experience of the system. [10]

Integration testing:

Following unit testing, our next step is to initiate the integration of units. This process aims to guarantee effective collaboration between the units and to ensure the seamless flow of data between them.

- We will use the Top-Down approach. This approach starts with testing the top-level components first, progressively start testing lower-level components. It allows for early identification of any issues in the higher-level structure of the system. [11]

System testing:

We will end the validation process by doing system testing, where we will test the software as an integrated whole to check the interactions between its components and to evaluate its whole functionalities.

- We will do Acceptance Testing which is known as user acceptance testing(UAT) to guarantee that the users can complete the goals set in the requirements, which determines whether the application is ready for delivery or not. [12]

- We will do Load Testing which involves putting load on the system to guarantee it remains constant during its operation. The goal is to identify how well the system operates when subjected to a specific amount of workload or concurrent user activity. This type of testing helps identify performance bottlenecks, assess system capacity, and to ensure that the application can handle the expected load without affecting its performance. [12]

- We will do Regression Testing. It's a system testing technique that it's used to re-running previously executed test cases to find any bugs that may arise due to changes or additions to the codebase and to ensure that it will not negatively affect the existing functionalities of the software. [12]

Acceptance criteria:

We will establish criteria outlining the procedure for users to install the software structured as follow:

Procedure:

The installing procedure will be user-friendly, with clear documentation presented in a step-by-step format. Additionally, the distribution will take place through an App Store.

Testing:

Integration testing: This process aims to guarantee effective collaboration between the units and to ensure the seamless flow of data between them.

Training:

We can arrange training sessions to familiarize users with the software functionalities. It can be orientation sessions or interactive tutorials.

Documentation:

Documentation holds significant importance for clients, and therefore we will provide a user manual for our system. This manual is designed to describe the MedUp system comprehensively and assist users in efficiently navigating and utilizing its functionalities.

Future Considerations

Ensuring the growth of **Medup** can be done via different approaches, one of which is to make sure that it is accessible for everyone or at least the majority of potential users, to achieve this objective we may support different Arabic accents in the AI chatbot. So that seniors are able to use the app on their own without needing any help.

One important feature that can be added to the AI chatbot is suggesting alternative medicines with similar formulas while also explaining the differences between them if there was any, this feature can be quite useful in many situations, snice pharmacies can run out of some medicines. This feature can also be separated from the chatbot.

The industry is always growing as the resources and researchers are constantly increasing and evolving, for that reason the machine learning chatbot model will need to be adjusted to new changes and new resources with extensive data training , periodically and as frequently as possible.

User trust on **Medup** is one of our biggest concerns as discussed in challenges, so testing the model constantly is a must. The model shall be tested by a team of actual doctors to ensure accuracy on the trust of our users, releasing a public report of the testing document may be needed to keep the users satisfied and safe.

Integrating **Medup** with (Sehhaty) may be a good addition to increase the number of users. since there are thousands of users that are using it and comfortable with it, by allowing the users to connect their accounts they can talk to and schedule appointments with the public health institutions healthcare providers while also getting the benefits of **Medup** chatbot and image processing features.

References

[1] What is natural language processing?. IBM. (n.d.). https://www.ibm.com/topics/natural-language-processing

[2] GeeksforGeeks. (2023, February 22). Digital Image Processing Basics.GeeksforGeeks. https://www.geeksforgeeks.org/digital-image-processing- basics/

[3] Ekareadmin. (2023, December 14). Advanced wound imaging | Wound measurement | Wound care App. eKare Inc. https://ekare.ai/advanced- wound-imaging/

[4] *What is artificial intelligence and how does AI work? TechTarget*. (2023, July 10). Enterprise AI. https://www.techtarget.com/searchenterpriseai/definition/AI- Artificial-Intelligence

[5] (2024, February 13). TeamGantt: The Easiest FREE Gantt Chart Maker Online. https://www.teamgantt.com/

[6] Google. (n.d.). *Cloud computing, hosting services, and apis*. Google. https://cloud.google.com

[7] *Agora Real-Time voice and video engagement*. (2024, March 21). Agora. https://www.agora.io/en/

[8] *yakeen*. (n.d.). elm. Retrieved April 2, 2024, from https://www.elm.sa/ar/e-services/Pages/yakeen.aspx

[9] *What is a chatbot? | IBM*. (n.d.). https://www.ibm.com/topics/chatbots

[10] Hasson, E. (2023, December 21). *What is Black Box Testing | Techniques & Examples | Imperva*. Learning Center. https://www.imperva.com/learn/application-security/black-box-testing/

[11] Terra, J. (2023, May 19). *What is Integration Testing: Examples, Challenges, and Approaches*. Simplilearn.com. https://www.simplilearn.com/what-is-integration-testing-examples-challenges-approaches-article

[12] Kinsbruner, E. (n.d.). *The complete guide to different types of testing | Perfecto by Perforce*. Perfecto by Perforce. https://www.perfecto.io/resources/types-of-testing