National University of Sciences & Technology (NUST)
School of Electrical Engineering and Computer Science (SEECS) Department of Computing

Final Year Project

Software Requirements Specification

Baymax – Remote Patient Monitoring System

Version [1.0]

Bisma Ijaz - 351995
Vimal - 343007
Syed Fazail Haider - 346062

Dr. Muhammad Daud Abdullah Asif (Advisor)
Dr. Wajid Mumtaz (Co-Advisor)

Date of preparation: 3rd November, 2023

Table of contents

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

The product named ‘Baymax’ is a remote patient monitoring system composed of two integral parts: the ‘Baymax Kit’ which is a medical monitoring kit containing sensors and microprocessors, and the ‘Baymax Portal’ which is an integrated Web-based platform. This version (1.0) document describes the ‘Baymax Portal’ that aims to provide services for the doctors and patients alike for the purpose of remote patient monitoring. By connecting patients with doctors, transmitting patient body vitals from the Baymax kit, and using ML based logic to predictions to aid diagnosis of clinical events, ensures timely medical interventions.

1.2 Document Conventions

The conventions for typography in our document are as follows

1. All the text is in font “Times New Roman.”
2. The font size for a normal body of text is 12px.
3. The font size for first level headings is 18px.
4. The font size for second level headings is 16px.
5. The font size for third level headings is 14px.

Secondary bullets are used to show further detailed information. Sub-numbering has been used to show sub-points. Higher numbers don't show more priority. The titles have been bold to make it more clear and visible what the information given is about. Bullet points are used within the whole document to make it more readable.

1.3 Intended Audience and Reading Suggestions

1.3.1 Intended Audience

1. Technical Team

A. **Developers:** The developers are the primary audience responsible for building the system. They need detailed technical specifications and requirements for the system's development.

B. **Data Scientists*:*** Data scientists play a crucial role in implementing machine learning algorithms. They require in-depth knowledge of data processing, analysis, and prediction.

2. Project Stakeholders

A. **Project Advisors and Committee Members:** Project managers need an understanding of the project's scope, goals, and milestones to ensure successful execution.

B. **Medical Authorities:** Regulatory authorities may need to review the document to ensure compliance with healthcare and data privacy regulations.

3. End Users:

A. **Patients:** Patients are the ultimate users of the system. They need to understand how the system benefits them and how to use and interact with it.

B. **Doctors:** Healthcare professionals need to know how the system aids their practice, including appointment management, data access, and communication.

C. **Administrators:** System administrators should grasp how the platform helps manage users, appointments, and overall system functionality.

1.3.2 Suggested Sequence for Reading

1. For Technical Team

1. Start with the **Product Perspective and Architecture Diagram** to understand the project at a high level.
2. Review the **User Classes and Use Case Diagrams** for in-depth functionality descriptions.
3. Proceed to the **Detailed Functional Requirements** to grasp the core functionalities.
4. Examine the **Hardware and Software Requirements** to understand the technology stack.

2. For Project Stakeholders

1. Begin with the **Introduction** to get an overall understanding of the project's purpose.
2. Move to the **Product Scope** to understand the goals, objectives and characteristics of the project.
3. Furthermore, go through the **Product Perspective and Architecture Diagram** to understand the project at a high level.
4. Consider the **Business Requirements** to understand the project's business context.

3. For End Users (Patients, Doctors, and Administrators)

1. Start with the **Introduction** to understand the project's purpose and benefits.
2. Read the **Overall Description** for a high-level understanding of the system's functionality.
3. Focus on the **User Interfaces and Functionalities** to learn how to use the system effectively.
4. Learn how to use the **Patient, Admin, and Doctor Dashboards** for role-specific functionality.

This reading sequence ensures that each type of reader can access information relevant to their role and needs within the "Baymax" project.

1.4 Product Scope

There are several platforms in Pakistan like eShafi, Sehat Kahani, and InstaCare [1] that provide digital solutions for remote patient monitoring. These applications provide online management systems and virtual consultation services but lack remote transmission of bio signals of patients and predictions of clinical events to aid diagnosis. Furthermore, according to the National Health Vision of Pakistan 2016-2025 [2], the nation faces several challenges in terms of accessibility, timely diagnosis, and efficient communication between patients and healthcare providers.

'Baymax Portal' represents an innovative healthcare system that aims to reshape the landscape of remote patient monitoring in Pakistan. It aims to address these issues by providing an integrated web-based platform that offers a range of services to patients, doctors, and administrators. It is specifically designed to serve a diverse range of individuals who require remote medical care, encompassing the elderly, those in need of regular checkups, individuals managing chronic conditions, expectant mothers, residents of remote areas, and individuals with mobility challenges like paralysis.

In pursuit of its mission, Baymax has set clear objectives and goals, which include:

1. Enhancing remote patient monitoring through utilization of the 'Baymax Kit' to provide patients with a reliable and convenient method for continuous health tracking.
2. Establishing efficient communication channels between patients and healthcare providers, reducing response times, and improving overall patient care.
3. Leveraging the power of machine learning to improve the accuracy of medical condition predictions by analyzing correlations among various bio signals to enhance the precision of diagnoses and empower healthcare providers to offer more effective care to patients.
4. Securely storing and analyzing patient data in the cloud.
5. Streamlining administrative processes and providing an efficient healthcare management platform.

In doing so, our vision aligns with one of the pivotal pillars 'Health Information Systems' in the National Health Vision of Pakistan [2] which aims to incorporate innovative technologies and integrate disease surveillance and response systems, with a particular focus on Early Warning System. Baymax stands as a comprehensive solution poised to bridge these gaps and elevate the standard of healthcare delivery across the nation.

1.5 References

[1] Kazi, A. M. et. al (2020). Current Challenges of Digital Health Interventions in Pakistan: Mixed Methods Analysis. Journal of medical Internet research, 22(9), e21691. https://doi.org/10.2196/21691

[2] World Health Organization (2016). National Health Vision Pakistan 2016-2025, 2016. https://phkh.nhsrc.pk/sites/default/files/2020-12/National%20Health%20Vision%20Pakistan%202016-2025.pdf

1.6 Table of Figures

2. Overall Description

2.1 Product Perspective

'Baymax Portal' is a self-contained product which operates independently with its ecosystem but interacts with external components to provide a comprehensive healthcare solution. It interfaces with health monitoring device kit 'Baymax Kit,' utilizes cloud services for data management, offers real-time communication features, and integrates external machine learning algorithms for clinical events predictions. A simplified diagram of the overall system might include the following major components and their interconnections:

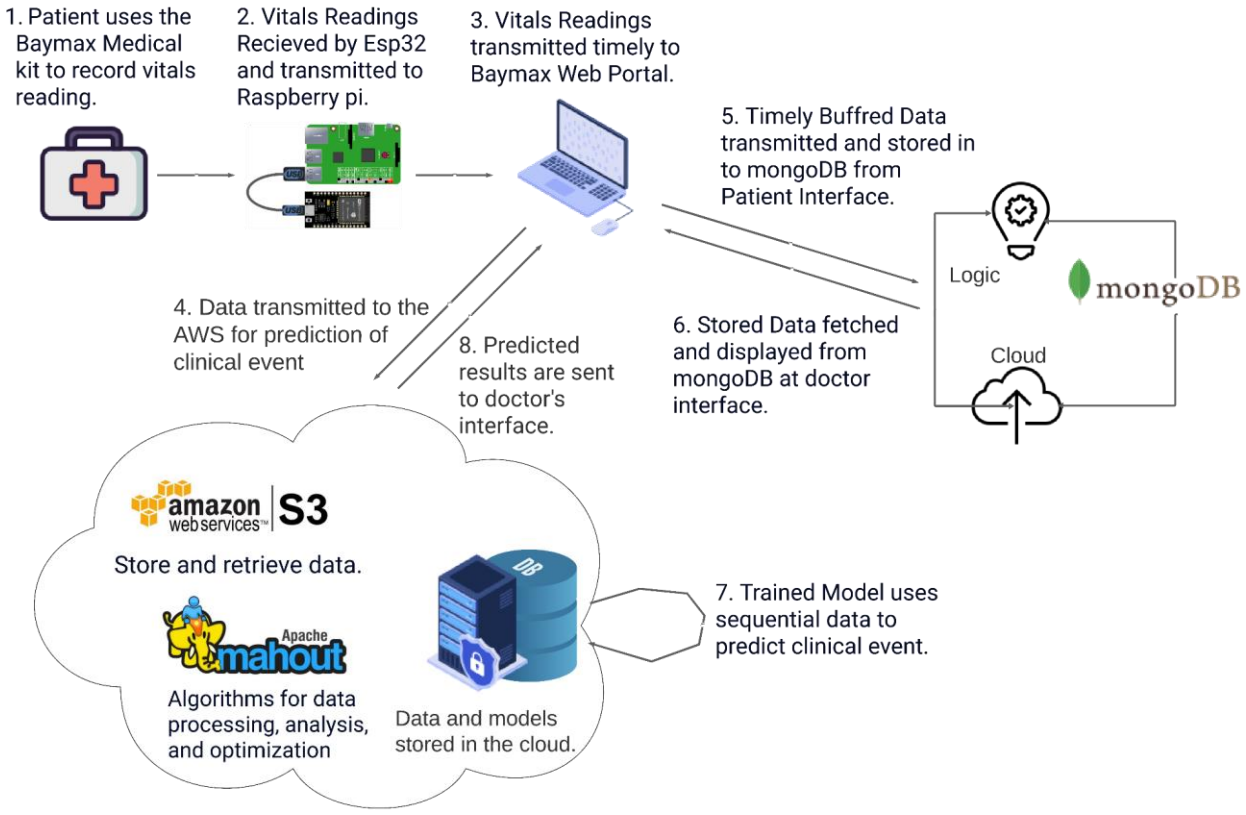

Figure 1. System Architecture

2.2 Product Functions

Here is a high-level summary of the major functions that the Baymax Portal must perform or allow users to perform:

- Register new users as patients, doctors, or administrators
- Create and manage user profiles with personal and medical information

- Provide customized dashboards for patients, doctors, and administrators with role-specific functionalities
- Enable real-time chat between patients and doctors
- Facilitate appointment booking and management
- Transmit vital signs and health data from monitoring hardware
- Utilize machine learning algorithms to predict clinical events
- Present predictive analysis to doctors for diagnosis
- Securely store and manage patient data in a cloud-based system
- Ensure data privacy and compliance with regulations

2.3 User Classes and Characteristics

In the context of the Baymax product, several distinct user classes can be identified, each with its unique characteristics and needs. These user classes have different roles and responsibilities within the system. Distinguishing the most important user classes is crucial, as their requirements may take precedence over others. Here are the primary user classes for Baymax:

1. Patients

Patients are the end-users of the system, often without extensive technical expertise. They have varying levels of medical knowledge and may use the system periodically to monitor their health. They belong to the most important user class. Patients need easy-to-use interfaces for data monitoring, appointment booking, and communication with doctors. Data privacy and security are paramount for this user class.

2. Doctors and Healthcare Providers

Doctors and healthcare professionals are experts in the medical field. They use the system extensively for diagnosis, communication with patients, and treatment planning. Doctors require tools for data analysis, communication, access to patient histories, and machine learning predictions. Data accuracy and system reliability are crucial for this user class.

3. Administrators

Administrators have access to the system's backend for management and oversight. They are responsible for user account management, system monitoring, and data privacy. Administrators need functionalities for user and system management.

The following use case diagram illustrates the main user characteristics as follows:

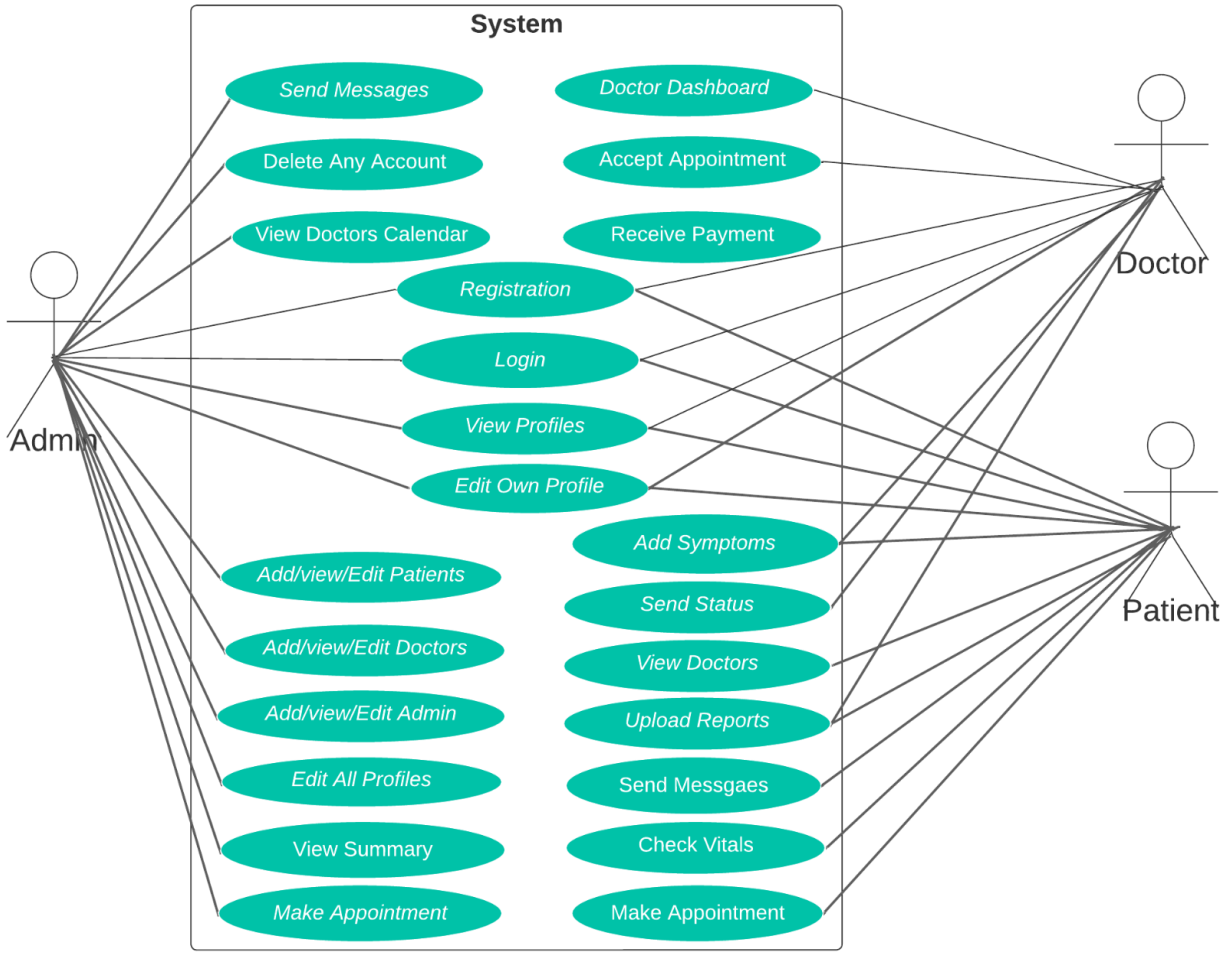

Figure 2. Use Case Diagram

2.4 Operating Environment

The Web portal must be compatible with the following operating environments:

- Browsers: Google Chrome, Mozilla, Microsoft Edge
- Database: MongoDB
- Platform: MERN Framework
- Cloud Storage: AWS

2.5 Design and Implementation Constraints

There are several items and issues that will limit the options available to the developers of the Baymax software. These constraints include:

1. **Web Technologies:** Baymax is built on the MERN (MongoDB, Express, React, Node.js) stack. Developers must adhere to the use of these specific technologies and frameworks to maintain consistency and compatibility within the system.
2. **Regulatory Compliance:** The software must adhere to healthcare and data protection regulations specific to the region of Pakistan. This includes ensuring data privacy and security as well as compliance with any emerging healthcare regulations. Developers must work within these legal boundaries.
3. **Security and Privacy:** Protecting patient data is of paramount importance. Developers are constrained by the need to implement robust security measures, encryption protocols, and access controls to ensure data privacy.

These constraints and limitations provide a framework within which the developers must operate to ensure that Baymax meets its objectives, complies with regulations, and functions effectively while maintaining security, data privacy, and reliability.

2.6 User Documentation

The users might be provided with detailed video instructions or user manuals on how to use the system. Complete Software Requirements Specification will be provided.

2.7 Assumptions and Dependencies

Here are some assumptions and dependencies listed below for Baymax:

1. **User Behavior and Adoption:** The assumption that users can read and write English, have enough resources to use the system, and will adopt and engage with the software as intended.

2. **Network Availability:** The assumption that users will have reliable access to the internet. If there are network issues or disruptions, this may affect real-time communication and data synchronization.

3. **Hardware and Infrastructure:** The assumption that the chosen hardware and infrastructure for hosting the software can support the expected user load. Any limitations or unexpected performance issues may require adjustments to the system.

4. **Data Privacy Regulations:** The assumption that existing and future data privacy regulations in Pakistan will not significantly change during the project's development. Changes in regulations could necessitate adjustments to the software to maintain compliance.

Addressing these assumptions throughout the project's development and adapting to changing conditions or factors is crucial to successfully meet the software's requirements and objectives.

3. External Interface Requirements

3.1 User Interfaces

The Baymax software interfaces with different user classes: patients, doctors, and administrators. Each interface is designed with specific characteristics tailored to the needs of the respective user class. Here, we provide an overview of the logical characteristics of each interface:

1. Common Interface

This is a common interface used by all users. A basic prototype for the landing page for users that are not registered is shown below.

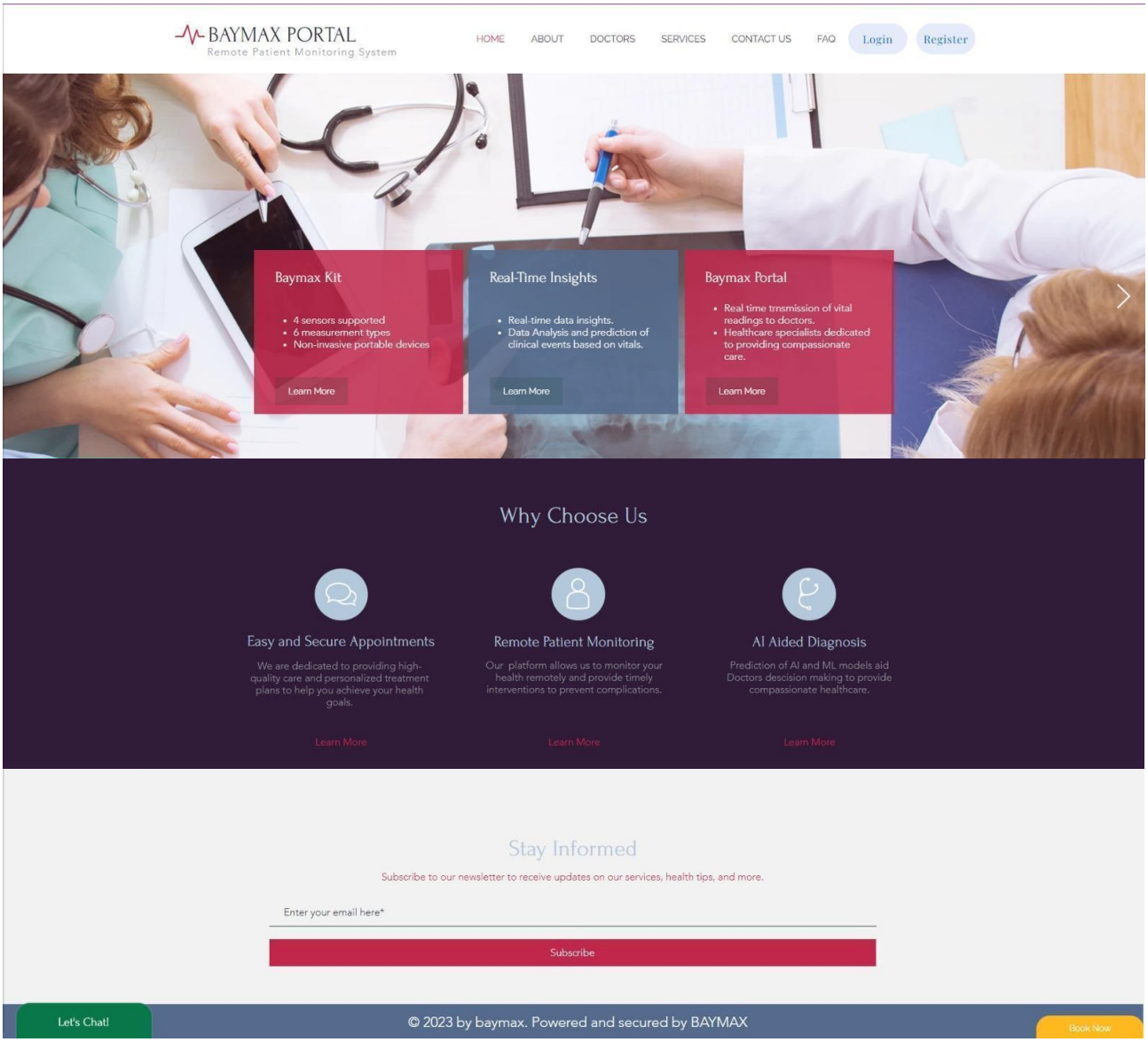

Figure 3. Common Interface

Following the process of registration each user will be directed to dashboards based on specified roles i.e patient, doctor or admin. Each specific dashboard will have unique features for the user to interact with. The overall functionalities with respect to user roles are shown in the use case diagram in Figure 2 of section 2.3.

2. Patient Dashboard Interface

The patient interface includes a dashboard with timely buffered vital sign data, appointment booking, data access to doctor, and chat features for communication with doctors etc. Prototype of the interface is as follows:

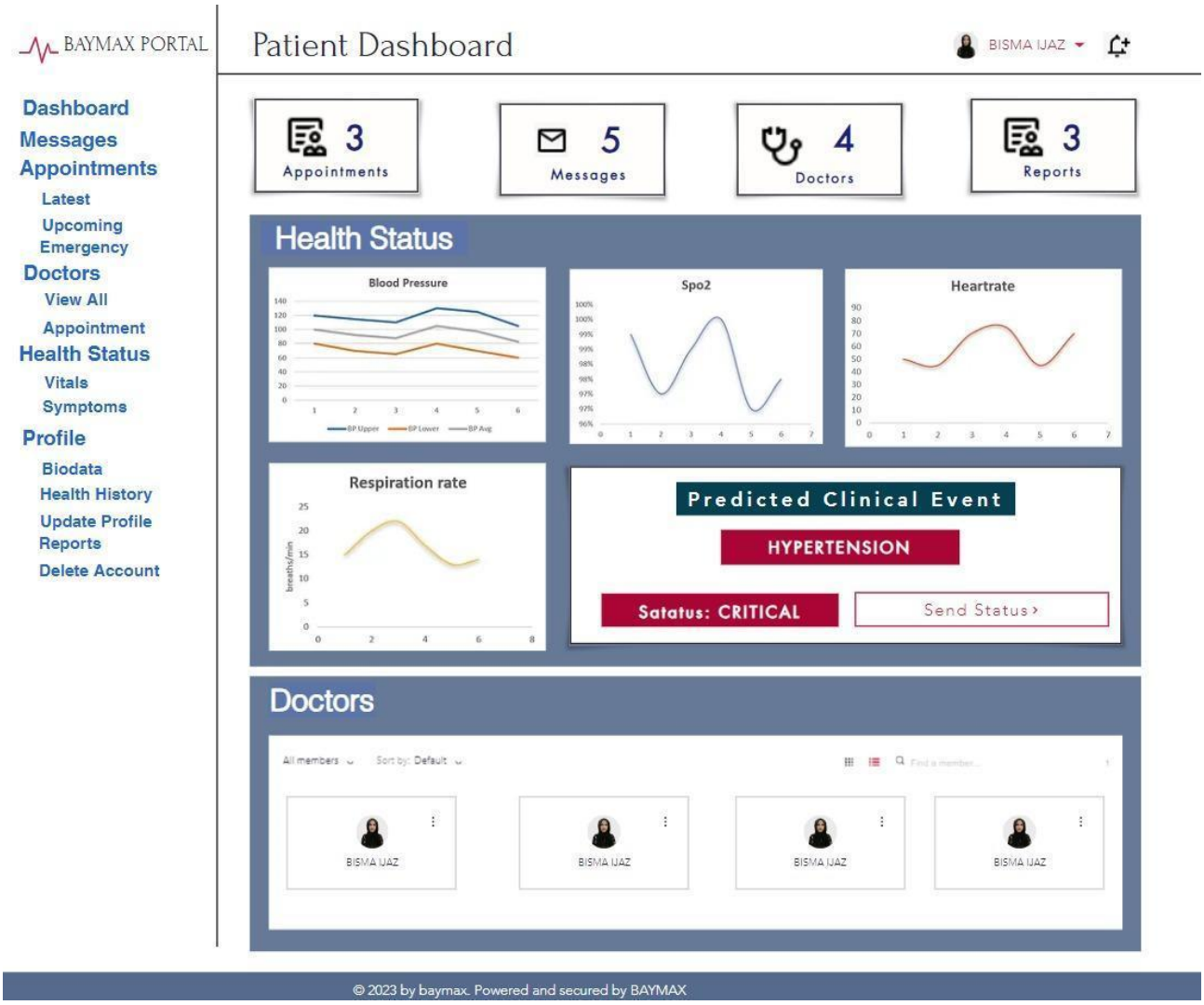

Figure 4. Patient Dashboard

3. Doctor Interface

The doctor interface offers access to patient data, predictive medical condition analysis, and real-time patient communication etc. Sample image of the interface is as follows:

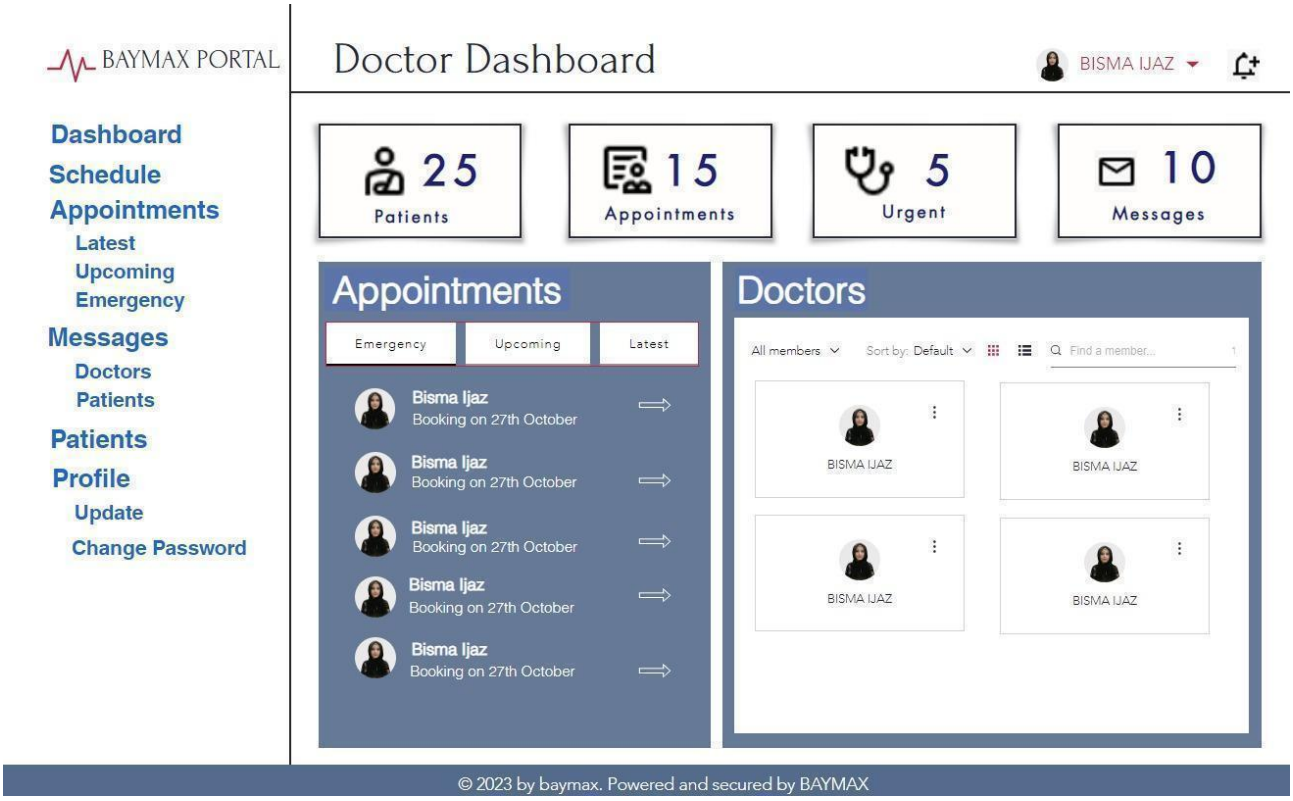

Figure 5. Doctor Dashboard

4. Administrator Interface

The admin interface offers an overview of the system and user management functionalities etc. Sample image of the interface is as follows:

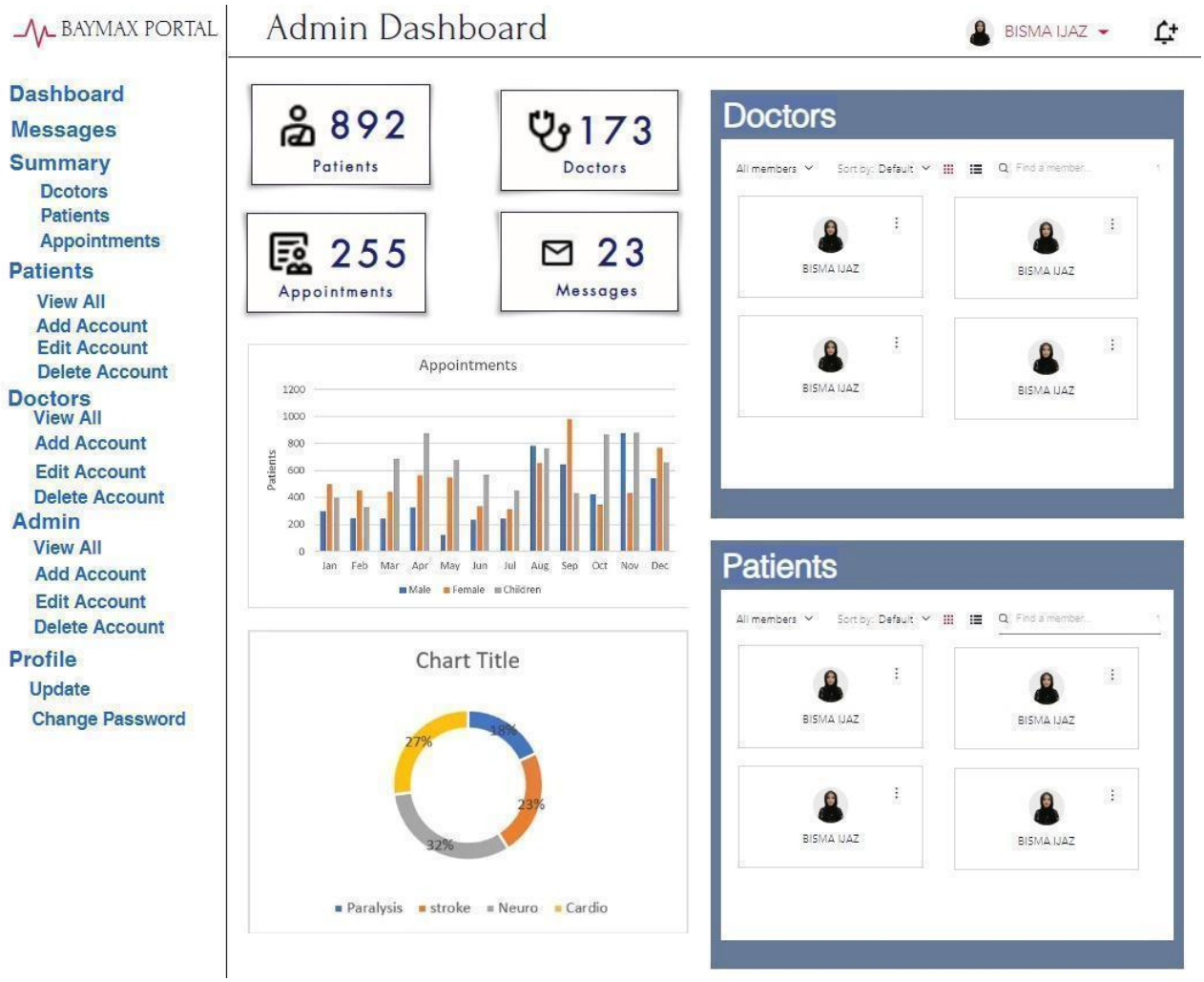

Figure 6. Admin Dashboard

The logical characteristics of each interface are designed to cater to the specific needs and responsibilities of the respective user classes.

1. **GUI Standards:** The interface will follow a clean and intuitive design with easy navigation and data visualization. It will adhere to user-friendly design conventions.
2. **Standard Buttons/Functions:** The interface includes buttons for data sharing, and functions.
3. **Keyboard Shortcuts:** Basic keyboard shortcuts for navigation will be available.
4. **Error Messages:** Clear and user-friendly error messages for data input and system feedback will be used.

3.2 Hardware Interfaces

The Baymax portal interacts with the Baymax medical kit that comprises sensors and microprocessors. Critical body vitals / bio signals of patients will be transmitted. Depending on what works the best, communication may occur through Bluetooth, Wi-Fi, or other wireless protocols using web sockets.

The web portal itself comprises frontend and backend components that will be hosted on a cloud platform. The primary protocol employed for this purpose will be HTTP/HTTPS.

3.3 Software Interfaces

The Web application interfaces with various software components:

1. **MongoDB (Database):** The application uses MongoDB as the database to store and retrieve data. It communicates with MongoDB through the Mongoose library for data modeling and interactions.
2. **Express (Server):** Express.js serves as the backend server framework, handling HTTPS requests and responses. It defines API routes and interacts with the MongoDB database.
3. **React (Frontend):** The React library is used for building the user interface. It communicates with the Express backend via HTTPS requests, fetching and displaying data to the user.
4. **Node.js (Runtime Environment**): Node.js is the runtime environment for the server. It executes server-side JavaScript and interfaces with the Express framework.
5. **AWS (Amazon Web Services):** AWS is utilized for the space of model training and deployment. The application interfaces with AWS services for machine learning model training.
6. **API Endpoints:** The application defines specific API endpoints for communication between the frontend and backend. These endpoints handle data requests and responses.

Data items and messages flow between these components, with the frontend making API requests to the backend, which in turn interacts with the database. Services like model training in AWS may involve data exchange for training data and model outputs. Proper API protocols and data structures are defined to facilitate these communications. Data sharing mechanisms within the application are implemented using RESTful API endpoints for CRUD operations.

3.4 Communications Interfaces

1. **Web Browser Communication:** Standard HTTPS protocols are used for client-server interactions. Data is transmitted in JSON format for structured communication between the client (web browser) and the server.

2. **Real-Time Chat Communication:** WebSocket are used for real-time chat communication. Chat messages are formatted as text messages, allowing for real-time, conversational interactions.

3. **Software Notifications:** SMTP (Simple Mail Transfer Protocol) is used to send email messages. Message Formatting: Email messages are formatted in HTML and plain text to deliver user-friendly notifications.

4. **Internal Communication:** Baymax interfaces with machine learning models, cloud services, and external systems through APIs for data exchange. APIs use standard HTP communication protocols. Data exchanged via APIs is typically formatted in JSON for structured information sharing.

5. **Data Security and Encryption:** Data transfer is secured using Transport Layer Security (TLS) or SSL (Secure Sockets Layer) encryption protocols. User authentication and authorization for secure communication are handled using OAuth or similar authentication mechanisms.

Local data transfer rates will vary based on the user's internet connection.

4. System Features

4.1 Registration Feature

1. Description and Priority

This feature allows users to create an account by providing their personal information, such as name, email, password etc. It is a fundamental aspect of the system and is of high priority.

2. Stimulus/Response Sequences

1. *The user navigates to the registration page.*
2. *The system displays the registration form.*
3. *The user enters their name, email, password and other fields as required.*
4. *The system validates the input data and, if valid, proceeds with registration.*
5. *If the registration is successful, the system notifies the user and creates an account for the user.*
6. *If there are errors in the registration process (e.g., duplicate email), the system displays appropriate error messages.*

3. Functional Requirements

REQ-1: The system must validate the user's email to ensure it's in a proper format.
REQ-2: The system must check if the entered email is not already associated with an existing account.
REQ-3: Passwords must meet security requirements (e.g., minimum length, complexity).
REQ-4: The system should send a confirmation email for account activation.

REQ-5: If the registration is not successful, the system should provide clear and informative error messages.

4.2 Login Feature

1. Description and Priority

This feature allows registered users to log in to their accounts by providing their email and password. It is a fundamental aspect of the system and is of high priority.

2. Stimulus/Response Sequences

1. *The user navigates to the login page.*
2. *The system displays the login form.*
3. *The user enters their email and password.*
4. *The system validates the input data and, if valid, proceeds with login.*
5. *If the login is successful, the system redirects the user to their account dashboard.*
6. *If there are errors in the login process (e.g., incorrect password), the system displays appropriate error messages.*

3. Functional Requirements

REQ-1: The system must validate the user's email to ensure it's in a proper format.
REQ-2: The system must check if the entered email is associated with an existing account.
REQ-3: The system must verify that the entered password matches the stored password for the associated account.
REQ-4: If the login is not successful, the system should provide clear and informative error messages.

4.3 Admin Dashboard Feature

1. Description and Priority

The Admin Dashboard feature provides administrators with the capability to manage various aspects of the system. It includes sub-features for managing doctor and patient accounts, appointments, payments, and overall application data. This feature is of high priority as it is critical for system management.

2. Sub-Feature: Manage Accounts and Appointments of Doctors and Patients

4.3.2.1 Description and Priority

This sub-feature enables administrators to create, delete, update, and view doctor and patient accounts. Additionally, administrators can manage appointments related to these accounts. It is of high priority.

4.3.2.2 Stimulus/Response Sequences

1. *The administrator logs in to the system and accesses the Admin Dashboard.*
2. *If the login is a success, the system presents options to manage doctor and patient accounts.*
3. *The administrator selects an action (e.g., create, delete, update, view).*
4. *The system prompts the administrator to provide necessary details and performs the requested action.*

3. Functional Requirements

REQ-1: The system should allow administrators to create new doctor and patient accounts, including their personal information and credentials.
REQ-2: Administrators should be able to update the information of doctor and patient accounts.
REQ-3: Administrators should be able to delete doctor and patient accounts.
REQ-4: The system should provide the ability to view details of doctor and patient accounts.

4. Sub-Feature: Manage Appointments

4.3.2.3 Description and Priority

This sub-feature allows administrators to view, create, and delete appointments. It is a vital part of the Admin Dashboard feature and is of high priority.

4.3.2.4 Stimulus/Response Sequences

1. *The administrator logs in to the system and accesses the Admin Dashboard.*
2. *If the login is successful, the system provides options to manage any appointments.*
3. *The administrator selects an action, such as viewing any appointments or creating a new appointment.*
4. *The system displays the relevant data or prompts the administrator to enter appointment details.*
5. *The administrator can delete any appointments as needed.*

4.3.2.5 Functional Requirements

REQ-1: The system should allow administrators to view a list of all appointments, including details like date, time, and associated doctors and patients.
REQ-2: Administrators should be able to create new appointments, specifying the date and time, as well as the involved doctor and patient.
REQ-3: The system should provide an option to delete any appointments.

4.4 Doctor Dashboard Feature

1. Description and Priority

The Doctor Dashboard feature provides doctors with the capability to manage various aspects of their practice. It includes sub-features for managing appointments, patient data, profile updates,

patient communication, and maintaining their availability timetable. This feature is of high priority for effective patient care.

2. Sub-Feature: Manage Appointments

4.4.2.1 Description and Priority

This sub-feature enables doctors to view, accept, schedule, and reschedule appointments, both upcoming and late ones. It is essential for managing their patient appointments and is of high priority.

4.4.2.2 Stimulus/Response Sequences

1. *The doctor logs in to the system and accesses the Doctor Dashboard.*
2. *If the login is successful, the system displays options to manage their appointments.*
3. *The doctor can view a list of their own upcoming and late appointments.*
4. *The doctor can accept, schedule, or reschedule their appointments by selecting the desired action.*
5. *The system provides tools for managing their appointment details.*

4.4.2.3 Functional Requirements

REQ-1: The system should allow doctors to view a list of their upcoming and late appointments, providing essential details.
REQ-2: Doctors should be able to accept and schedule their appointments.
REQ-3: Doctors should have the option to reschedule their appointments, with the system providing necessary tools for this.
REQ-4: The system should record and display their appointment details and updates.

3. Sub-Feature: Patient Data Management

4.4.3.1 Description and Priority

This sub-feature empowers doctors to view patient data, including profiles, medical history, current symptoms, disease predictions, timely vital readings, and predictions from backend ML based models. It is crucial for providing effective patient care and is of high priority.

4.4.3.2 Stimulus/Response Sequences

1. *The doctor logs in to the system and accesses the Doctor Dashboard.*
2. *If the login is successful, the system provides options to view patient data.*
3. *The doctor selects a patient and can access their profile, medical history, current symptoms, disease predictions, live vital readings, and backend model predictions.*

4.4.3.3 Functional Requirements

REQ-1: Doctors should be able to access detailed patient profiles.
REQ-2: The system should provide access to a patient's medical history.
REQ-4: Doctors should be able to access and view disease predictions for the patient based on API data.
REQ-5: Doctors should be able to access and buffer vital readings and backend model predictions for the patient.

4. Sub-Feature: Profile and Password Updates

4.4.4.1 Description and Priority

This sub-feature allows doctors to update their profiles and passwords. It is a standard feature for account management and is of medium priority.

4.4.4.2 Stimulus/Response Sequences

1. *The doctor logs in to the system and accesses the Doctor Dashboard.*
2. *If the login is successful, the system provides options to update the doctor's profile and password.*
3. *The doctor can make necessary changes to their profile information and update their password.*

4.4.4.3 Functional Requirements

REQ-1: Doctors should be able to update their profile information, such as contact details and specialty.
REQ-2: Doctors should have the capability to change their account password.
REQ-3: The system should send a confirmation email for change of password.

5. Sub-Feature: Communication with patients and other doctors

4.4.5.1 Description and Priority

This sub-feature enables doctors to communicate with patients and doctors via a chat feature. It is a valuable tool for doctor-patient communication and is of medium priority.

4.4.5.2 Stimulus/Response Sequences

1. *The doctor logs in to the system and accesses the Doctor Dashboard.*
2. *If the login is successful, the system provides options to initiate and respond to patient chats or doctor chats.*
3. *The doctor can send messages and receive messages from patients and doctors through the system.*

4.4.5.3 Functional Requirements

REQ-1: The system should provide a chat feature for doctors to communicate with patients and doctors securely.
REQ-2: Doctors should be able to initiate and respond to patient chats.

6. Sub-Feature: Availability Timetable

4.4.6.1 Description and Priority

This sub-feature allows doctors to maintain a timetable that records their availability. It is important for scheduling appointments and is of medium priority.

4.4.6.2 Stimulus/Response Sequences

1. *The doctor logs in to the system and accesses the Doctor Dashboard.*

2. *If the login is successful, the system provides options to manage the doctor's availability timetable.*
3. *The doctor can set and update their availability for appointments.*

4.4.6.3 Functional Requirements

REQ-1: Doctors should be able to set and update their availability timetable.
REQ-2: The system should display the doctor's availability to assist in appointment scheduling.

4.5 Patient Dashboard Feature

1. Description and Priority

The Patient Dashboard feature provides patients with tools to manage their healthcare and interact with healthcare providers. It includes sub-features for managing appointments, viewing doctor profiles, updating their own patient profile, doctor communication, and vital readings. This feature is of high priority for improving patient care and experience.

2. Sub-Feature: Manage Appointments

4.5.2.1 Description and Priority

This sub-feature enables patients to view, request, and reschedule appointments. It is essential for patients to manage their healthcare appointments and is of high priority.

4.5.2.2 Stimulus/Response Sequences

1. *The patient logs in to the system and accesses the Patient Dashboard.*
2. *If the login is successful, the system provides options to manage their own appointments.*
3. *The patient can view their own upcoming appointments.*
4. *The patient can request a new appointment or request to reschedule an existing one.*

4.5.2.3 Functional Requirements

REQ-1: The system should allow patients to view their upcoming appointments.
REQ-2: Patients should be able to request new appointments.
REQ-3: Patients should be able to request rescheduling of their existing appointments.
REQ-4: The system should notify patients of their appointment confirmations and changes.

3. Sub-Feature: Doctor Profiles

4.5.3.1 Description and Priority

This sub-feature allows patients to view information about doctors, including specialization, education, and more. It is important for patients to make informed choices about their healthcare providers and is of medium priority.

4.5.3.2 Stimulus/Response Sequences

1. *The patient logs in to the system and accesses the Patient Dashboard.*

2. *If the login is successful, the system provides options to view doctor data.*
3. *The patient can select a doctor and access their details, including specialization, education, and more.*

4.5.3.3 Functional Requirements

REQ-1: The system should provide a list of available doctors.
REQ-2: Patients should be able to access detailed profile information about each doctor.
REQ-3: Doctor details should include specialization, education, and any other relevant information.

4. Sub-Feature: Profile and Password Updates

4.5.4.1 Description and Priority

This sub-feature allows patients to update their profiles and passwords. It is a standard feature for account management and is of medium priority.

4.5.4.2 Stimulus/Response Sequences

1. *The patient logs in to the system and accesses the Patient Dashboard.*
2. *If the login is successful, the system provides options to update the patient's profile and password.*
3. *The patient can make necessary changes to their profile information and update their password.*

4.5.4.3 Functional Requirements

REQ-1: Patients should be able to update their profile information, including contact details.
REQ-2: Patients should have the capability to change their account password.
REQ-3: The system should send a confirmation email for change of password.

5. Sub-Feature: Doctor Communication

4.5.5.1 Description and Priority

This sub-feature enables patients to communicate with their doctors via a chat feature. It is a valuable tool for doctor-patient communication and is of high priority.

4.5.5.2 Stimulus/Response Sequences

1. *The patient logs in to the system and accesses the Patient Dashboard.*
2. *If the login is successful, the system provides options to initiate and respond to doctor chats.*
3. *The patient can send messages and receive messages from doctors through the system.*

4.5.5.3 Functional Requirements

REQ-1: The system should provide a chat feature for patients to communicate with their doctors securely.
REQ-2: Patients should be able to initiate and respond to doctor chats.

6. Sub-Feature: Vital Readings

4.5.6.1 Description and Priority

This sub-feature allows patients to view readings from medical devices monitoring their vitals, which are sent to the backend for analysis using a trained model to predict clinical conditions. It is essential for patients' healthcare management and is of high priority.

4.5.6.2 Stimulus/Response Sequences

1. *The patient logs in to the system and accesses the Patient Dashboard.*
2. *If the login is successful, the system provides options to view vital readings.*
3. *The patient can view vital readings from medical devices, which are transmitted to the backend for analysis.*

4.5.6.3 Functional Requirements

REQ-1: The system should collect and display vital readings from medical devices, including data on the patient's vitals (e.g., heart rate, blood pressure).
REQ-2: Vital readings collected from the patient's medical kit should be transmitted to the backend system.
REQ-3: The backend system should use a trained model to analyze the vital readings and predict clinical events or abnormalities.
REQ-4: The system should facilitate the sharing of patient-provided vital data with healthcare providers (doctors) for diagnosis and treatment.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

1 The system must ensure timely data transmission and display vital readings from the medical monitoring kit to the web portal.

2 The system should have a maximum allowable delay of 8 seconds for transmitting vital readings to the web portal.

3 The web app should be capable of handling concurrent connections from multiple users without performance degradation.

4 The application must maintain constant online availability to ensure that patients can promptly reach doctors whenever they want.

5.2 Safety Requirements

1 The medical monitoring kit must be designed to prevent electrical hazards and ensure patient safety during use.

2 The web app should provide clear warnings or alerts in case of abnormal vital readings, enabling patients and doctors to take appropriate actions.

3 Compliance with relevant medical device safety standards and regulations is mandatory.

4 The medical monitoring kit must obtain and maintain certifications, such as FDA approval, to ensure its safety for medical use.

5.3 Security Requirements

1 Patients' consent to share their medical data should be mandatory.

2 Patient data collected by the monitoring kit and web app must be encrypted and stored securely.

3 Access to patient data should be restricted to authorized healthcare professionals.

4 Safety, confidentiality, and integrity of user data must be ensured.

5.4 Software Quality Attributes

1. Reliability

The system must have a system uptime of at least 99 %. Data integrity and reliability are essential to avoid errors in vital readings and patient records.

2. Usability

The web app should have a user-friendly interface that enables patients and healthcare providers to navigate and interact with ease.

5.5 Business Rules

The application is intended for use exclusively within Pakistan, and registered user hospitals and doctors must have physical locations within the country. We may consider the possibility of expanding our target area in the future.

6. Other Requirements

To enhance the platform's centralization, it is essential to onboard doctors, patients, hospitals, and clinics.