Project Report
International Project Management
[April-May 2023]

My Mobile Doctor

SRH Hochschule Heidelberg
Faculty for Information, Media & Design
Degree M.SC. Applied Computer Science

Under the guidance of:
Prof. Dr. Gerd Moeckel
Dr. Josef Hermanns

Authors:
Akshata Umakant Naik

Ankith Reddy Kanapuram

Jordan Dsouza

Mahesh Nidugala

Malvina Vianna Dsouza

CONTENTS

1. LIST OF FIGURES 3
2. LIST OF TABLES 5
3. INTRODUCTION 6
3.1 Personas and solutions 7
3.2 Project Overview 8
3.3 Task Description 9
3.4 Project Environment 9
3.5 Project Description 11
3.6 Target State 12
3.7 Team Roles and Responsibility 13
4. MOSCOW ANALYSIS & MILESTONES 14
4.1 Moscow Analysis 14
4.2 Milestones 15
5. REQUIREMENT ANALYSIS 17
5.1 Functional Requirements 17
5.2 Technical Requirements 17
5.3 Operational Requirements 18
5.4 Customer Requirements 19
5.5 Infrastructural Requirements 19
6. STAKEHOLDER & RISK ANALYSIS 20
6.1 Stakeholder analysis 20
6.2 Risk Analysis 21
7. PROJECT METHODOLOGY & PLANNING 23
7.1 V-Model 23
7.2 Agile Scrum Process 25
7.3 Epics and User Stories 26
7.3 Testcases 40
7.4 Sprint Planning 44
8. TECHNOLOGY AND APPLICATION FLOW 47
8.1 Technical Architecture Frontend 47
8.2 Technical Architecture Backend 48
8.3 Application Flow 49
8.4 Use Case Diagrams 50
9. USER INTERFACE AND DESIGNS 52
9.1 User Interface for Mobile Application 52

9.2 User Interface for Web Application 57
10. FUTURE WORK 64
11. SUMMARY 65
11.1 Project Features 65
11.2 Total Budget 65
11.3 Project Date 65
11.4 Key Risks 65
12. REFERENCES 66

1. LIST OF FIGURES

TITLE	PAGE NO.
Fig 1 Number of Patients booked in 2014	6
Fig 2 Target State Sequence Diagram	12
Fig 3 Moscow chart	14
Fig 4 Milestone Graph	16
Fig 5 Stakeholder Map	21
Fig 6 Risk Matrix	22
Fig 7 V - model	23
Fig 8 Scrum ceremonies	25
Fig 9 Gantt Chart	46
Fig 10 Frontend Architecture	47
Fig 11 Backend Architecture	48
Fig 12 My Mobile Doctor Flow chart	49
Fig 13 Patients' Use Case Diagram	50
Fig 14 Doctor's Use Case Diagram	51
Fig 15 Account Selection Mobile UI	52
Fig 16 Patient Login Mobile UI	52
Fig 17 Patient Registration Mobile UI	52
Fig 18 Insurance Upload Mobile UI	52
Fig 19 Chatbot prompt Mobile UI	53
Fig 20 Chatbot Mobile UI	53
Fig 21 Chatbot Mobile UI	53
Fig 22 Instant Messaging with Doctor Mobile UI	53
Fig 23 Home Page Mobile UI	54
Fig 24 Side bar Mobile UI	54
Fig 25 Medicine Order Mobile UI	55

Fig 26 Doctor Login Mobile UI 55

Fig 27 Doctor's Dashboard Mobile UI 56

Fig 28 Doctor's Dashboard Mobile UI 56

Fig 29 Report Upload Mobile UI 57

Fig 30 Account selection Web UI 57

Fig 31 User Login Web UI 58

Fig 32 User Registration Web UI 58

Fig 33 Insurance Upload Web UI 59

Fig 34 Home page Web UI 59

Fig 35 Chatbot Web UI 60

Fig 36 Chatbot Web UI 60

Fig 37 Instant Messaging with doctor Web UI 61

Fig 38 Doctor Login Web UI 61

Fig 39 Doctor Registration Web UI 62

Fig 40 Doctors' dashboard Web UI 62

Fig 41 Medicine Order Web UI 63

Fig 42 Prototype of Smart Watch 64

2. LIST OF TABLES

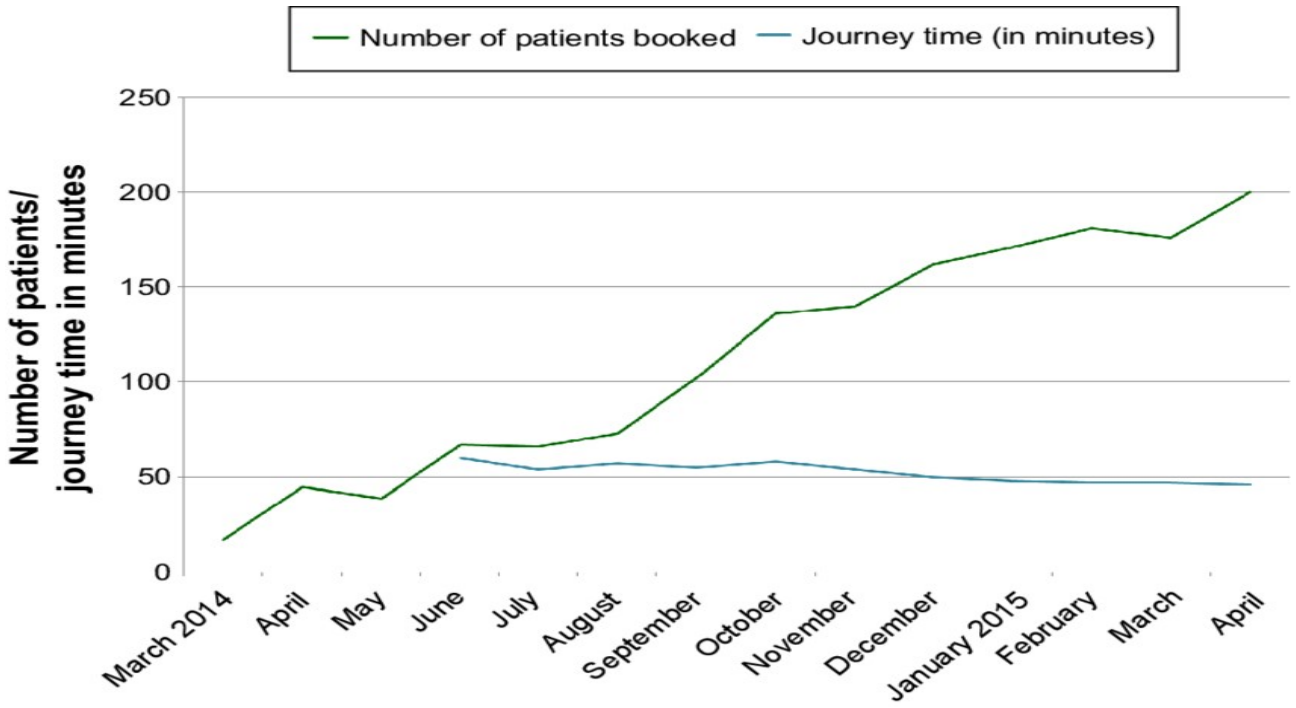
TITLE	PAGE NO.
Table 1 Personas & Solutions	7
Table 2 Project Overview	8
Table 3 Project Environments	9
Table 4 Team Roles and Responsibility	13
Table 5 Milestones	15
Table 6 Technical Requirements	17
Table 7 Operational Requirements	18
Table 8 Infrastructural Requirements	19
Table 9 Risk Analysis	21

3. INTRODUCTION

Health care is one of the most important aspects of any nation as people's well-being is essential and a strong foundation for a healthy nation in general. Patients with critical health issues require urgent and intensive medical care to stabilize their condition and prevent further deterioration. Critical health conditions are often treated using a multidisciplinary approach, with a team of medical experts cooperating to manage the patient's condition. Critically ill patients frequently need to be hospitalized so they may get round-the-clock medical attention and monitoring.

Hospitals and clinic can improve their appointment system by prioritizing the appointments based on the urgency of a patient's condition. This can be done by implementing triage systems and prioritizing patients who require urgent care. To ease the appointment system an online platform is required that can aid the patients with online treatment if a visit to a doctor is not necessary and alerts a hospital/clinic if a patient requires immediate attention and care. Hence the application "My Mobile Doctor" is the solution that allows patients to consult the doctor from his/her mobile, book an appointment to visit the hospital or clinic if it is critical, opt for emergency door-to-service, order medicines online, and manage test or lab reports online.

Fig 1: Number of Patients booked in 2014



Source: https://www.researchgate.net/

3.1 Personas and solutions

Table 1: Personas & Solutions

Personas	Description and Solution
Patient with minor health issue staying in rural areas. **Example:** Person: Mathew Age: 45 Occupation: Accountant Health Issue: Headache	**Description:** A patient named Mathew with a minor headache who is residing in the countryside needs assistant from the doctor. But he is not able to travel way to the city and take an appointment for a consultation. **Solution:** Instead of the patient traveling to the city to make an appointment with a doctor, a doctor consultation application that is available from mobile or desktop with an AI-trained chatbot to grasp the patient's concern and deliver a prescription would address the problem.
Patient with critical health issue staying in rural areas. **Example:** Person: John Age: 48 Occupation: Engineer Health Issue: Diabetes	**Description:** A patient named John who is residing in the countryside and suffering from diabetes needs assistant from the doctor. But he is not able to travel way to the city and take an appointment for a consultation. **Solution:** Instead of the patient traveling to the city to make an appointment with a doctor, a doctor consultation application that is available on mobile or desktop with instant messaging features with doctors would address the problem. Doctors decide if the person needs to be at the clinic for further consultation.
Elderly patient in need of emergency care who is computer/mobile illiterate. **Example:** Person: David Age: 70 Health Issue: Heart Related	**Description:** A device is required to notify David, a patient who is elderly and untrained in mobile and desktop applications, that there is an emergency. **Solution:** A smart watch with an integrated SOS feature would address the problem
Busy doctor who struggles to manage the many appointments they receive each day **Example:** Person: Dr.Tim Age: 40 Occupation: Doctor	**Description:** To make scheduling appointments easier and to limit them to situations where the patient's health is in grave danger, a desktop or mobile application is needed. **Solution:** A chatbot with AI training can understand a patient's worry and, if it can identify the issue, offer medication. The patient can communicate with the doctor as a second level of assistance. A doctor will determine if the patient needs to go to a hospital or clinic.

A patient who lives in the country and requires medication yet there is no drugstore close by. **Example:** Person: Rahul Age: 40 Occupation: Software Engineer	**Description:** Rahul lives in a rural area where there is not a pharmacy close by. He has a long way to go to purchase the medications. **Solution:** The issue might be solved with the help of a medication ordering system included in the doctor consultation app that links to local pharmacies to order the medications and delivers them to patients' doorsteps.

Source: Own Representation

3.2 Project Overview

Table 2: Project Overview

PROJECT: MY MOBILE DOCTOR

DESCRIPTION: A mobile and web service called "**My Mobile Doctor**" offers real physicians and AI-trained chatbots to help patients with their problems. This aids hospitals and clinics in streamlining the appointment process and doing away with lengthy wait times for doctor appointments.

BUDGET PLANNING

- **Resource Budget**

Resources	Working Hours	Cost
Backend Developers	200 Hours (40 Hours a week for 5 months)	0.4 million Euros
Frontend Developers	200Hours (40 Hours a week for 5 months)	0.8 million Euros
UI/UX Designers	200 Hours (40 Hours a week for 5 months)	0.5 million Euros
Testers	200Hours (40 Hours a week for 5 months)	0.1 million Euros

- **Software and Hardware licence**: For 4 years 0.5 million Euros
- **Smart Watch:** 0.8 million Euros

Project Manager: Prof. Dr. Gred Moeckel **Project Sponsor: Dr. Josef Hermans**
Approvals: ➢ Project Approvals ➢ Requirement Approvals ➢ Code Acceptance ➢ QA Acceptance

Source: Own Representation

3.3 Task Description

The main task is to build mobile and web-based applications with the AI-trained chatbot that acts as the first level of contact and provides immediate consulting for minor issues to the patients. The chat with the live doctors should also be implemented to provide consulting for patients' critical issues.

In addition to this application, it should also have features like a medicine ordering system, emergency services delivered to your doorstep, and the ability to view and download test reports. For this purpose, project management concepts must also be included with methodology, network plan, risk analysis, system architecture, UI/UX designs, and stakeholders

3.4 Project Environment

Table 3: Project Environments

Mobile and Desktop Devices	The "My Mobile Doctor" application is going to be developed for mobile and web devices, so it is important to consider the different kinds of mobile and desktop devices and operating systems that it will need to run on.
Internet Connectivity	In order to use the application, you will need a good internet connection, because you will need to consult doctors online, make appointments, and get notifications.
Security and privacy	In order to protect sensitive patient information, it is important that the application has appropriate security measures in place. • User authentication and access control: To protect sensitive patient data, applications should require users to authenticate themselves, like through passwords or biometrics.

	• Data encryption: Data containing sensitive information, like medical histories and PII, should be encrypted both in transit and at rest. • Regular data backups: To guarantee that data can be restored in the event of a security breach or other data loss incident, patient data should be regularly backed up. • Secure coding practices: To avoid typical security vulnerabilities like SQL injection attacks and cross-site scripting (XSS) assaults, developers should adhere to secure coding principles. • Regular security assessments: To find potential security risks and vulnerabilities, the application should go through routine security assessments including penetration testing and vulnerability scanning. • Compliance with regulatory requirements: The application must follow medical regulations in order to secure patient data. • Incident response plan: An incident response strategy should be in place for the application to react swiftly and successfully to security incidents like data breaches.
Medical databases	To access patient data and medical histories, the application might need to establish a connection with medical databases like electronic health records (EHRs).
User interface	Both doctors and patients should be able to easily utilize the application's user-friendly design.
Backend infrastructures	To perform properly, the application will need backend infrastructure like servers and databases.
Development Tools	To create and test the application, the development team will need development tools including testing frameworks and integrated development environments (IDEs) like Visual Studio for frontend and STS for Backend.
Collaboration Tools	To collaborate successfully and efficiently, the development team may employ tools like project management software and communication platforms like Jira, and Confluence.
User Testing	To make sure that it satisfies the needs of both patients and doctors and is operating as intended, the application will need to be put through user testing.
Political Aspects	The project must adhere to all laws, rules, and regulations of the area or nation in which it is being built. The project's developers are responsible for making sure it complies with regulations on security, privacy, and data accessibility.

Source: Own Representation

3.5 Project Description

Even though Germany offers one of the best health care facilities to its citizens, the appointment system in Germany is quite debatable as well because even though priority appointments are reserved for masses suffering from severe ailments, people even with a slight cold weigh in to claim a priority appointment. Another reason for the burden on the appointment system is the shortage of medical professionals in certain regions of the country. This has resulted in high demand for appointments, with many patients having to wait several weeks or even months to see a doctor.

The project's aim is to provide a mobile and web-based platform that allows patients to consult with healthcare providers remotely. The application's main objective is to offer patients with fundamental health problems first-level consulting from an AI-trained chatbot, which also determines whether the patient's problem is critical. Additionally, chatbots should aid in disease early detection and offer information on how diseases develop. If the condition is determined to be critical and the chatbot is unable to identify it, the application should connect users to a human doctor who is available for second-level consultation and can advise whether the patient must visit the hospital and take urgent medication.

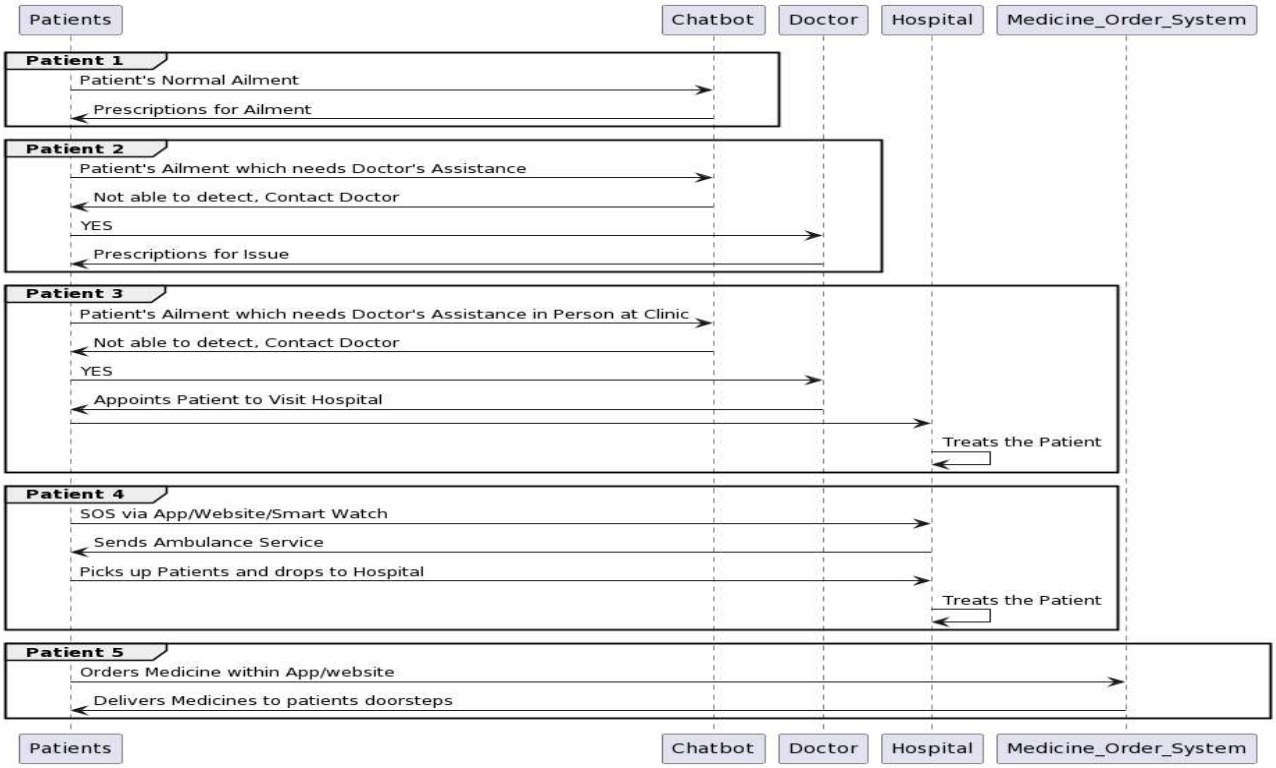
The chatbots should be trained using natural language processing. To comprehend, interpret, and produce human language, NLP should incorporate text classification, speech recognition, and text generation. By utilizing methods like part-of-speech tagging, named entity identification, and sentiment analysis, NLP can be applied to improve the chatbot's capacity to comprehend natural language. Using these methods, the chatbot can pick up on crucial details from the patient's interaction, including symptoms, and medical history. The chatbot's capacity to comprehend patients' problems will grow as it engages with more patients and gains knowledge from such encounters. Patients can provide feedback on the efficacy and accuracy of the chatbot's responses by using feedback mechanisms that can be incorporated within the chatbot.

In addition to these, the app also has a feature that allows users to order medications by connecting to nearby pharmacies. Users of the apps can read and download the test report using a function that is included in the software. In the application, the emergency service is also included, and in an emergency, the patient can select it to send a signal to a nearby hospital. The hospital may locate the user's geolocation and dispatch an ambulance to the person's doorstep. The application also must consider elderly people who are not familiar with the application. For this purpose, the wristwatch with the integrated SOS feature, monitoring heart rate, and sugar level will also be provided as part of the "My Mobile Application" development. The elderly can opt for a wristwatch that can be used in case of an emergency that sends signals to a nearby hospital for emergency service.

3.6 Target State

The main objective of the web and mobile application "My Mobile Doctor" is to enable patients to receive in-person diagnosis and treatment of minor health conditions from our AI chatbot trained in Natural Language Processing in the comfort of their own homes. The processing of appointments, as previously said, takes a considerable amount of time, and it could take a patient anywhere from three days to a week to hear back regarding their request for an appointment. "My Mobile Doctor" will be a boon to healthcare and hospitality, helping to eventually end this waiting game predicament. The chatbot directs a patient to a network-registered doctor if the patient is in urgent need of treatment and the chatbot is unable to provide treatment. When a patient needs intensive care treatment, "My Mobile Doctor" will be able to help in such cases as well. In this situation, the app will keep an eye on the patient for any irregularities and, in the event of an emergency, it will alert the local hospital or clinic to dispatch an ambulance for further care either of them.

Fig 2: Target State Sequence Diagram



Source: Own Representation

3.7 Team Roles and Responsibility

Table 4: Team Roles and Responsibility

Team	Roles	Chapters on Report
Mahesh Nidugala Malvina Dsouza Akshata Naik	UI/UX Designing	User Interface and Designs
Mahesh Nidugala	Requirement Analysis	Functional Requirements, Technical Requirements, Operational Requirements, Customer Requirements, Infrastructural Requirements
Mahesh Nidugala Jordan Dsouza Akshata Naik Malvina Dsouza Ankith Reddy	Project Analysis	Project Overview and Description, Personas, Target States, Moscow Analysis and Milestones.
Ankith Reddy Malvina Dsouza Akshata Naik Jordan Dsouza	Risk and Stakeholder analysis	Stakeholder and Risk Analysis
Mahesh Nidugala	Project Planning, Gantt chart, User stories and Test Scenarios	Project methodology and planning
Jordan Dsouza Akshata Naik	Technical Architect	Technology and Application Flow

Source: Own Representation

4. MOSCOW ANALYSIS & MILESTONES

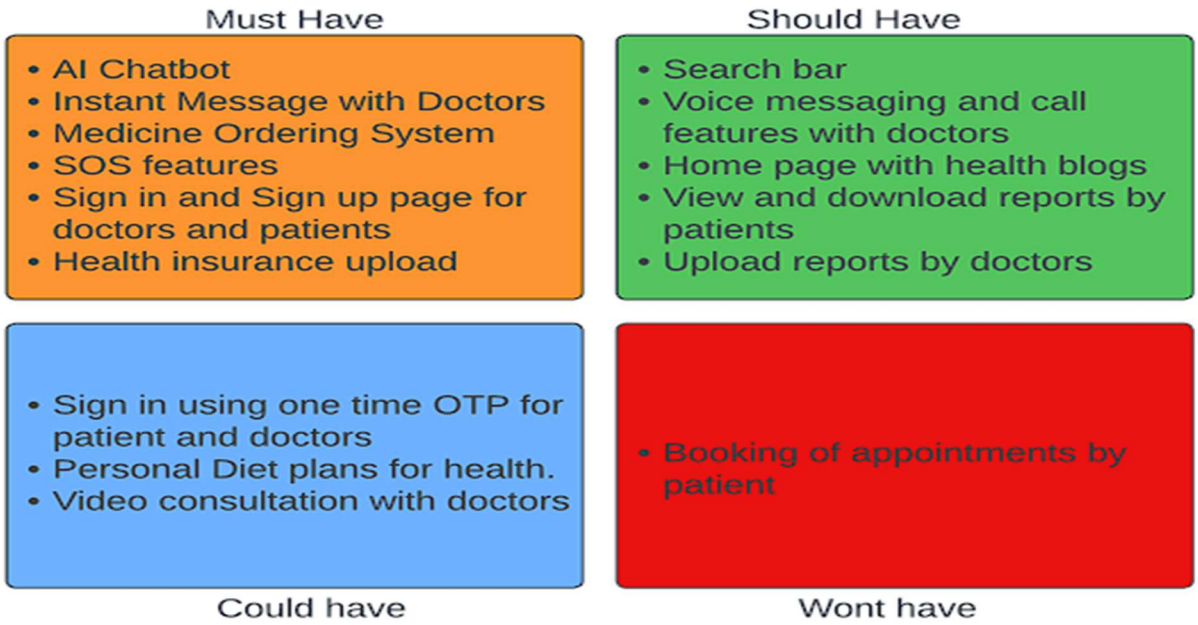
4.1 Moscow Analysis

In project management classifying needs according to significance is called the MOSCOW analysis. It represents:

- Must-haves: These specifications must be present in the final delivery for the project to be successful.
- Should-haves: These are specifications that are necessary but not necessary. If time and resources allow, they can be added.
- Could-haves: These are features that, although desirable, are not essential. If extra time or money is available, they can be added.
- Won't-haves: These are specifications that are not part of the project's scope and will not be met.

Below figure shows the Moscow chart for "**My Mobile Doctor"**

Fig 3: Moscow chart



Source: Own Representation

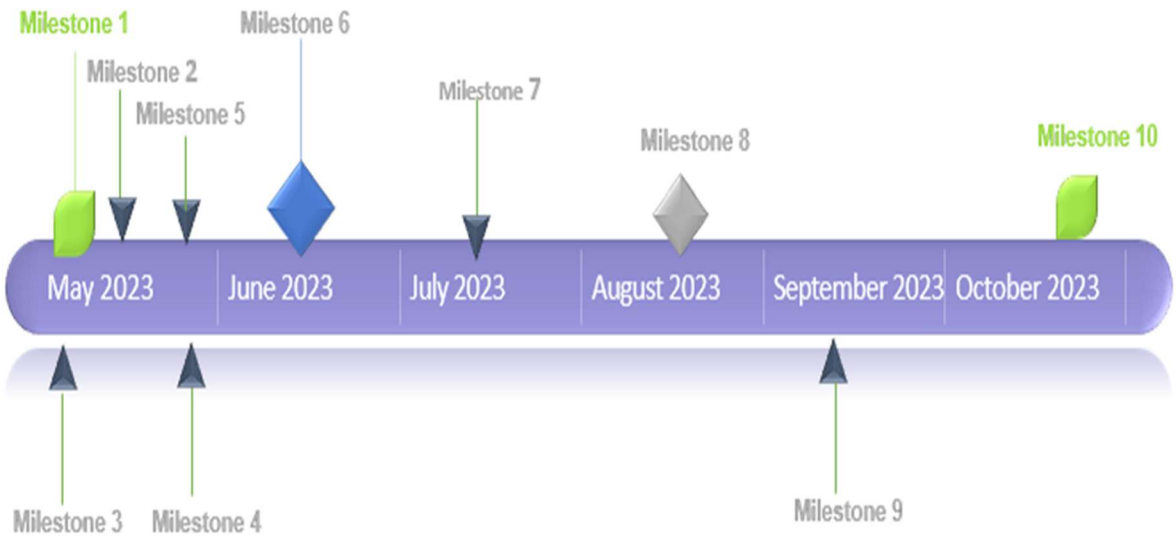
4.2 Milestones

Table 5: Milestones

Milestones	Description
1. Software Setup	Setup the required software for development. **Month**: May 2023
2. Requirement gathering	Analysing the required functionality for application and create user stories for each requirement. **Month**: May 2023
3. Risk and Stakeholder Analysis	Analysing the risk and stakeholders involved in the application. **Month**: May 2023
4. Test scenarios	Defining the test scenarios for each functionality **Month**: May 2023
5. Login and Registration Page development for user and testing	Developing the login and registration page for user with backend APIS and unit test the functionality **Month**: May 2023
6. Login and Registration page for doctor, AI chatbot development, Instant messaging features with doctor and testing	Developing the login and registration page for doctor with backend APIS, Implementing chatbot functionality and instant message **Month**: June 2023
7. Doctor Account and Patients Health Insurance Upload and testing	Developing Doctors dashboard with messaging patient features, test report upload features, notification of patients online with priority, Patients health insurance upload and unit testing **Month**: July 2023
8. SOS, Home Page, Side bar and user testing	Developing SOS features, home page with privacy policy, about us, contact us and side bar with navigations and unit testing **Month**: August 2023
9. Medicine Ordering System and confirmation	Developing medicine ordering page with conformation page and unit testing **Month**: September 2023
10. Regression testing and sanity check and go live	Regression testing and sanity check, Go/ No-go and Go live **Month**: October 2023

Source: Own Representation

Fig 4: Milestone Graph



Source: Own Representation

5. REQUIREMENT ANALYSIS

Requirement analysis is crucial for ensuring that the finished software product satisfies project requirements while also meeting the needs and expectations of end users. It aids in reducing project risks, making sure the project stays on schedule and under budget, and improving customer satisfaction. Following are the requirements for “My Mobile Application”

5.1 Functional Requirements

The “My Mobile Doctor” is a mobile and web application where end users can consult an AI-trained doctor and a live doctor online for medical assistance. Users should be able to create accounts using username, email, phone, and password and securely log in using username and password to consult the doctor online. To activate and use the application the user should upload his/her insurance card details online. Users should be able to message AI-trained chatbots and securely live doctors in a secure manner. This can include text messages and voice messages.

Doctors should be able to access the dashboard with the list of patients’ issues according to priority and prescribe drugs using the app. The application should allow users to view and download their medications. Users should be able to manage their medical history using the app, including previous diagnoses, test outcomes, prescriptions, and allergies. Through the application, physicians should be able to make appointments with patients if the patient’s issue needs more diagnosis and the patient needs to be in person at the hospital.

The software must also enable end customers to order their prescription medications online for home delivery. The app must have SOS functionalities, which send a signal to the local hospital for ambulance services when the user clicks SOS. The "My Mobile Doctor" software development includes a wristwatch elderly that has functions like SOS, blood pressure, and heart rate monitoring.

5.2 Technical Requirements

Table 6: Technical Requirements

Mobile and Web Application Front Development	The application should be developed using web and mobile application development technologies. For Mobile application development, React-native is used and for a web application, React.js is used with material UI for CSS.
Back-end Development	REST APIs should be developed to perform CRUD operations and consumed by the front end. Node.js with GraphQL is used for backend development.
Database	A database management system (DBMS) should be used by the application to store user data, medical records, and other data. MongoDB should be used as the primary database. MongoDB provides security and highly available functionalities.

UI/UX	The application should have a user-friendly, intuitive user interface and navigational layout. Figma is considered in this project for UI/UX designs.
AI Integration	NLP should be used to train the chatbot to understand and interact with the human language
Deployment Tools	Docker and Jenkins are used as deployment platforms for CI/CD.
Methodology	Agile scrum methodology should be implemented for the whole project.
Project Management	Jira should be used for ticket management and Confluence should be used for project management and documentation.
Version Control	GitLab and Bitbucket should be used for code management.

Source: Own Representation

5.3 Operational Requirements

Table 7: Operational Requirements

Availability and reliability	Users should be able to access and rely on the app around-the-clock. This necessitates keeping an eye on and maintaining the databases, networks, and servers for the application.
Scalability	The application should be designed to be scalable, meaning it should be scaled for different desktop and mobile screens.
Performance	The application must function well, with quick loading times and little latency for messaging and replays.
User Support	A support mechanism ought to be in place for the program to assist users in resolving problems and obtaining information.
Data backup and recovery	In the event of a disaster, system failure, or other unforeseen circumstances, the application needs to have a data backup and recovery plan.

Source: Own Representation

5.4 Customer Requirements

Hospitals and clinics should be able to manage their appointment system with ease thanks to the "My Mobile Doctor" application. The hospitals anticipate putting less strain on the appointment system and simply scheduling doctors when necessary. The chatbot ought to be the initial point of contact for clients; if necessary, a client can talk with a doctor, who ought to schedule an emergency visit to the hospital or clinic. Hospitals anticipate avoiding unneeded appointments for minor issues and prioritizing serious ones.

5.5 Infrastructural Requirements

Table 8: Infrastructural Requirements

Servers	For web applications, Nginx should be used as a server. A reverse proxy, load balancer, mail proxy, and HTTP cache are additional functions of the web server Nginx. For mobile applications, AWS is used as a server.
Smartphones	Since the application will be available on smartphones, it should be compatible with different versions of smartphones.
Smartwatch	Features like SOS, heart rate, and blood pressure monitoring must be considered for elderly customers on smartwatches.
RF wireless technology	In order to interact with alarms and transmit signals to hospitals if a customer needs emergency assistance, radio-frequency wireless technology must be considered.

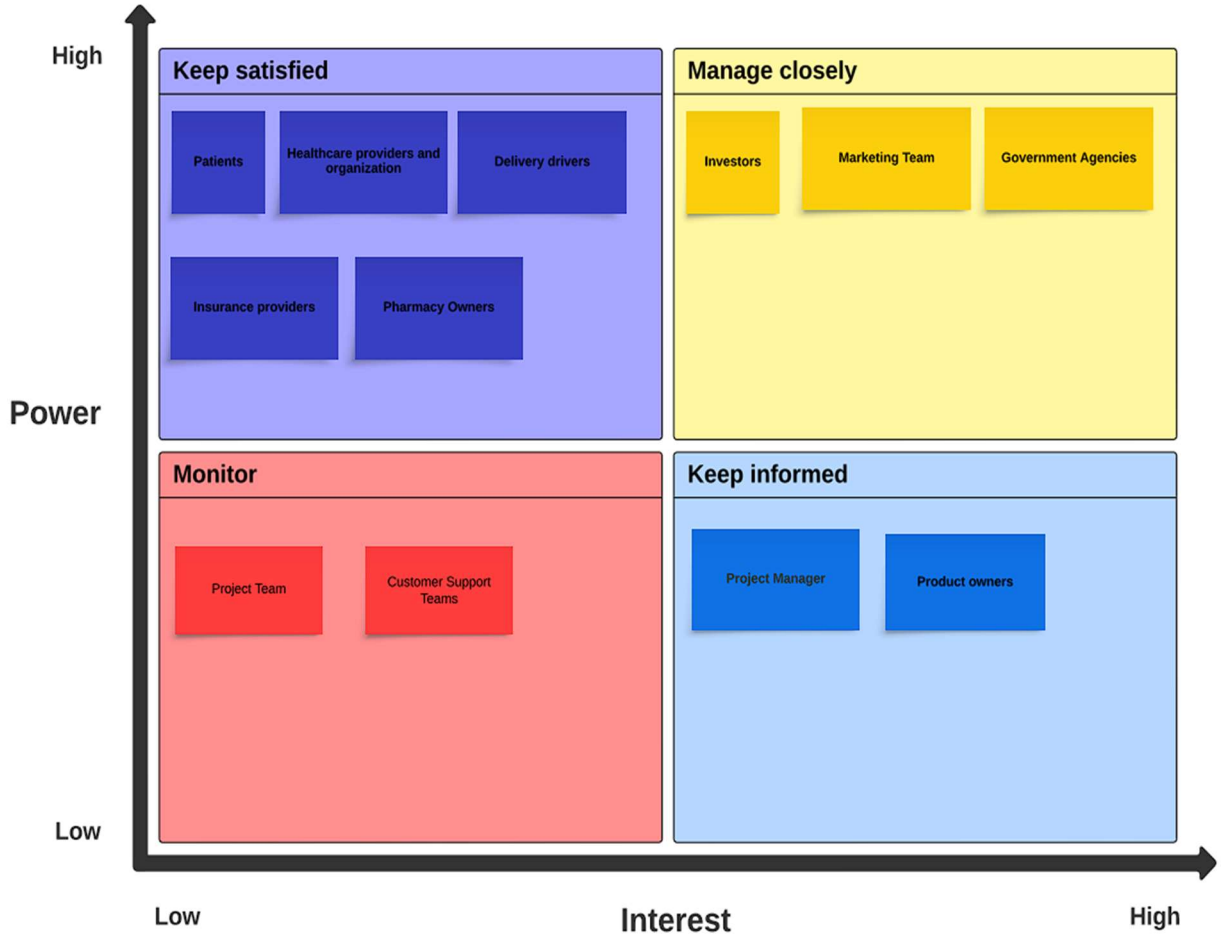
Source: Own Representation

6. STAKEHOLDER & RISK ANALYSIS

6.1 Stakeholder analysis

- **Patients**: They will use the app to speak with physicians, and manage their health information. These are the app's main users.
- **Insurance providers**: Insurance providers are part of the stakeholders as healthcare is free in Germany.
- **Healthcare providers and organization**: The application is used by physicians, hospitals to provide medical treatment and interact with patients.
- **Investors**: If investors think the application has a chance to be profitable, they could be willing to contribute money to its development.
- **Project Manager**: Oversees the whole project team to ensure that the delivery of the project is on schedule.
- **Project Team**: The project team oversees the timely delivery of the project's needs.
- **Marketing Team**: These parties oversee informing potential users, healthcare providers, and other parties about the application and selling them on it.
- **Pharmacy Owners**: They oversee orders, complete prescriptions, and package and transport pharmaceuticals.
- **Delivery drivers**: They oversee timely and effective medication delivery to customers.
- **Product owners**: the product owner oversees making sure the item satisfies the requirements of the client and other stakeholders.
- **Customer Support Teams**: They oversee offering users who have queries or problems with the app technical support and help.
- **Government agencies**: The app and its use may be subject to regulatory supervision by government organizations like the FDA.

Fig 5: Stakeholder Map



Source: Own Representation

6.2 Risk Analysis

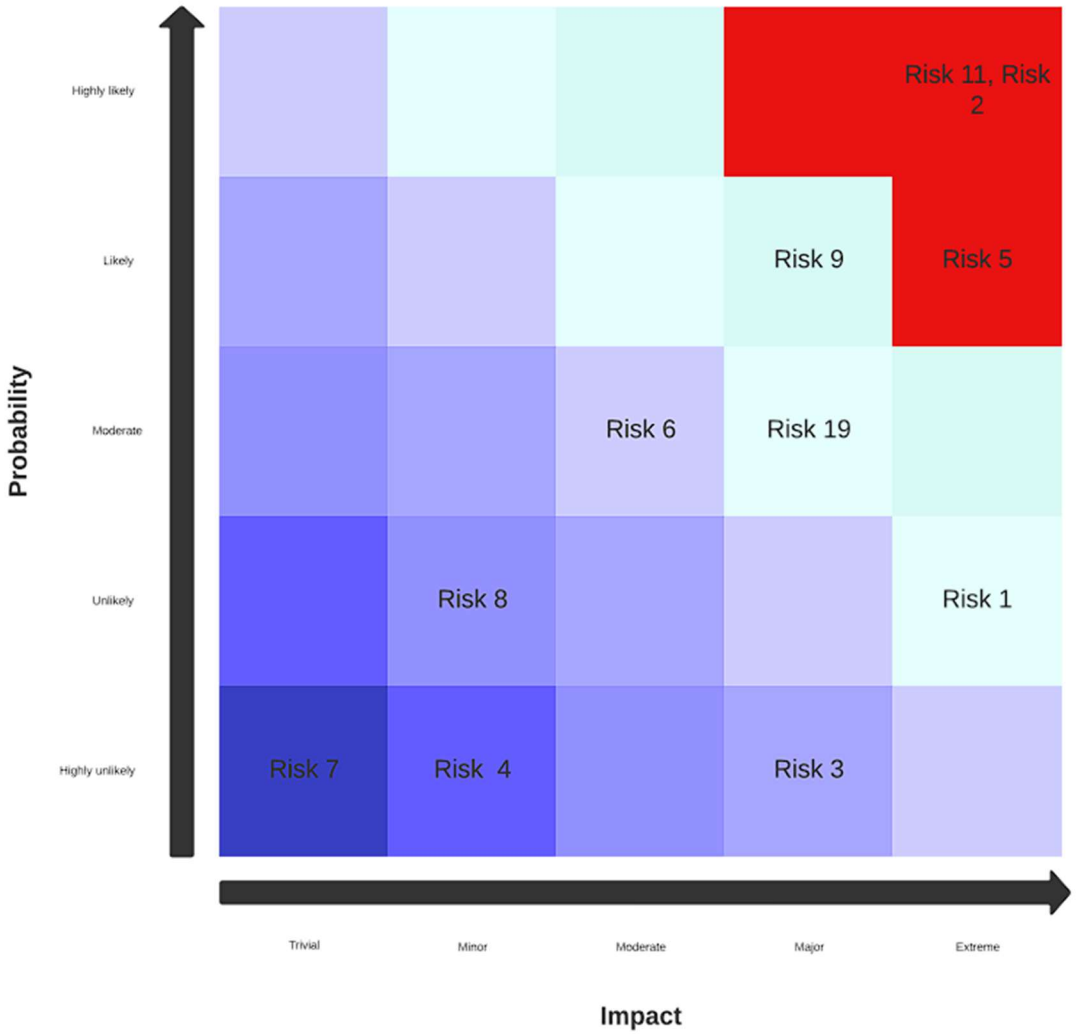
Table 9: Risk Analysis

Risks	Description
1. Availability of Doctors	Shortage of doctors.
2. Consultation of AI chatbot is incorrect	A trained AI chatbot could provide incorrect information or poor advise.
3. Privacy and Transparency	Patient information is not sufficiently safeguarded.
4. Delivery of medicines	The delivery of medications must be timely. Furthermore, orders must be precise.
5. Access to internet	Poor network connection.
6. SOS failure	Medical services or attention may be delayed in this situation.

7. App's ability to handle complex tasks	Programming errors are more likely to harm complex apps.
8. Feedback of customers	Misleading/Negative Feedback.
9. Security breach	Security breaches towards patients' health data as the application is online.
10. Legal and Regulatory risk	The application must abide by several laws and rules on healthcare, data privacy, and other topics.
11. Server down	The whole application may affect if the server is down.

Source: Own Representation

Fig 6: Risk Matrix



Source: Own Representation

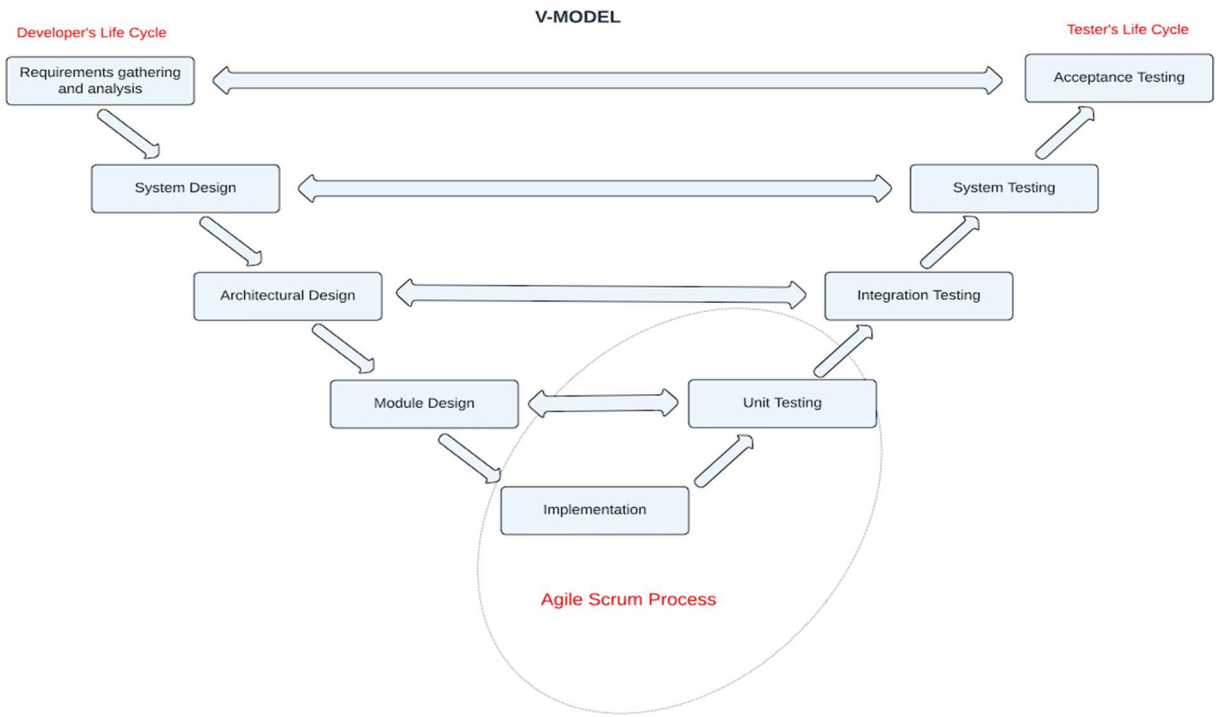
7. PROJECT METHODOLOGY & PLANNING

7.1 V-Model

A software development process called the V-model is employed to make sure that the software being created is high-quality and fits the needs of the customer. The development process is shown as a V-shaped diagram, with the initial requirements collecting and design phases at the top and the testing and delivery phases at the bottom. This is why the model is known as the "V" model. Software development teams can lower the likelihood of mistakes and defects, enhance quality, and boost customer happiness by adhering to the V-model. The following is a list of the main explanations for using the V-model approach for this project.

- Before beginning the real development phase, there must be specific requirements, test cases, test strategies, test scripts, and planning because this project involves the safety of the patient and is the most crucial from a safety standpoint.
- The V-model promotes communication and cooperation among the various software development teams, including the developers, testers, and business analysts.
- The V-model assists teams in identifying issues early in the development lifecycle, when they are simpler and less expensive to address, by putting a priority on testing and validation throughout the development process.

Fig 7: V-model



Source: Own Representation

Fig 7 shows the different steps involved in the V-model for software development.

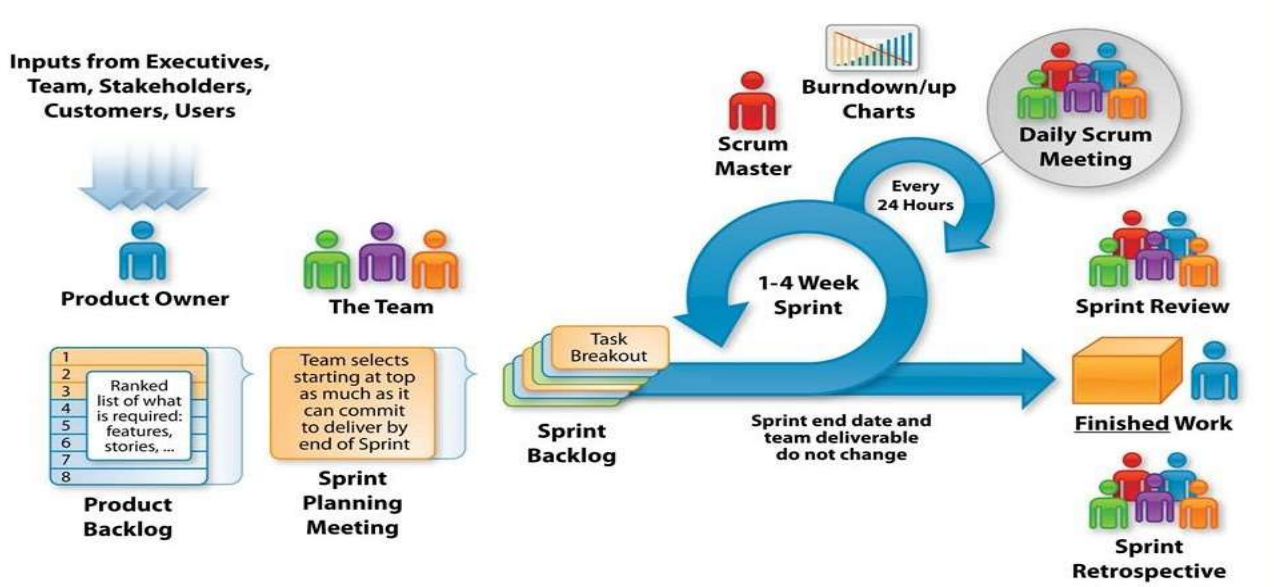
- Requirement gathering and analysis: The criteria for the “My Mobile Doctor” application are created and documented during this phase. This involves learning more about the intended user’s requirements, and the features and functionalities that the application needs. As business requirements can be used as input for acceptance testing, the design planning for the acceptance test is completed at this stage.
- System Design: It is time to design the full system after you are certain of the specifications for your product. The system design will incorporate a comprehensive understanding of the hardware and communication configuration for the ongoing project. The system test plan is designed based on the system design. More time will be available for the actual test run if this is done sooner.
- Architectural Design: This stage involves comprehending and designing architectural specifications. Usually, several technological approaches are suggested, and the ultimate choice is made based on the technical and economic viability. The system design is divided into additional modules with various functions. High-Level Design (HLD) is another name for this. At this stage, the exchange of data and communication with external systems (other systems), as well as internal modules, is understood and defined. Using this knowledge, integration tests can be created and documented at this point.
- Module Design: The internal details of each system module are specified at this phase, referred to as Low-Level Design (LLD). The design must work with both other external systems and other system architecture elements. Any development process must include unit testing since it allows for the early detection and correction of most bugs and mistakes. These unit tests may now be developed based on the internal module designs.
- Implementation: The system modules generated during the design phase are coded during the implementation phase. Coding is carried out in compliance with the coding guidelines and standards. The code goes through several code reviews and is performance-optimized before the finished build is checked into the repository.
- Unit Testing: Unit tests built during the module design step are run against the code during this validation stage. Unit testing helps in the early bug-elimination phase by doing tests at the code level.
- Integration Testing: Integration testing is related to architectural design. The effectiveness of the internal components of the systems is assessed through integration testing.
- System Testing: System tests look at both the system's connection with other systems and its general operation. Most of the software and hardware compatibility issues can be discovered when this system test is being run.

- Acceptance Testing: Acceptance testing, which is related to the business requirement analysis process, entails assessing the product in a user environment. Acceptance testing uncovers issues with other systems that are incompatible with the user environment. Additionally, it finds non-functional problems in the real user environment, such as load and performance flaws.

7.2 Agile Scrum Process

Agile Scrum is a well-liked project management technique for creating software that emphasizes incremental, iterative development. It is a fluid and adaptive structure that enables team members to work together and continuously develop. The Agile Scrum technique can be utilized in the context of a "My Mobile Doctor" to control the development process and guarantee that the application is created promptly and effectively. Here is how the Agile Scrum framework can be applied to a "My Mobile Doctor" application:

Fig 8: Scrum ceremonies



Source: google.com

- Product Owner: The Product Owner is responsible for defining the requirements and priorities of the application. In the case of an online doctor consulting application, the Product Owner would be responsible for defining the features and functionality that the application must have, such as the ability to schedule appointments, conduct message consultations, and manage patient records.

- Scrum Master: The Scrum Master oversees overseeing the team's productivity and assisting the development process. Daily stand-up meetings, sprint planning meetings, and sprint retrospectives are all their responsibility.
- Development Team: The Development Team oversees putting the application's features and functioning into practice. They collaborate to develop the program in sprint-style iterations. Each sprint usually lasts between two and four weeks, and it produces a functioning piece of software.
- Sprint Planning: The team holds a sprint planning session at the start of each iteration to determine the tasks and user stories that will be performed during the sprint.
- Daily Stand-up Meetings: Each day, the team convenes for a quick stand-up meeting to discuss the day's work, identify any obstacles, and discuss progress.
- Sprint Review: The team does a sprint review after each sprint to show the working increment of the software and solicit comments from stakeholders.
- Sprint Retrospective: The team does a sprint retrospective to look back on the sprint and pinpoint areas for improvement after the sprint review.

7.3 Epics and User Stories

- **Epic 0**

Epic 0: Setup Development and Testing Infrastructure, Analysis.			
User Story 0.1	Priority Must	Complexity	Team Developer
As a Developer I want the software to be set up for mobile and desktop application development So that I can start a development			
Acceptance Criteria: Given the required software When the functionality/task is available The actual development can start.			

User Story	Priority	Complexity	Team
0.2	Must		Tester

As a Tester

I want the software to be continuously unit tested before deployment

So that I can write test cases which are then done automatically.

Acceptance Criteria:

Given the developed code

When the functionality is implemented

The actual testing can start.

User Story	Priority	Complexity	Team
0.3	Must		Business Analyst / Product owner

As a product owner

I want a requirement analysis to be done

So that I can create an Epics and User Story for functionality

For each sprint.

Acceptance Criteria:

Given the project's requirements

When the requirement analysis is done

The user story creation can start.

User Story	Priority	Complexity	Team
0.4	Must		Product owner, PMO

As a product owner and PMO

I want a risk analysis to be done

So that I can identify the potential risk

Acceptance Criteria:

A thorough risk assessment technique must serve as the foundation for the risk analysis process.

- **Epic 1**

Epic 1: User Login and Registration Page for Mobile/Web Application			
User Story 1.1	Priority Must	Complexity	Team Backend Developer
As a user I want the login page So that I can log in to the application And use the application.			
Acceptance Criteria: Given the username and password When the sign-in button is clicked Then it should authenticate. Develop the GET API with username as the request param to get the user details from USER table.			
User Story 1.2	Priority Must	Complexity	Team Backend Developer
As a user I want the registration page So that I can register for the application And use the application.			
Acceptance Criteria: Given the username, phone, email, Password When the sign-up button is clicked, then it should register the patient. Develop the POST API with username, phone, email, and password as the request body to save the user details to USER table.			

User Story 1.3	Priority Must	Complexity	Team Frontend Developer
As a user I want the text field for the login So that I can enter the details And use the application.			
Acceptance Criteria: Create the text field for username and password. Create the sign-in button. Create a validation for username and password.			
User Story 1.4	Priority Must	Complexity	Team Frontend Developer
As a Patient I want the text field for the registration So that I can enter the details And register for the application.			
Acceptance Criteria: Create the text field for username, email, phone, and password, and confirm the password. Create the signup button. Create a validation for password mismatch, the phone should accept only numbers.			

- **Epic 2**

Epic 2: Doctor Login and Registration Page for Mobile/Web Application			
User Story 2.1	Priority Must	Complexity	Team Backend Developer
As a doctor I want the login page So that I can log in to the application And use the application to give consultation to patients.			
Acceptance Criteria: Given the username and password When the sign-in button is clicked or the sign-in with google Then it should authenticate. Develop the GET API with username as the request param to get the doctor's details from DOCTOR table.			
User Story 2.2	Priority Must	Complexity	Team Backend Developer
As a doctor I want the registration page So that I can register for the application And use the application.			
Acceptance Criteria: Given the username, phone, Email, Unique Id, Password When the sign-up button is clicked Then it should register the doctor's details. Develop the POST API with username, phone, email, Unique Id, and password as the request body to save the doctor details to DOCTOR table.			

User Story	Priority	Complexity	Team
2.3	Must		Frontend Developer

As a doctor

I want the text field for the login

So that I can enter the details

And use the application.

Acceptance Criteria:

Create the text field for username and password.

Create the sign-in button.

Create a validation for username and password.

User Story	Priority	Complexity	Team
2.4	Must		Frontend Developer

As a doctor

I want the text field for the registration

So that I can enter the details

And register for the application.

Acceptance Criteria:

Create the text field for username, email, phone, hospital location, and password, and confirm the password.

Create the signup button.

Create a validation for password mismatch, the phone should accept only numbers.

- **Epic 3**

Epic 3: Select Your Account and Health Insurance Verification Mobile/web application			
User Story 3.1	Priority Must	Complexity	Team Frontend Developer
As a user I want to select whether I am a patient or doctor So that I can select the appropriate features.			
Acceptance Criteria: On the app load, it should prompt an option to choose whether they are patient or doctor. The user should be taken to a patient login if they choose "patient" as their selection. If the user chooses "doctor," they ought to be taken to a login for doctors.			
User Story 3.2	Priority Must	Complexity	Team Backend Developer
As a user Once the registration is complete I want to upload my health insurance details And avail of the benefits			
Acceptance Criteria: Given the registration details When the health insurance card is uploaded Then it should activate the user for Medicare benefits. Develop a function to read the insurance card. Develop a POST API to save the card details in USER Table.			

User Story 3.3	Priority Must	Complexity	Team Frontend Developer
As a patient I want to upload my health insurance details And avail of the benefits			
Acceptance Criteria: Create a page to upload the health insurance card Create an upload button			

- **Epic 4**

Epic 4: AI Chatbot and Live Doctor Messaging Mobile/web application			
User Story 4.1	Priority Must	Complexity	Team Backend Developer
As a user I want to be able to interact with an AI trained Chatbot So that I can get answers to my health-related question Without having to schedule an appointment.			
Acceptance Criteria: I can use a web or mobile platform to access the doctor chatbot as a user. I want the chatbot to be able to respond to queries about doses, side effects, and interactions with other medications if I need medication information. Integrate the NLP function for text-to-text and text-to-speech recognition. The chatbot must be able to comprehend natural language requests and quickly deliver pertinent answers.			

User Story 4.2	Priority Must	Complexity	Team Frontend Developer
As a user I want to easily interact with AI chatbot through the user-friendly frontend interface So that I can get answers to my health-related question			
Acceptance Criteria: Launch the AI chatbot's frontend interface. The user should be able to locate the chatbot and communicate with it with the frontend interface's clear and simple design.			
User Story 4.3	Priority Must	Complexity	Team Backend Developer
As a user I want to easily interact with live doctors So that I can get answers to my health-related question If the AI chatbot is not able to provide a solution.			
Acceptance Criteria: Connect to live available doctors online. Patients and doctors must be able to communicate instantly. Implement the instant messaging features.			

- **Epic 5**

Epic 5: Develop SOS features for mobile/web			
User Story 5.1	Priority Must	Complexity	Team Backend Developer
As a user with emergency need I want a SOS to be integrated withing app So that I can use it in an emergency.			

Acceptance Criteria: The user's location must be able to be transmitted to the specified emergency contact(s) through the SOS function. When used by the user, the SOS function must be capable of dialling emergency services.			
User Story 5.2	Priority Must	Complexity	Team Frontend Developer
As a user with emergency need I want to have a SOS button So that I can use it in an emergency.			
Acceptance Criteria: The user interface must make the SOS button simple to find. The SOS button must be activated by a single tap from the user. The SOS button must be labelled clearly as an emergency feature.			

- **Epic 6**

Epic 6: Home Page with Health Care Blogs			
User Story 6.1	Priority Should	Complexity	Team Backend Developer
As a user I want a home page with and educational health blog posts so that I may learn more about different health issues and decide for myself what is best for my personal health and wellness.			
Acceptance Criteria: Develop a GET API to get the health details from BLOG Database.			

User Story	Priority	Complexity	Team
6.2	Should		Front Developer

As a user

I want to have user friendly home page design with health blogs

so that I can easily navigate and find relevant health information

Acceptance Criteria:

The home page for website/app with health blog must have a simple, contemporary look that is pleasing to the eye.

Users must be able to read blog posts without being distracted, with clear information hierarchy and tidy typography.

The website must load quickly and responsively with minimal latency or content display delays.

User Story	Priority	Complexity	Team
6.3	should		Front Developer

As a user

I want to have user friendly header section for home page

so that I can easily navigate to relevant information

Acceptance Criteria:

The header for home page for website has a simple, contemporary look that is pleasing to the eye.

The header must have the About, News, Services, Contact Us, Privacy Policy sections which is clickable.

User Story	Priority	Complexity	Team
6.4	should		Front Developer

As a user

I want to have access to About, Contact Us, Privacy Policy, News, Services

so that I can learn more about the website and get in touch with website/app owners

Acceptance Criteria:

It must include links or buttons that are prominently labelled and easy to find on all website pages.

The contact information for the website's owners or administrators, such as their email address, phone number, or physical location, must be correct and current.

- **Epic 7**

Epic 7: Side Bar for Website/App			
User Story 7.1	Priority Should	Complexity	Team Front Developer
As a user I want to have a side bar section on mobile and website so that I can easily populate content and navigate.			
Acceptance Criteria: The sidebar material must be simple for users to browse, with distinct headers and titles. The sidebar must include user profile photo, name, test report section, order medicine section. The sidebar area must not occupy an excessive amount of screen real estate and should be simple to collapse or extend.			
User Story 7.2	Priority Must	Complexity	Team Backend Developer
As a user I want to view and download the test report so that I can keep track of my test results.			
Acceptance Criteria: Develop the GET API to get the user's test details from REPORT Database. Write a function to convert the user details from JSON to pdf or csv format.			
User Story 7.3	Priority Must	Complexity	Team Frontend Developer
As a user I want a user friend interface to view and download the test report so that I can keep track of my test results.			

Acceptance Criteria: The user's account dashboard or a specific area of the website must provide quick access to the test report download and view part. Users must be able to get their test results with only one click in a standard file format, such PDF or CSV.

- **Epic 8**

Epic 8: Medicine Ordering System			
User Story 8.1	Priority Must	Complexity	Team Backend Developer
As a user I want to able to order medicines online so that I can purchase medicines without going to pharmacy			
Acceptance Criteria: Develop a GET API with area code as request parameter to access the specified medical database.			
User Story 8.2	Priority Must	Complexity	Team Frontend Developer
As a user I want to able to order medicines online so that I can purchase medicines without going to pharmacy			
Acceptance Criteria: The website must feature an intuitive user interface that responds to user input, as well as clear and accurate information about each pharmaceutical, such as its name, dose, and cost.			

- **Epic 9**

Epic 9: Medicine Ordering Confirmation			
User Story 9.1	Priority Must	Complexity	Team Frontend Developer
As a patient I want to receive a confirmation of order so that I can assure that my order has been placed.			
Acceptance Criteria: Ask a user to enter his address, telephone before placing order. Create an order confirmation page. Send the confirmation to phone.			

- **Epic 10**

Epic 10: Doctor's Account			
User Story 10.1	Priority Must	Complexity	Team Backend Developer
As a doctor I want to check the patient online and message Instantly so that I can ask know their problem and give consultation.			
Acceptance Criteria: Implement the instant messaging feature			
User Story 10.2	Priority Must	Complexity	Team Frontend Developer
As a doctor I want to have an interactive messaging interface so that I can response to the patients queries quickly			

Acceptance Criteria: Design the messaging interface. Design the voice recognition button.			
User Story 10.3	Priority Must	Complexity	Team Backend Developer
As a doctor I want to upload the patient report so that I can give his test result online			
Acceptance Criteria: Develop the POST API with report details as request body to store the details in REPORT Database.			
User Story 10.4	Priority Must	Complexity	Team Frontend Developer
As a doctor I want an interactive upload dashboard so that I can upload the patient report			
Acceptance Criteria: There must be options and instructions for selecting and uploading files on the website that are both user-friendly and safe. Users must be allowed to upload several file types, including PDF and Word.			

7.3 Testcases

Testcase 1 Patient and Doctor Login Frontend	Valid Login: • Make that the system allows the user to log in successfully and leads them to the dashboard/homepage by entering the proper username and password. Invalid Login: • Make sure that if you enter an erroneous username, the system will notify you with an error message.

	• Make that the system allows the user to log in successfully and leads them to the dashboard/homepage by entering the proper username and password. • Make careful to enter an erroneous password so that the system will issue a warning that the password is invalid. Forgot Password: • Make sure the system prompts the user to input the email address connected to their account when they click the "Forgot Password" link. • Make sure you enter a valid email address so that the system can send the user a link to reset their password.
Testcase 2 Patient and Doctor Login Backend	Valid: • Test Login GET API by using user id as request parameter • User details should be expected in JSON. • Status code should be 200 Invalid: • If not found then return 404 • Internal server error then return 500 • Param missing then return 400
Testcase 3 Patient and Doctor Registration Page Frontend	Valid Registration: • Make careful to fill out all needed fields with accurate data so that the system can properly register the user account and lead them to the health card upload page and for doctor it should lead them ID card upload page • Make sure the person may successfully register by using a special username and email address that have never been used before. Invalid Registration: • If a field is needed, leave it empty and make sure the system generates the proper error message to let you know. • If you enter an email address with an incorrect format (for example, one without the "@" sign), make sure the system delivers a suitable error message to let you know the address is wrong. • Enter a password that does not match the complexity criteria and is weak (for example, it is too short or lacks special characters), and make sure the system issues a suitable error message to let you know it is weak.
Testcase 4 Patient and Doctor Registration Page Backend	Valid: • Test Register POST API by sending the user details in request body. • Status code should be 200

	Invalid: • Internal server error then return 500 • Request body missing then return 400
Testcase 5 Select Account, Health Insurance Verification, Doctors Id verification frontend	Select account: • When clicked on patient account it should redirect to patient login screen • When clicked on doctors account it should redirect to doctor's login screen Health Insurance, Doctor Id Upload and Verification: • Make sure you upload a file of a supported file format and check to see whether the system shows a success message once the upload has been completed. • Try to upload a file that has a file type that is not supported (like.exe or.bat) and make sure the system issues the proper error message to let you know the file type isn't supported.
Testcase 6 Upload Insurance Backend	Valid: • Test upload POST API by sending the details in request body • Status code should be 200 Invalid: • Internal server error then return 500 • Request body missing then return 400
Testcase 7 AI Chatbot	• Check to see if the chatbot recognizes and understands your message or query before sending it. • Try to enter a message that contains typos or grammatical problems to see whether the machine can still grasp and interpret it somewhat well. • Make sure the chatbot, using its knowledge and abilities, responds to the user's input in an appropriate and correct manner.
Testcase 8 Instant message to doctor	• Create a fresh message and make sure it gets sent successfully to the target recipient using the system. • Check that the message's recipient, sender, and timestamp are all accurate.
Testcase 9 SOS	• Send an emergency alert message to the user's emergency contacts by activating the SOS function. • Verify that the user's location details and a brief description of the emergency scenario are included in the emergency alert message. • In an emergency, make sure the system can pinpoint the user's location precisely and in real time.
Testcase 10 Home page, Header for web page	• Make sure the user can access the health blog area from the main page with ease. • Make sure the health blog area is clearly visible on the home page so that users can locate and access it.

	• Send an emergency alert message to the user's emergency contacts by activating the SOS function. • Verify that the user's location details and a brief description of the emergency scenario are included in the emergency alert message. • In an emergency, make sure the system can pinpoint the user's location precisely and in real time. • Make sure the diet, exercise, and mental health categories are included in the health blog area. • Header should include Home, About Us, Contact Us, Privacy Policy, News. • Make that the company's history, mission, beliefs, and ambitions are accurately and recently described on the "About Us" page. • Make that phone number, email address, and physical location of the application support team are all shown on the "Contact Us" page and are correct and up to date.
Testcase 11 Side Bar	• Make sure the sidebar has a menu that the user may utilize to quickly access the various website and mobile parts. • Make sure the menu for navigating the website is simple to use, well labelled, and offers direct access to the various website and mobile parts. • Make user that user profile photo, name is available which is clickable that leads to user information page.
Testcase 12 View/Download report	• Check that the user can access the report in the specified format (such as PDF, csv) • Check that the report may be downloaded in the user's preferred format (such as PDF, csv).
Testcase 13 Medicine Order	• Check to see if the user can look for a certain drug by name or category. • Verify that pertinent information, such as the medication name, dose, adverse effects, and cautions, are included on the pharmaceutical details page.
Testcase 14 Medicine Order Confirmation	• Make sure the user receives an order confirmation after submitting their medication order. • Ensure that the order confirmation contains information about the order, including the name of the drug, the amount. • Check that the website allows users to follow the status of their orders.
Testcase 15 Instant messaging with patient	• Patients issue with priority should be placed at top level. • Create a fresh message and make sure it gets sent successfully to the target recipient using the system. • Check that the message's recipient, sender, and timestamp are all accurate.
Testcase 16 Upload report	• Check that the user can access the report in the specified format (such as PDF, csv)

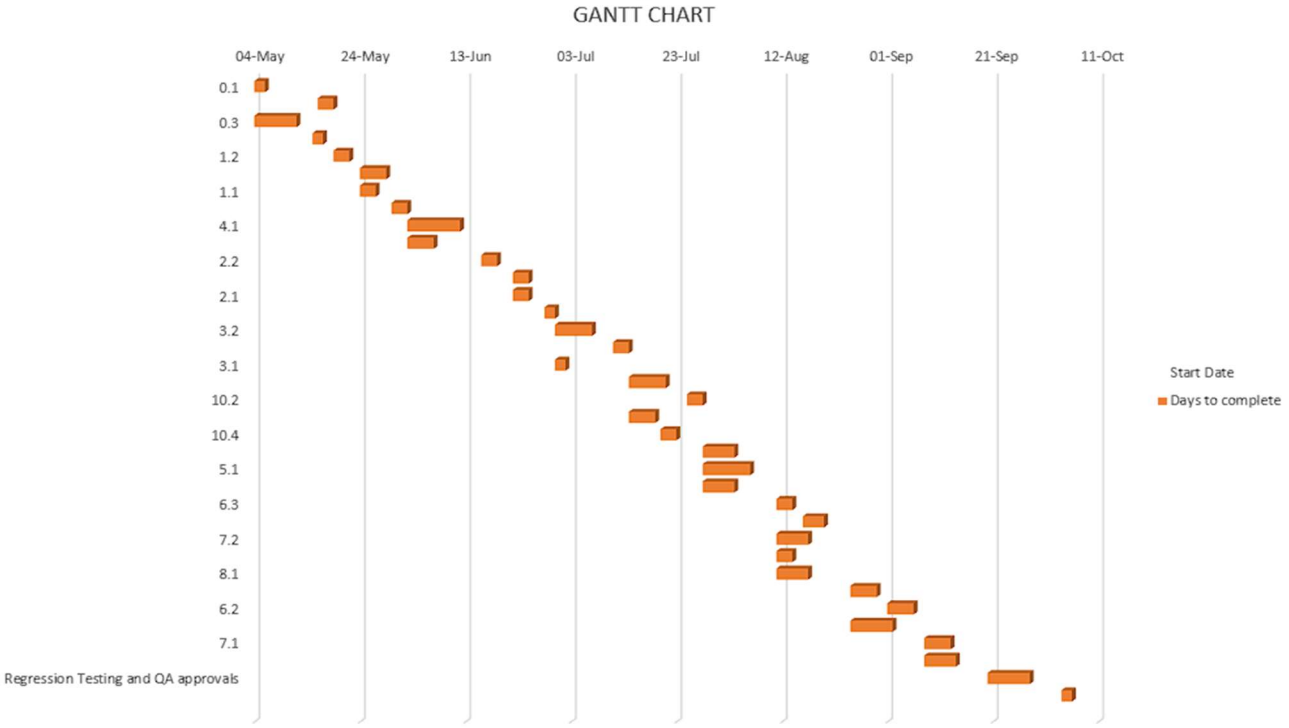
7.4 Sprint Planning

Epic	User Story	Test	Sprint	Start Date	End Date	Days	Assigned To	Story Point	Dependency	Priority
0	0.1		0	04-05-2023	06-05-2023	2	Developers	5		M
	0.2		0	16-05-2023	18-05-2023	3	Testers	8	0.3	S
	0.3		0	04-05-2023	15-05-2023	8	Product Owners	13		M
	0.4		0	15-05-2023	16-05-2023	2	Product Owners	5	0.3	M
1	1.2		1	19-05-2023	23-05-2023	3	Backend Dev	5		M
	1.4		1	24-05-2023	30-05-2023	5	Frontend Dev	8	1.2	M
	1.1		1	24-05-2023	29-05-2023	3	Backend Dev	5	1.2	M
	1.3		1	30-05-2023	01-06-2023	3	Frontend Dev	5	1.2, 1.1	M
		3	1	31-05-2023	01-06-2023	2	Tester	3		
		4	1	31-05-2023	01-06-2023	2	Tester	3		
		2	1	30-05-2023	31-05-2023	2	Tester	3		
		1	1	31-05-2023	01-06-2023	2	Tester	3		
4	4.1		2	02-06-2023	15-06-2023	10	Backend Dev	13		M
	4.2		2	02-06-2023	08-06-2023	5	Frontend Dev	8		M
2	2.2		3	16-06-2023	21-06-2023	3	Backend Dev	5		M
	2.4		3	22-06-2023	27-06-2023	3	Frontend Dev	5	2.2	M
	2.1		3	22-06-2023	27-06-2023	3	Backend Dev	5	2.2	M
	2.3		3	28-06-2023	29-06-2023	2	Frontend dev	3	2.2, 2.1	M
		7	3	16-06-2023	18-06-2023	3	Tester	5		
		4	3	22-06-2023	23-06-2023	2	Tester	3		
		3	3	28-06-2023	29-06-2023	2	Tester	3		
		2	3	28-06-2023	29-06-2023	2	Tester	3		

3	3.2		4	30-06-2023	10-07-2023	7	Backend dev	13		M
	3.3		4	11-07-2023	13-07-2023	3	Frontend dev	5	3.2	M
	3.1		4	30-06-2023	03-07-2023	2	Frontend dev	3		S
		1	4	30-06-2023	01-07-2023	3	Tester	5		
		6	4	11-07-2023	13-07-2023	2	Tester			
10	10.1		5	14-07-2023	24-07-2023	7	Backend dev	8		M
	10.2		5	25-07-2023	27-07-2023	3	Frontend dev	3		M
	10.3		5	14-07-2023	20-07-2023	5	Backend dev	8		S
	10.4		5	20-07-2023	24-07-2023	3	Frontend dev	3	10.3	S
		5	5	14-07-2023	18-07-2023	3	Tester	5		
		16	5	25-07-2023	27-07-2023	3	Tester	5		
4	4.3		6	28-07-2023	04-08-2023	6	Backend dev	5	10.1, 10.2	M
5	5.1		6	28-07-2023	10-08-2023	9	Backend dev	5		M
	5.2		6	28-07-2023	04-08-2023	6	Frontend dev	5		M
		15	6	28-07-2023	02-08-2023	4	Tester	5		
6	6.3		7	11-08-2023	15-08-2023	3	Frontend dev	5		S
	6.4		7	16-08-2023	22-08-2023	4	Frontend dev	5	6.3	S
		9	7	11-08-2023	15-08-2023	3	Tester	5		
		10	7	23-08-2023	24-08-2023	2	Tester	3		
7	7.2		7	11-08-2023	18-08-2023	6	Backend dev	8	10.3, 10,4	M
	7.3		7	11-08-2023	15-08-2023	3	Frontend dev	5	7.2	S
		11	7	16-08-2023	18-08-2023	3	Tester	5		
8	8.1		7	11-08-2023	18-08-2023	6	Backend dev	8		M
6	6.1		8	25-08-2023	31-08-2023	5	Backend dev	8		S

	6.2		8	01-09-2023	07-09-2023	5	Frontend dev	8	6.1	S
8	8.2		8	25-08-2023	05-09-2023	8	Frontend dev	13	8.1	M
7	7.1		9	08-09-2023	14-09-2023	5	Frontend dev	8		S
9	9.1		9	08-09-2023	15-09-2023	6	Frontend dev	8		M
		10	8	08-09-2923	12-09-2023	3	Tester	5		
		13	9	08-09-2023	13-09-2023	4	Tester			
		14	9	18-09-2023	19-09-2023	2	Tester	3		
		11	9	15-09-2023	19-09-2023	3	Tester	5		
Regression Testing and QA Approvals Start date: 20-09-2023 End Date: 29-03-2029										
Go Live Start Date: 04-10-2023 End Date: 05-10-2023										

Fig 9: Gantt Chart



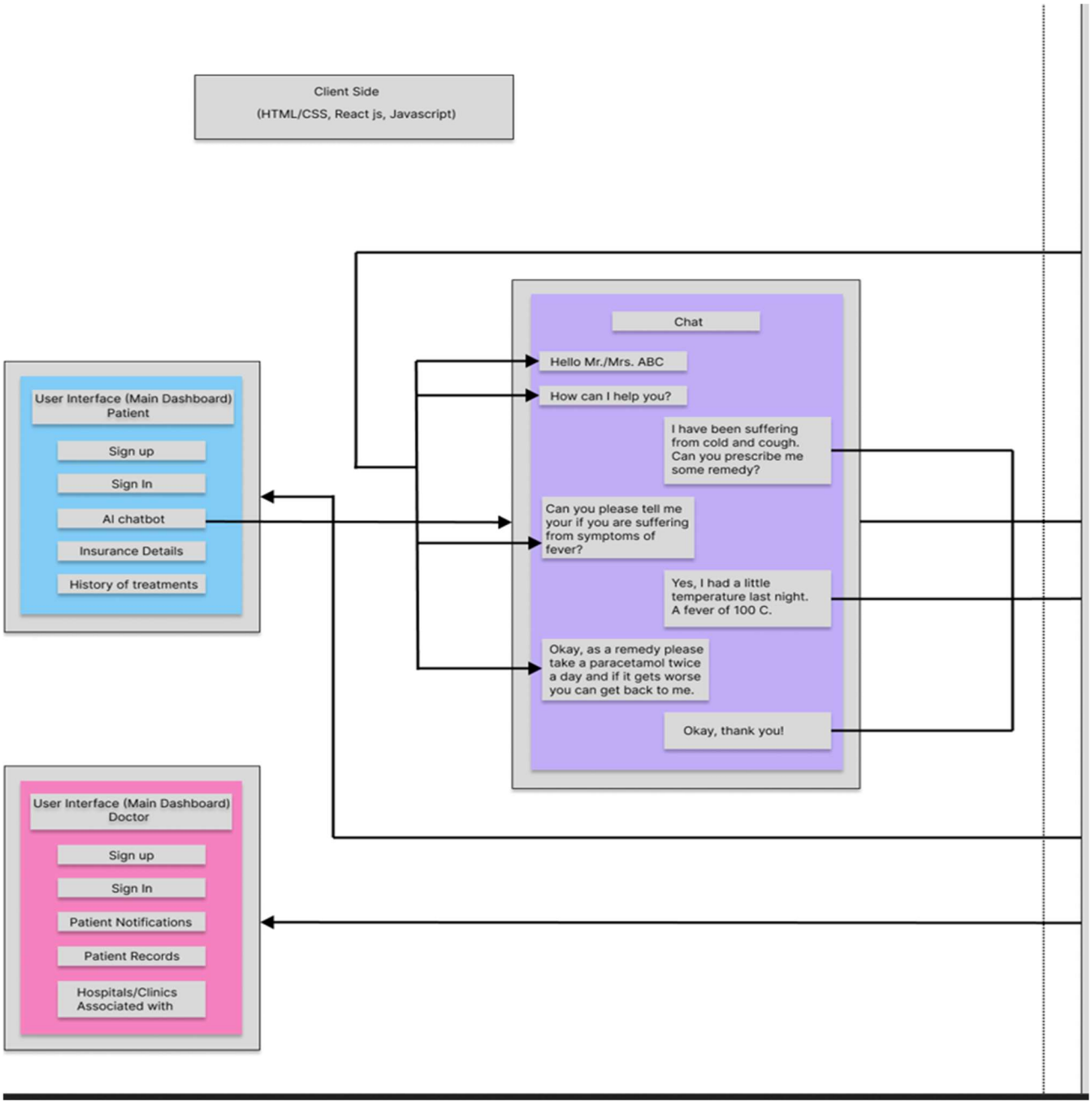
Source: Own Representation

8. TECHNOLOGY AND APPLICATION FLOW

8.1 Technical Architecture Frontend

The user interface for My Mobile Doctor was developed utilizing frontend technologies such as react.js and react-native for cross-platform mobile apps and online applications, respectively. The components are styled using Material UI. To complete the CRUD tasks, these frontend components will use a REST API.

Fig 10: Frontend Architecture

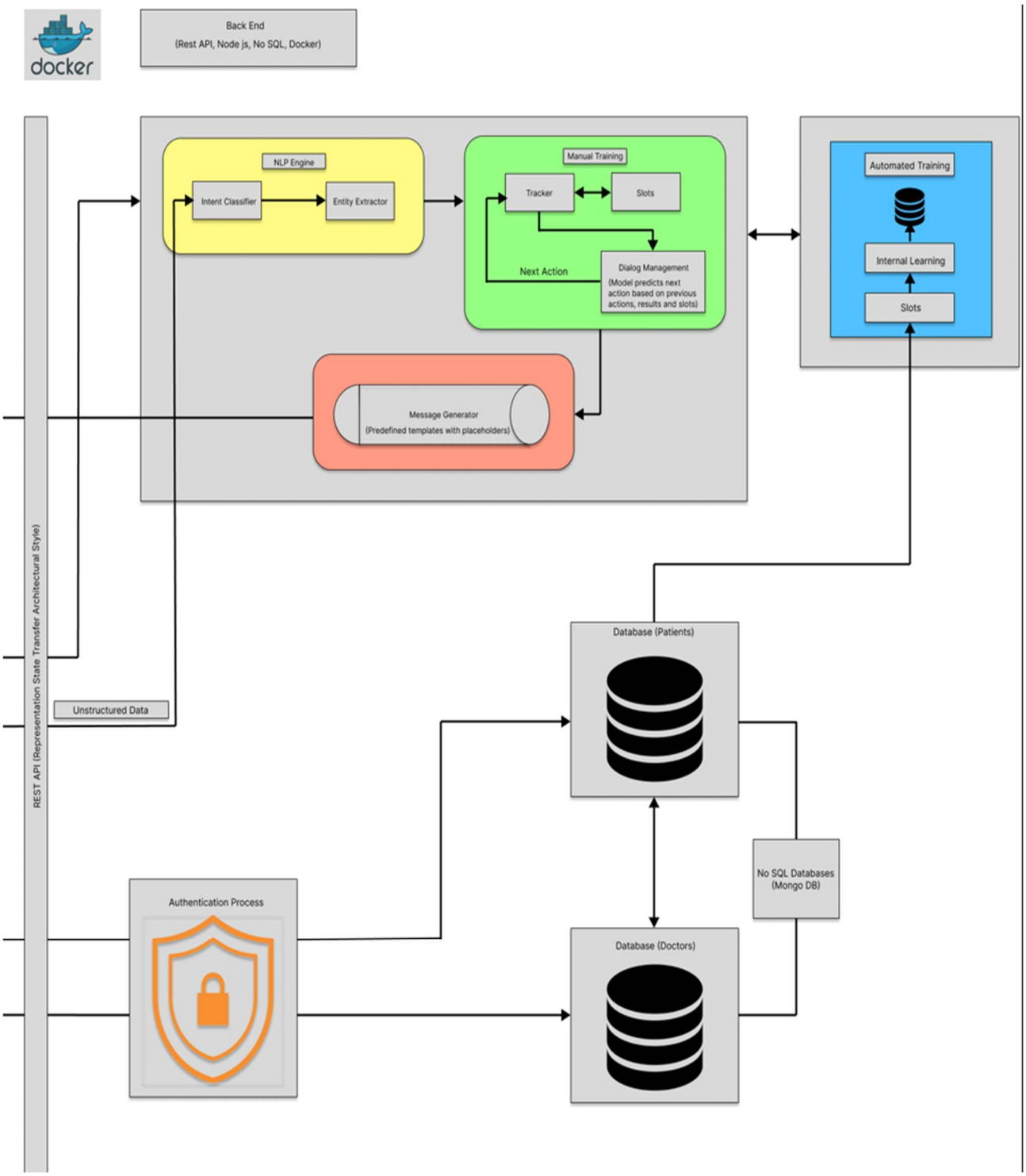


Source: Own Representation

8.2 Technical Architecture Backend

For backend Node.js is used in microservice development and MongoDB is used as a NO SQL database. Microservices are deployed using docker. NLP is used to train the chatbot using text-to-text and speech-to-text.

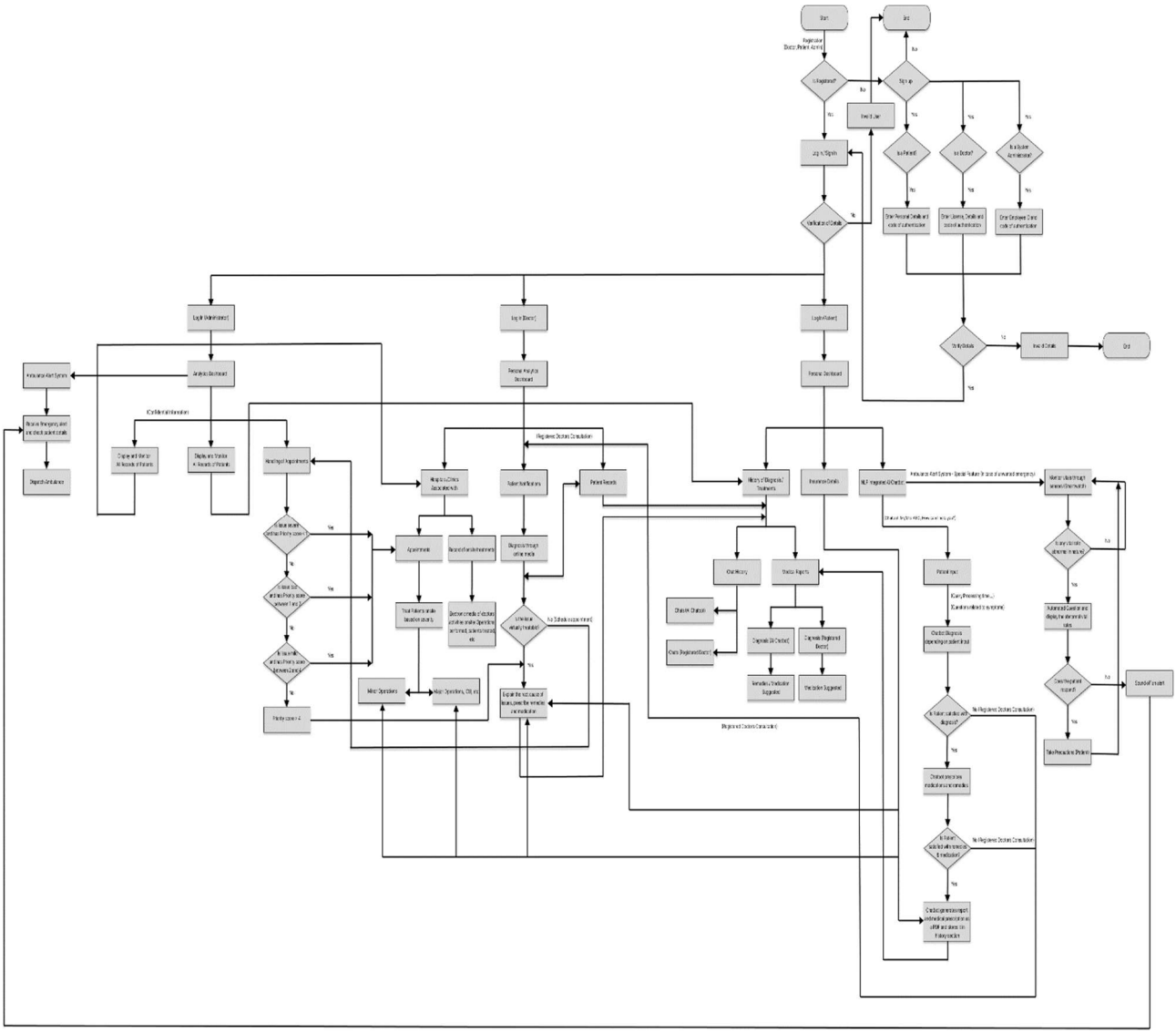
Fig 11: Backend Architecture



Source: Own Representation

8.3 Application Flow

Fig 12: My Mobile Doctor Flow Chart



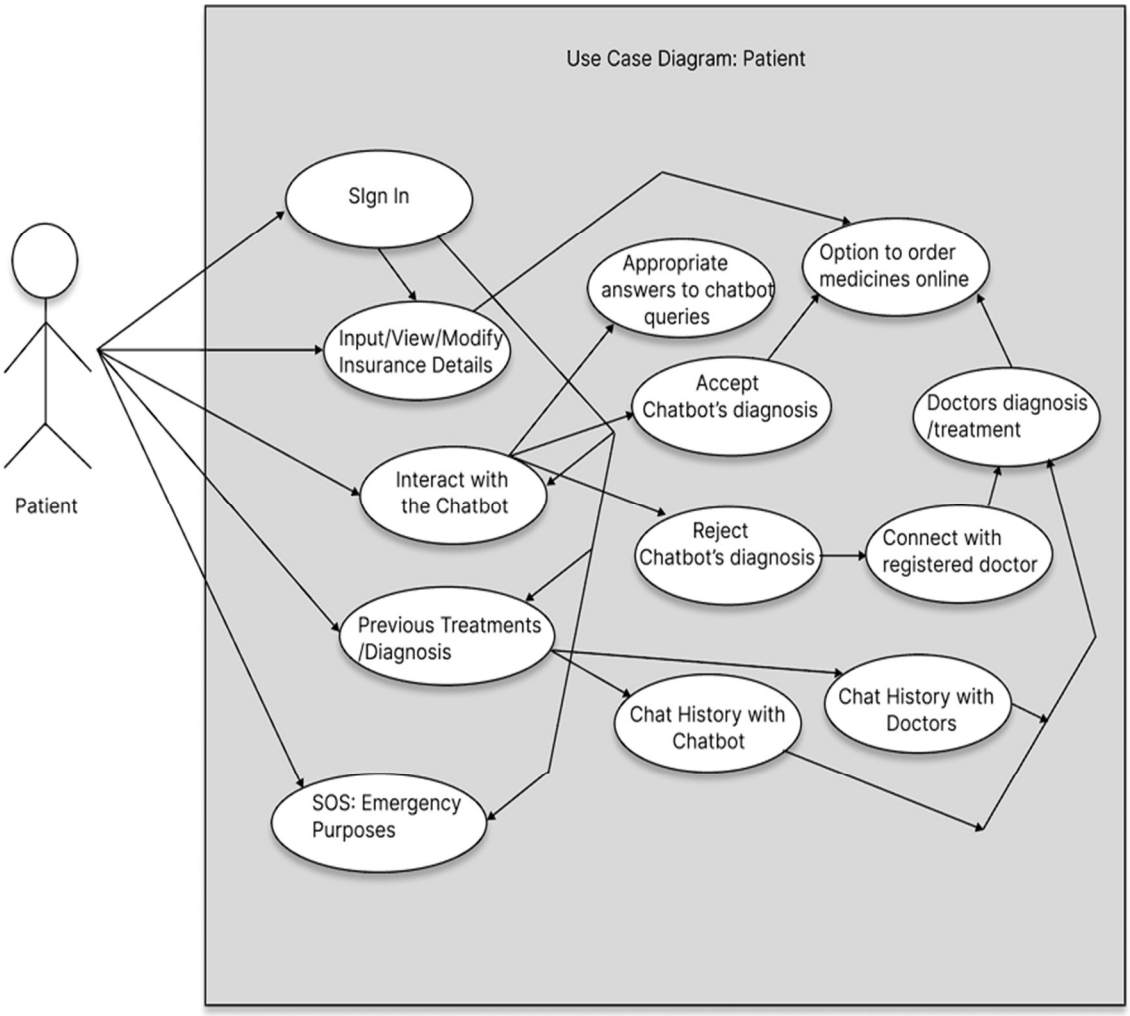
Source: Own Representation

- **Start**: On Strat of the application, the login page loads up. The login page has a link to redirect to a registration page for new users.
- **User Registration**: The User needs to complete the registration to use the application.
- **User health insurance verification**: The User uploads their insurance card to activate the account.
- **User Login**: The User login the application once the account is activated.
- **Chatbot Prompt**: On Successful login, the chatbot prompts to chat with the user regarding his issues.

- **Instant Messaging with doctor:** Chatbot is not able to prescribe the medicines, connect to live doctors for instant messaging for prescription.
- **Home page with SOS features:** The home page has SOS features for emergency services for users.
- **Medicine Ordering:** Users can navigate to order pages to order medicines at their doorsteps.
- **Test Report:** View/Download the test report uploaded by doctors.
- **End:** Logout features in the side bar of the application.

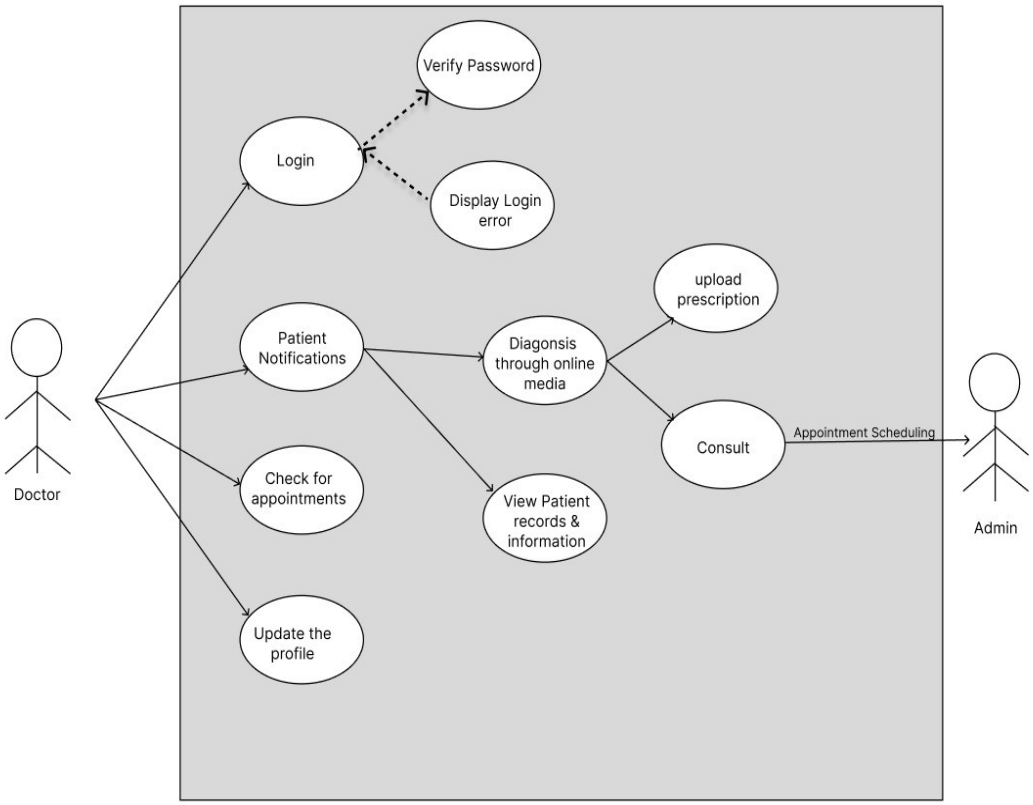
8.4 Use Case Diagrams

Fig 13: Patients' Use Case Diagram



Source: Own Representation

Fig 14: Doctors' Use Case Diagram



Source: Own Representation

9. USER INTERFACE AND DESIGNS

9.1 User Interface for Mobile Application

Fig 15: Account Selection Mobile UI

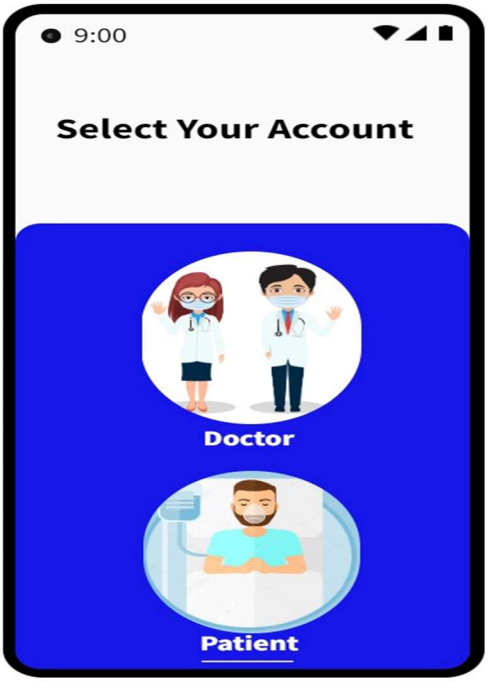


Fig 16: Patient Login Mobile UI

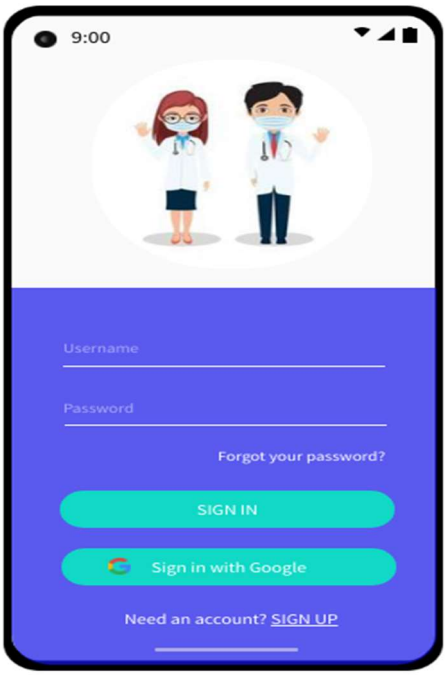


Fig 17: Patient Registration Mobile UI

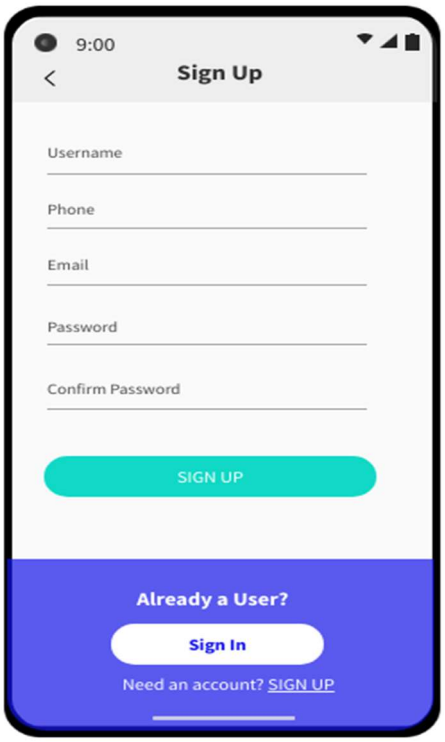


Fig 18: Insurance Upload Mobile UI

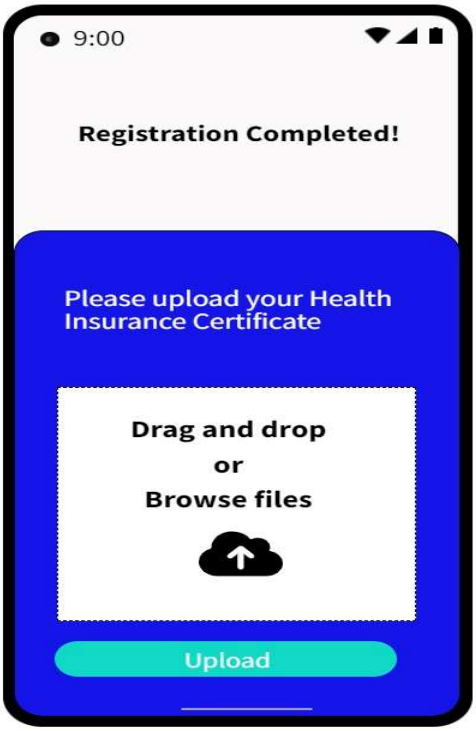


Fig 19: Chatbot popup Mobile UI

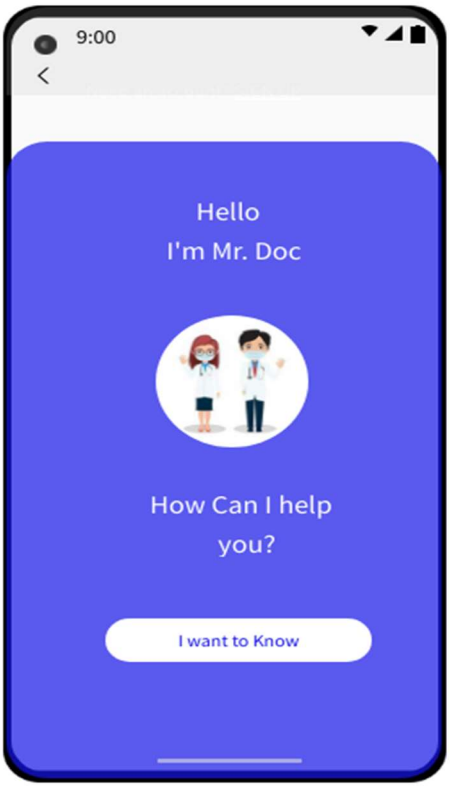


Fig 20: Chatbot Mobile UI

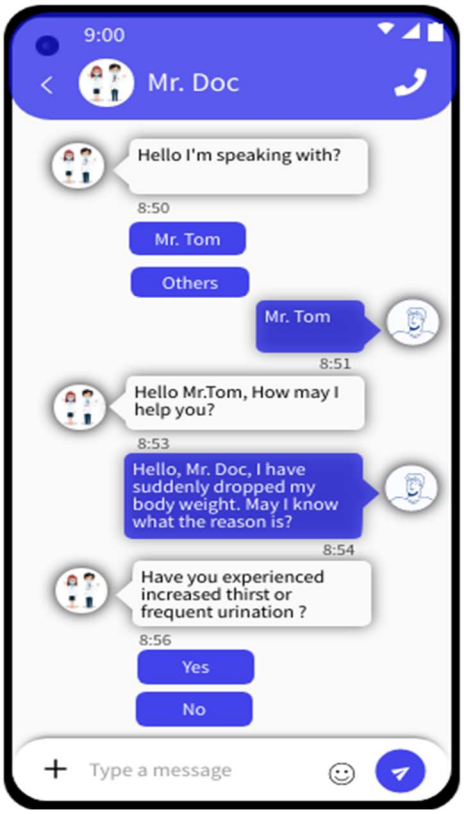


Fig 21: Chatbot Mobile UI

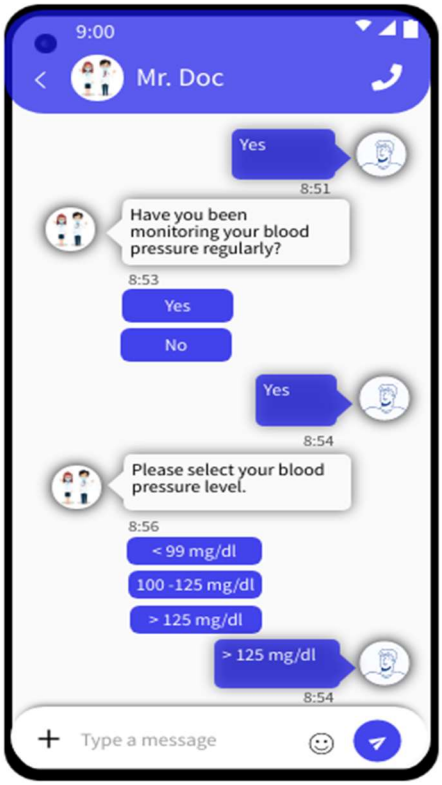


Fig 22: Instant Messaging with Doctor Mobile UI

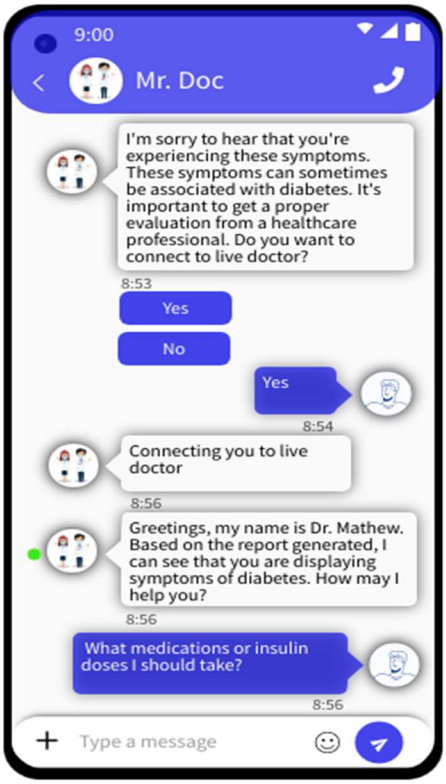


Fig 23: Home Page Mobile UI

Fig 24: Side bar Mobile UI

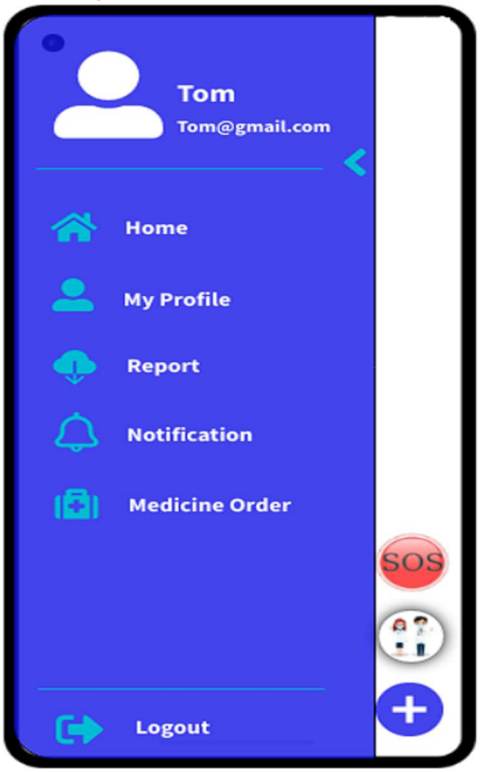


Fig 25: Medicine Order Mobile UI

Fig 26: Doctor Login Mobile UI

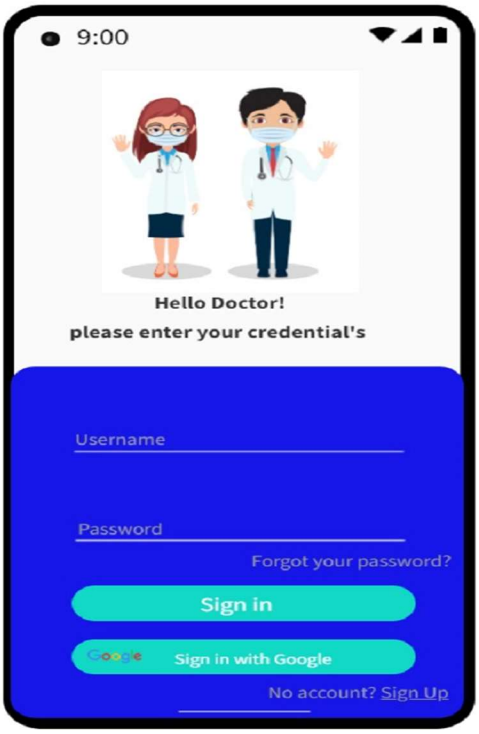


Fig 27: Doctor Registration Mobile UI

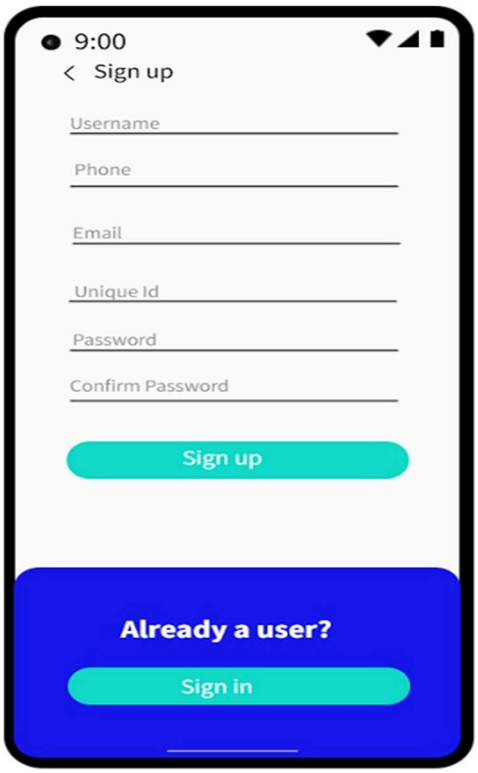


Fig 28: Doctors' Dashboard Mobile UI

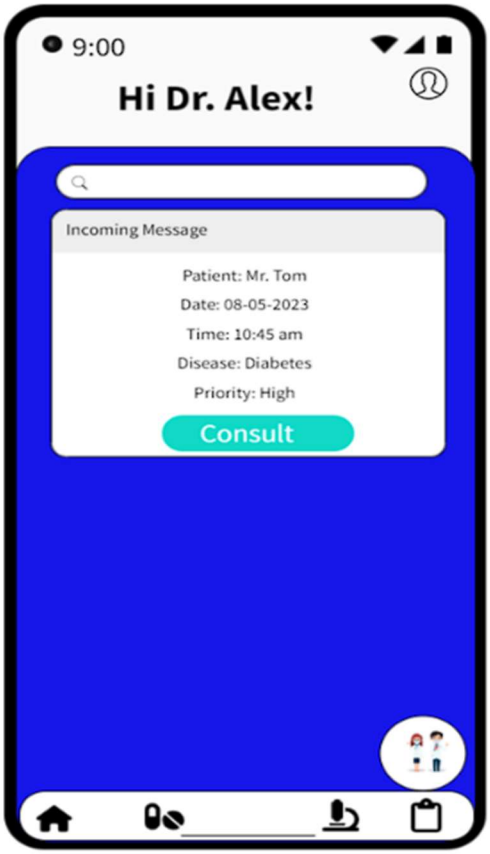
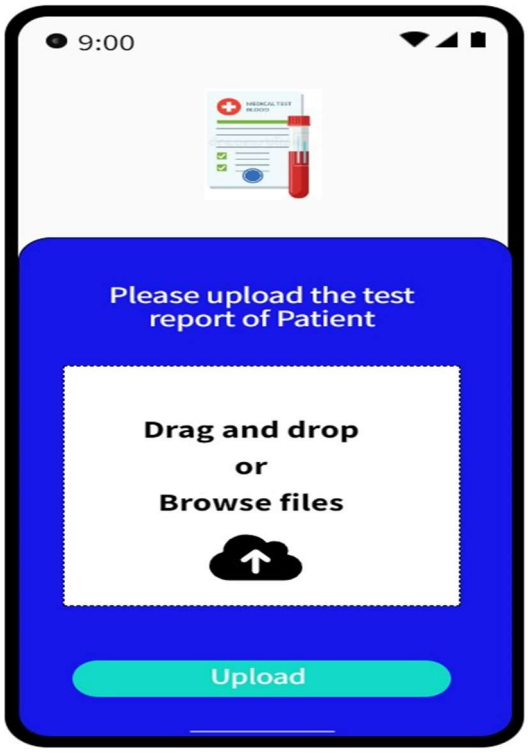


Fig 29: Test report upload Mobile UI



9.2 User Interface for Web Application

Fig 30: Account selection Web UI

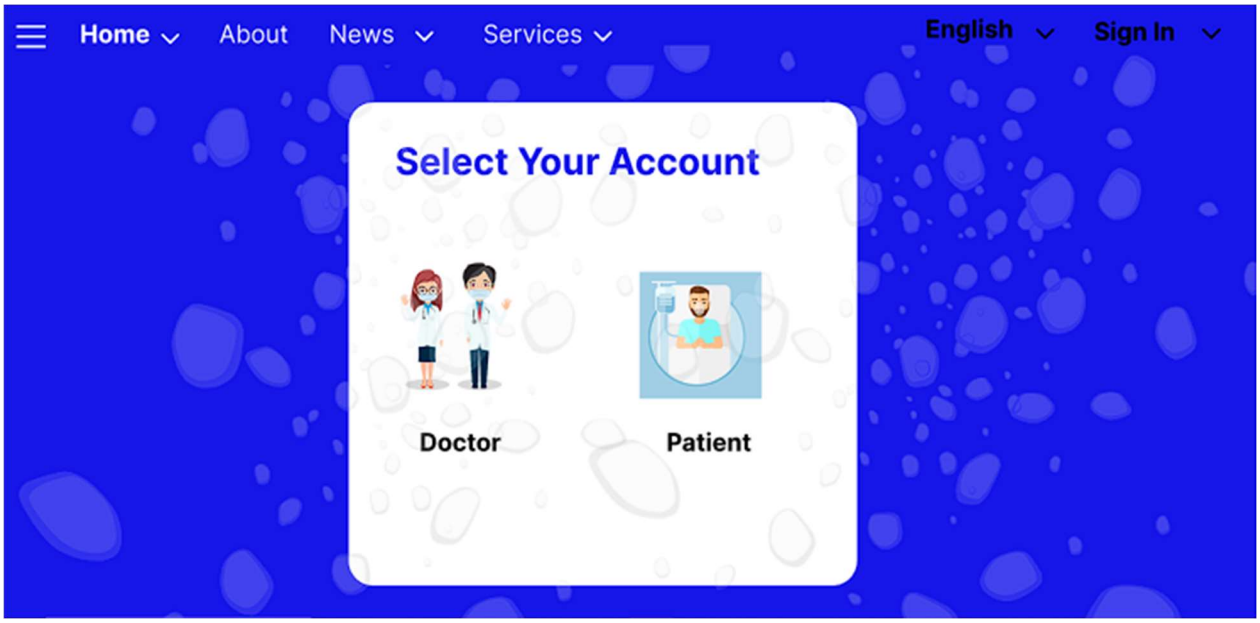


Fig 31: User Login Web UI

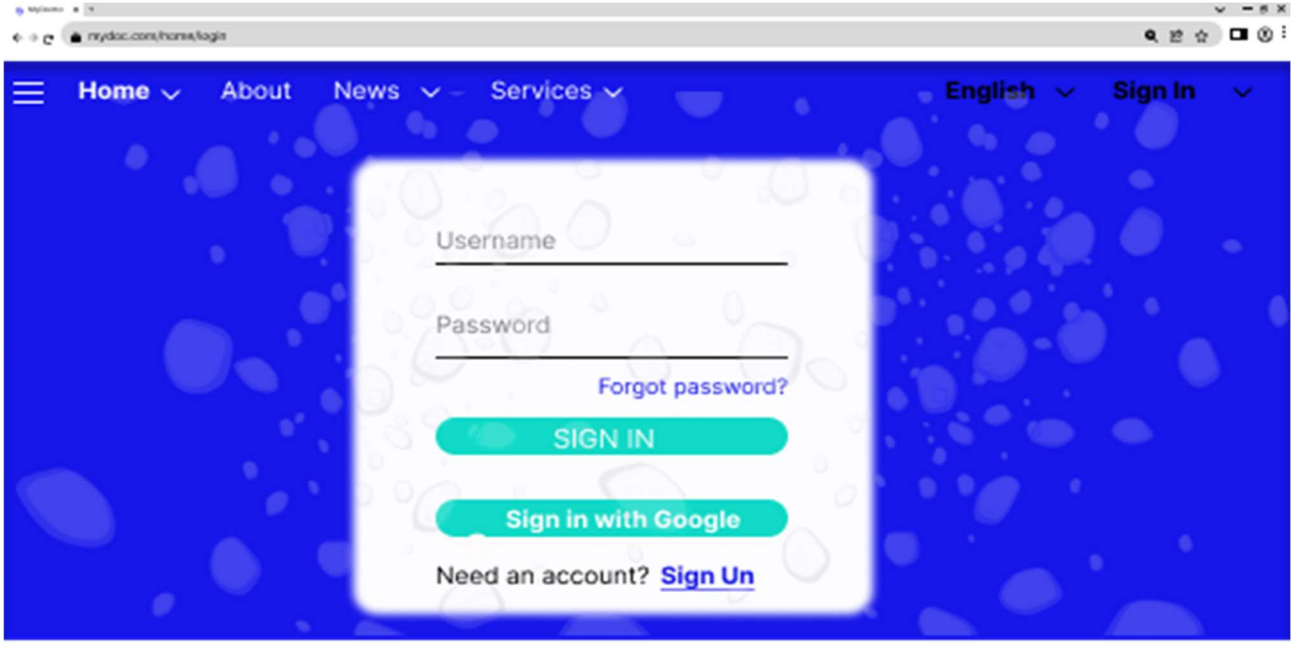


Fig 32: User Registration Web UI

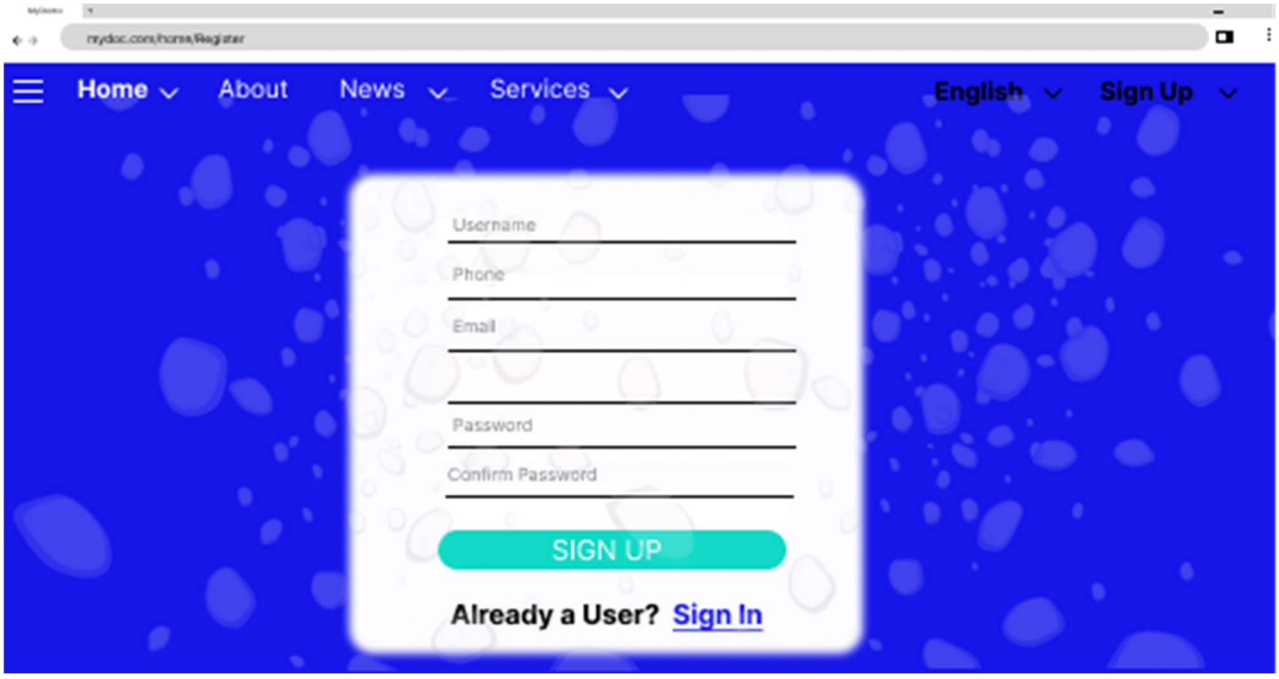


Fig 33: Insurance Upload Web UI

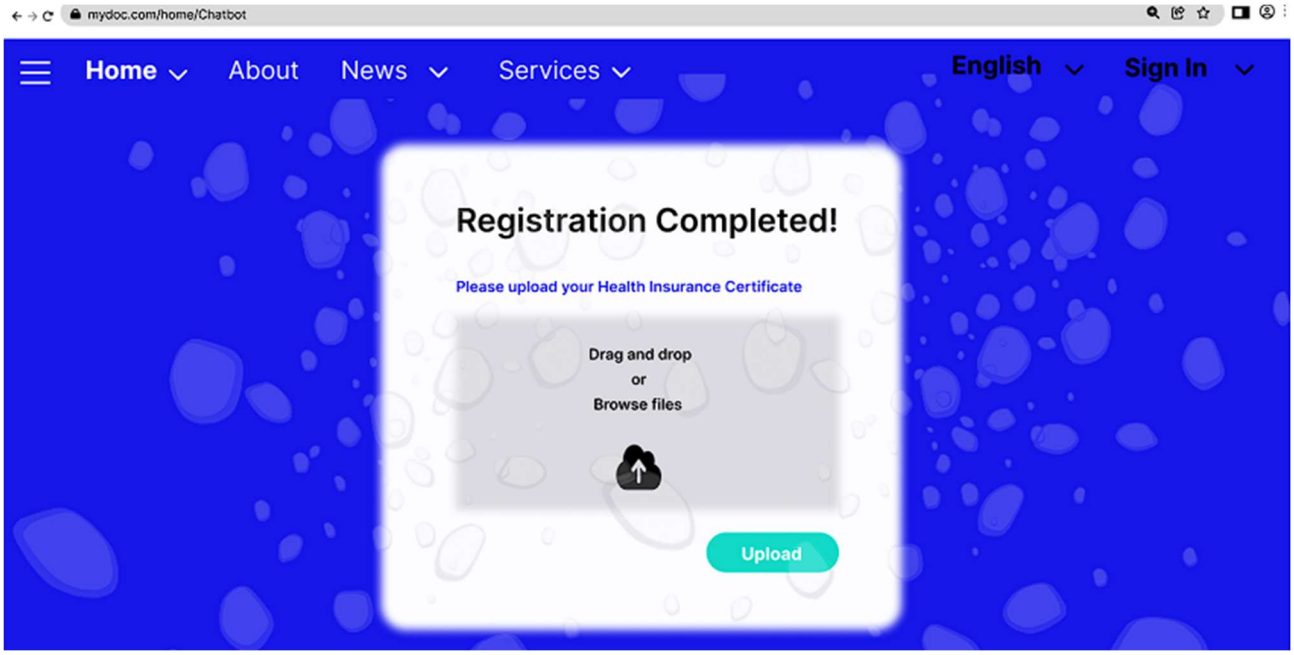


Fig 34: Home Page Web UI

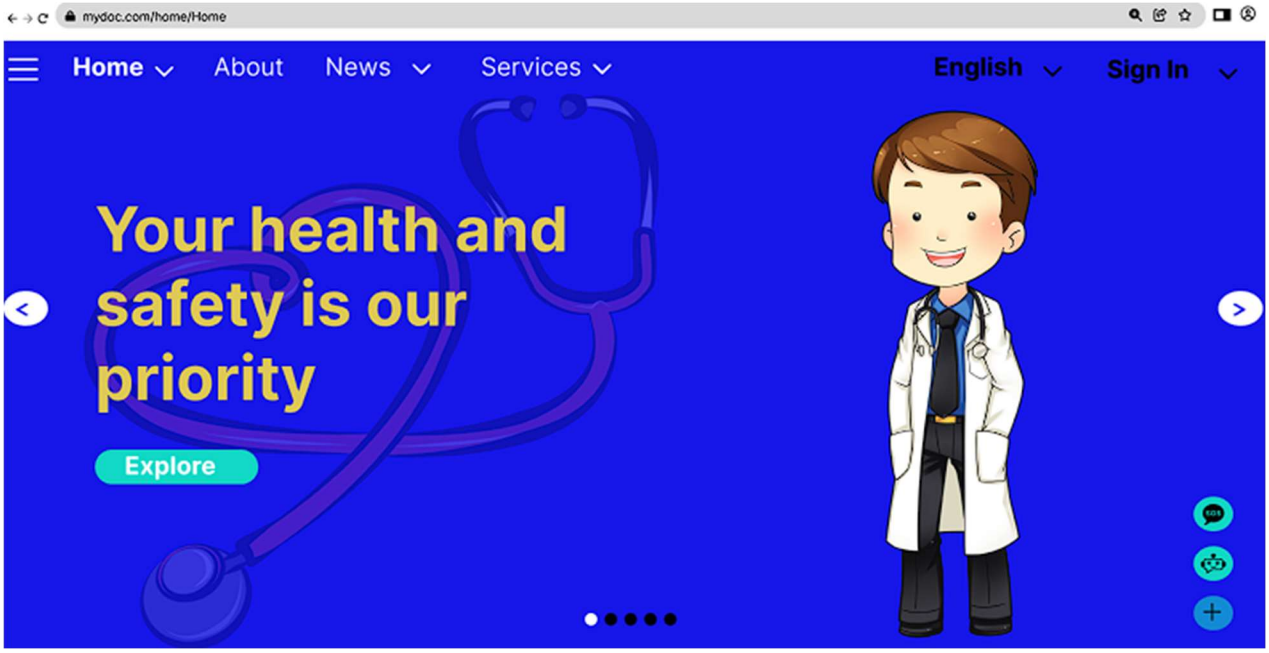


Fig 35: Chatbot Web UI

Fig 36: Chatbot Web UI

Fig 37: Instant Messaging with doctor Web UI

Fig 38: Doctor Login Web UI

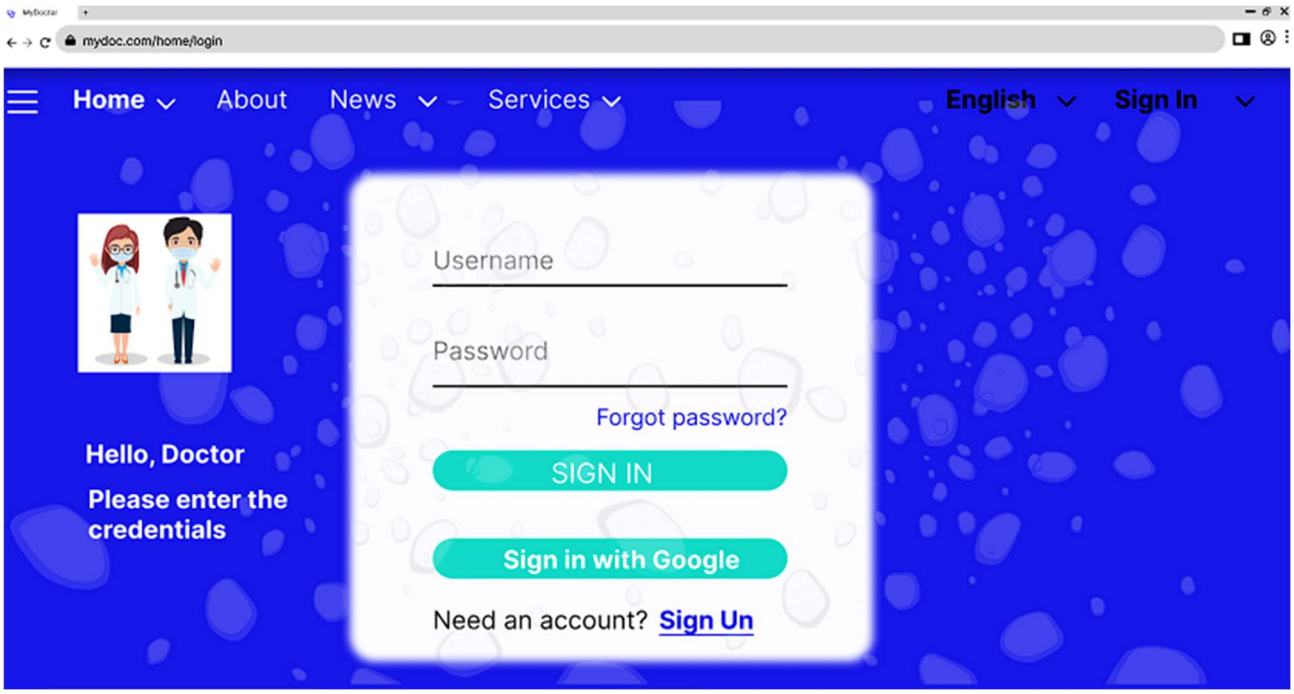


Fig 39: Doctor Registration Web UI

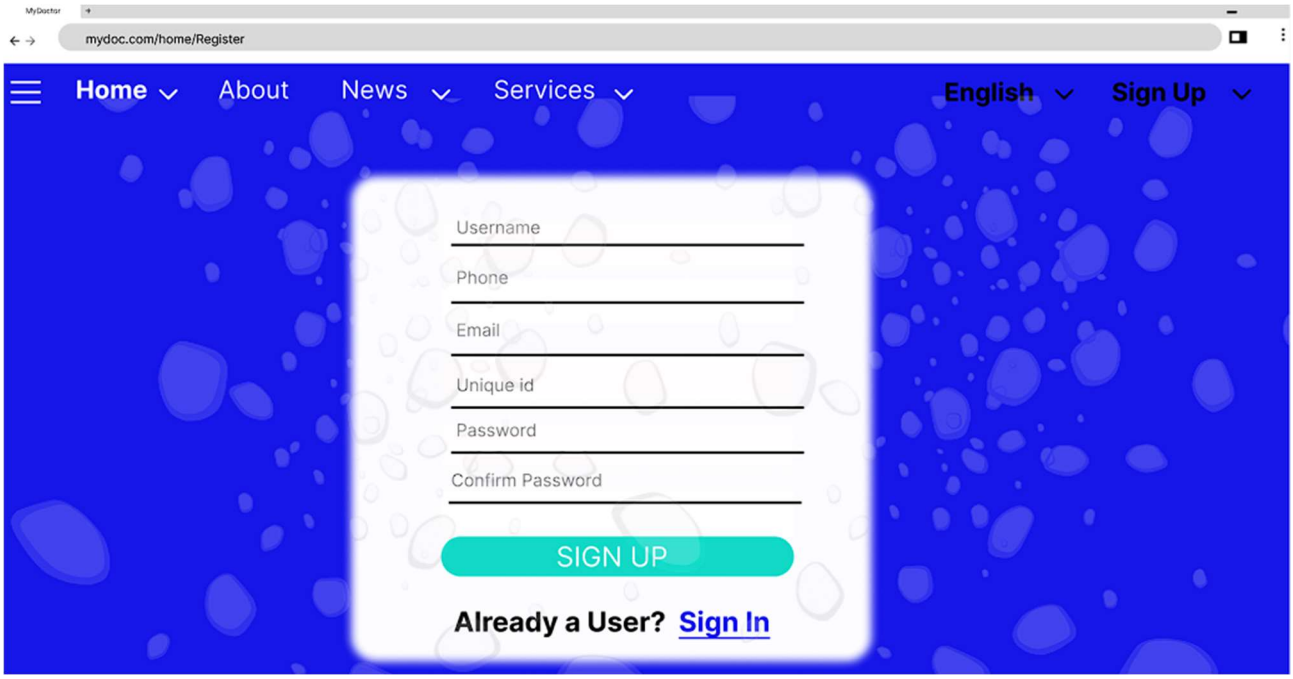


Fig 40: Doctors' dashboard Web UI

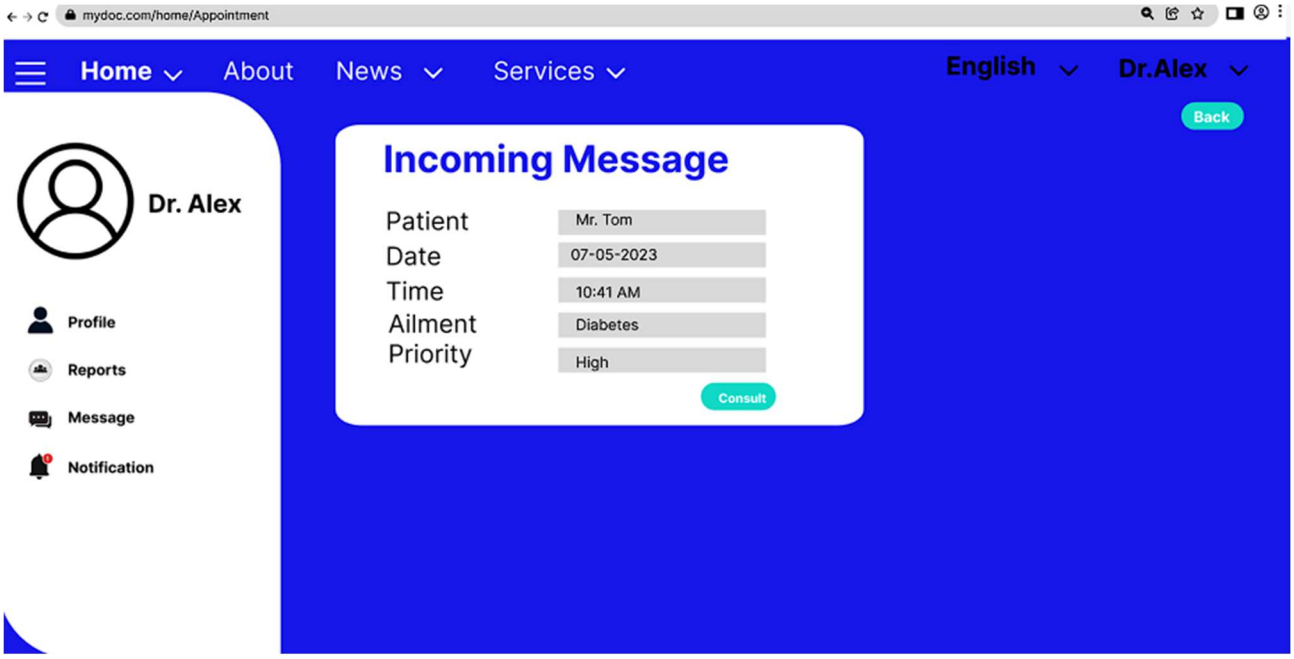
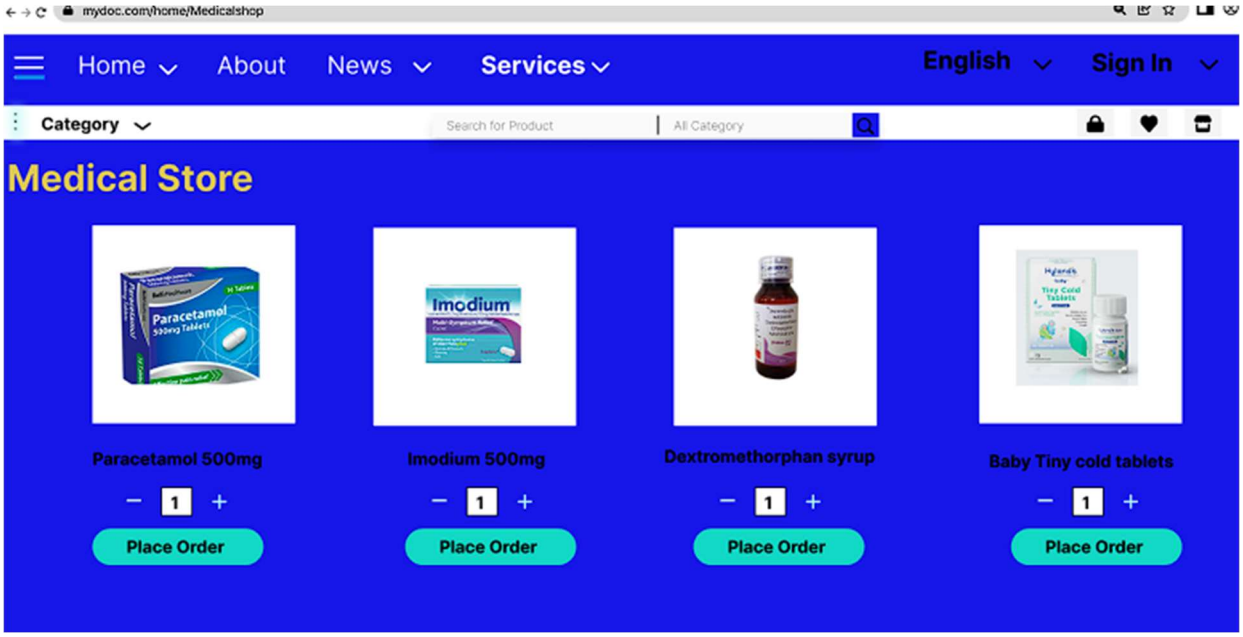


Fig 41: Medicine Order Web UI



10. FUTURE WORK

In the Version 2 release of this program development, a smartwatch with integrated SOS capabilities for older people who are illiterate about utilizing mobile and desktop applications has been taken into consideration. One proposal is to use the watch's RF technology to transmit signals to surrounding hospitals, informing them of the patient's position so they may dispatch an ambulance to their doorstep. The smartwatch will be acquired as part of this development from the current smartwatch suppliers, and our team will create the features. This project is expected to cost 2 million euros and take three months to complete.

Fig 42: Prototype of Smart Watch

Source: google.com

11. SUMMARY

11.1 Project Features

From a business perspective, "My Mobile Doctor" will assist all hospitals and clinics in general, especially those in rural regions, as they need patients to operate and the local patients will benefit in an emergency. Mismanagement of any kind that was common in the manual method will be reduced. Improvement in patient involvement and treatment adherence, to put it simply.

From the perspective of the patient, the program enables them to experience virtual diagnosis and consultation from the convenience of their homes, either from the AI chatbot or, if necessary, from a licensed expert physician. Simply put a lessening of the strain on conventional healthcare systems.

From the perspective of the doctors, the program can save them time in terms of routine situations because patients no longer need to physically visit the doctor, allowing them to focus on patients with more serious illnesses. Prioritization of patients is, to put it simply, determined by the severity of the ailment.

11.2 Total Budget

This project would have a total budget of about 4 million euros, which would cover resources, software and hardware licenses, smart watches, privacy and security regulations, and other costs.

11.3 Project Date

The project would start on the 4th of May 2023 with 9 iterations and in each iteration, the project should meet the sprint goal. The regression testing and acceptance testing will be conducted in September. The projected go-live date is on 5th October 2023

11.4 Key Risks

- Data breach of patient records.
- Incorrect answers by the chatbot.
- Server down
- Unavailability of online doctor’s

12. REFERENCES

Websites:

[1] https://www.wrike.com/blog/how-carry-out-requirements-analysis/#What-are-the-benefits-of-a-requirements-analysis

[2] https://www.smartsheet.com/content/requirements-analysis

[3] https://www.researchgate.net/figure/Graph-showing-number-of-patients-booked-into-the-service-per-month-green-with-average_fig2_282913592

[4] What are Chatbot Slots and How to Define Them - discover.bot

[5] The Ultimate Guide for chatbot conversation flow - Rakebots blog

[6] New chat (openai.com)

[7] https://www.mindtools.com/abhkwcn/risk-analysis-and-risk-management

Videos:

[1] https://www.youtube.com/watch?v=-oD50HSBBBI