IN
PARTNERSHIP
WITH
PLYMOUTH
UNIVERSITY

Name: Priyantha Ranasinghe

Student Reference Number:10899343

Module Code: PUSL2021	Module Name: Computing Group Project – B2

Coursework Title: Referral C1 [Coursework]
Interim submission 01(report)

Deadline Date: 08/08/2024	Member of staff responsible for coursework: Mr.Chaminda Attanayake.

Programme: BSc(honours) Computer Security

Please note that University Academic Regulations are available under Rules and Regulations on the University website www.plymouth.ac.uk/studenthandbook.

Group work: please list all names of all participants formally associated with this work and state whether the work was undertaken alone or as part of a team. Please note you may be required to identify individual responsibility for component parts.

Names	Plymouth ID
Ranasinghe Ranasinghe	10903090
Priyantha Ranasinghe	10899343
Dodampe Nimna	10899233
Rankira Kosgollage	10899228
Rakitha Gunawardena	10899271

We confirm that we have read and understood the Plymouth University regulations relating to Assessment Offences and that we are aware of the possible penalties for any breach of these regulations. We confirm that this is the independent work of the group.

Signed on behalf of the group:

Individual assignment: ***I confirm that I have read and understood the Plymouth University regulations relating to Assessment Offences and that I am aware of the possible penalties for any breach of these regulations. I confirm that this is my own independent work.***

Signed:

Use of translation software: failure to declare that translation software or a similar writing aid has been used will be treated as an assessment offence.

I *have used/not used translation software.

If used, please state name of software………………………………………………………………

Overall mark _____% Assessors Initials ______ Date_________

*Please delete as appropriate

Computing Group Project - PUSL2021

Healthcare React Native Mobile Application – Final Report

(Referral assessment)

by

Group B2

A Project Report

Plymouth University

March 2024

Acknowledgements

We would like to express our sincere gratitude to all those who have contributed to the successful completion of the Healthcare React Native Mobile Application project. We extend our heartfelt thanks to our esteemed module leader, Mr. Chamindra Attanayake, for their invaluable guidance, support, and encouragement throughout this project.

We also extend our gratitude to our dedicated team members, whose unwavering commitment, collaboration, and hard work have been the key to bringing this project to life. Each member's unique skills and contributions have played a crucial role in overcoming challenges and achieving our goals.

Furthermore, we would like to thank Plymouth University and NSBM University for providing us with the opportunity to undertake this project, allowing us to apply our knowledge and skills in a real-world setting.

Lastly, we express our appreciation to our families and friends for their understanding, encouragement, and unwavering support throughout this journey.

Thank you all for being a part of this project's success.

Abstract

This project aimed to develop a healthcare application to address the challenges of healthcare accessibility and patient engagement. The main problem addressed was the limited access to healthcare services and resources, especially for individuals in remote or underserved areas. The objectives were made to design and implement a mobile application that integrates AI-driven chatbots to provide users with convenient access to healthcare services, appointment scheduling, medication reminders, and mental health support.

React Native was used to build the front end of the app, Node.js to manage the back end, MySQL to store data, and Dialogflow to implement the AI chatbot.

Key results of the project include the successful development and implementation of the healthcare application, demonstrating improved healthcare access and user engagement. The AI-powered chatbot effectively provided mental health support, while the application's user-friendly interface enhanced usability and satisfaction. There were some concerns which include keeping data secure and making the app work with other systems, which require further attention.

In conclusion, the healthcare application shows the potential to improve healthcare accessibility and patient engagement. Future work includes enhancing AI chatbot capabilities, integrating with emerging trends, and conducting further studies to determine long-term impact.

Table of contents

List of abbreviations 7
Chapter 1: Introduction 8
1.1 Purpose of the Project 8
1.2 Justification for the project 8
1.3 Scope & Objectives 9
1.4 Overview 9
Chapter 2: Literature review 10
2.1 Examination of the Body of Literature 10
2.2 Advantages and Disadvantages of Existing Systems 10
2.3 Comparison with Current Research 10
2.4 Theoretical Framework for the solution 11
2.5 Conclusion 11
Chapter 3: Methodology 12
3.1 Overall Approach 12
3.2 Work Breakdown Structure 12
3.2.1 Project Phases and Deliverables 12
3.2.2 Project Timeline 13
3.2.3 Gantt Chart 14
3.2.4 Resource Allocation 14
3.3 User Requirements 15
3.3.1 Identification of Users 15
3.3.2 Requirements Prioritization 15
3.3.3 User Interviews/Observations and Surveys 16
3.3.4 Use Case Analysis 18
3.3.6 Validation and Verification of the Findings 23
3.4 Functional Specifications 25
3.4.1 System Overview 25
3.4.2 Functional Requirements 25
3.4.3 Summary of Functional Specifications 27
3.4.4 User Communication: 28
3.4.5. Integration of Systems: 28
3.5 Technical Specifications 28
3.5.1 User Interface Design - UI/UX 28

3.5.2 Data Model - DFD/DB Design 30
3.5.3 Relational Schema 31
3.5.4 System Architecture 31
3.5.5 ER Diagram 32
3.5.6 Deployment and Infrastructure 32
3.5.7 Testing Strategies 33
3.5.8 Dependencies 34
Chapter 4: Results & Discussion 34
4.1 Results 34
4.2 Discussion 34
4.2.1 Achievements: 34
4.2.2 Challenges and Rectifications: 35
4.2.3 Future Improvements and Development Path: 35
Chapter 5: Conclusion and Future Works 35
5.1 Conclusion 35
5.2 Future Works 36
Reference list 38
Appendix 39
Contribution Matrix 39
Meeting Minutes 40

List of abbreviations

AI	**A**rtificial **I**ntelligence
NPL	**N**atural **P**rocessing **L**anguage
API	**A**pplication **P**rogramming **I**nterface
SQL	**S**tructured **Q**uery **L**anguage
JWT	**J**SON **W**eb **T**okens
JSON	**J**avaScript **O**bject **N**otation
GDPR	**G**eneral **D**ata **P**rotection **R**egulation
HIPAA	**H**ealth **I**nsurance **P**ortability and **A**ccountability **A**ct

Chapter 1: Introduction

In today's world, managing one's health can be a challenging task. From keeping track of medication intake to scheduling appointments and seeking emotional support, individuals face various challenges in maintaining their well-being. Recognizing these challenges, our project focuses on developing a Healthcare React Native Mobile Application aimed to address these concerns effectively through technological innovation. With the frequent use of smartphones and the growing demand for digital healthcare solutions, our application aims to provide users with a platform to manage their health effortlessly. By utilizing AI-powered chatbots, appointment booking systems, and access to relevant medical resources, our application seeks to improve how individuals engage with their healthcare needs.

Our project emphasizes key features such as medication management, emotional support through AI-driven chatbot, e-channeling, and access to emergency services. Through these features, we aim to encourage users to take control of their health while ensuring their safety and well-being.

Building upon the existing research and technological advancements in healthcare technology, our project seeks to contribute to the ongoing debates surrounding digital health solutions. By bridging the gap between technology and healthcare, our application aims to address pressing societal needs and improve the overall quality of healthcare delivery.

1.1 Purpose of the Project

Individuals usually struggle with controlling their health. Common concerns include forgetting to monitor medication usage, having difficulty making appointments on time, and a lack of emotional support. Our project's primary purpose is to develop a mobile healthcare application that uses technology to simplify and improve health management. The application will give users with options for monitoring their health, scheduling medical appointments, receiving emotional support, and accessing emergency assistance, addressing these common issues thoroughly.

1.2 Justification for the project

- Health Management Needs: The project addresses the demand for effective health management by providing tools for appointment scheduling and medication tracking.

- Technology Integration: The application enhances the user experience and involvement in healthcare through the use of AI and natural language processing.

- Emotional Support: AI chatbots offer mental health support, addressing a large social need.

- Efficiency in Healthcare Delivery: The program seeks to streamline healthcare operations, making it easier for consumers to monitor their health.

- Users have access to educational tools, which help them keep informed about health issues.

- Convenience: The app makes it easier to identify nearby medical providers and offers fast aid in an emergency.

1.3 Scope & Objectives

Scope: The project aims to create a detailed mobile healthcare application with the following features:

- AI-powered chatbot offers emotional assistance and health risk evaluations.
- Secure Data Sharing: Ensures the safe management of personal and medical information.
- Appointment Scheduling: Users can plan appointments with certain doctors.
- Emergency Services: Provides emergency contacts, location tracking, and medical information.
- Medication Reminders: This feature sends reminders to remind users to take their medications.
- Medical Content Access provides articles, videos, and educational materials about health concerns.

Objectives:

- Create an easy-to-use application that helps people manage their health risks and needs.
- Provide a safe platform for exchanging and managing personal and medical information.
- To make it easier to schedule appointments with healthcare providers.
- To offer emotional assistance with an AI-powered chatbot.
- Integrate emergency services to provide quick aid.
- To notify consumers about their medication intake.
- Provide relevant medical content and educational resources.

1.4 Overview

The later section of this report goes into further detail regarding the existing literature surrounding digital healthcare solutions in Chapter 2. Chapter 3 outlines the methodology employed in designing and implementing the application. Following this, Chapter 4 presents the results and discussion based on the outcomes of our project. Furthermore, Chapter 5 discusses the conclusions drawn from our work and proposes avenues for future research and development.

Chapter 2: Literature review

In this literature review, we explored existing research and projects related to healthcare applications which are developed using React Native. By analyzing the work of previous researchers and developers, we can have a full understanding of the current state of the topic and identify areas for the improvement of our own project.

2.1 Examination of the Body of Literature

Several studies and projects have explored the use of mobile applications in healthcare, with a particular focus on solutions developed using React Native technology.

Shashank Baranwal's project on "Healthcare Application Using React-Native" provides insights into the integration of corporate Medicare management systems, appointment scheduling, medical advice dissemination, and treatment recommendations.

Baranwal's project highlights the potential benefits of mobile technology in healthcare, such as improved access to medical services and enhanced communication between healthcare providers and patients.

2.2 Advantages and Disadvantages of Existing Systems

Baranwal's project underscores the advantages of mobile healthcare applications, including increased accessibility, streamlined communication, and improved healthcare outcomes. However, it also highlights several challenges, such as data confidentiality concerns, security issues, and the complexity of integrating with existing IT systems.

These insights provide us with valuable considerations for our project's development process, helping us to solve future obstacles while also enhancing the benefits of our application.

2.3 Comparison with Current Research

Our project aims to use findings of the research by implementing new features such as AI-powered chatbots for emotional support, medication management systems, and integration with emergency services.

While Baranwal's project focuses on corporate Medicare management, our project targets a greater audience by addressing the needs of patients, caregivers, and healthcare professionals alike.

By using React Native technology, we seek to create a versatile and user-friendly healthcare application that enhances access to medical services and promotes positive health outcomes

2.4 Theoretical Framework for the solution

Design and Development Theories:

- User-Centered Design Principles - We construct our applications using user-centered design principles to ensure that the interface and functionality are matched to the needs and preferences of the users. This strategy improves usability and engagement by prioritizing user experience from the start.
- Mobile Health Application Frameworks - We use well-established frameworks for mobile health apps, such as the Health Information Technology (HIT) standards, which provide principles for integrating healthcare data and assuring interoperability between systems.
- React Native Development Practices - React Native provides a solid framework for creating cross-platform applications, which is critical for ensuring that our product runs smoothly on both iOS and Android devices. The component-based architecture of React Native and its efficient rendering engine contribute to a smooth user experience and excellent

Justification of Theories:

- User-Centered Design - This idea is appropriate for our project because it directly affects customer satisfaction and application engagement. By focusing on the user experience, we ensure that the application is intuitive and accessible, meeting the different needs of our target population.
- Mobile Health Application Frameworks - These frameworks are critical for ensuring that our application meets industry standards and can seamlessly interface with other healthcare systems. This compliance is critical for protecting data and improving the application's operation.
- React Native Development Practices - React Native's cross-platform capabilities are justified by the requirement to reach a large number of people across multiple devices. Its performance benefits and fast development processes are consistent with our project's objectives of delivering a high-quality, scalable healthcare solution.

2.5 Conclusion

This literature study looks into the current status of mobile healthcare applications built with React Native, highlighting both their benefits and limits. Existing research, such as Baranwal's work, demonstrates benefits such as enhanced accessibility and communication while also highlighting issues such as data security and system integration. Our project expands on these insights by introducing innovative features such as AI-powered chatbots and efficient medication monitoring, thereby addressing larger needs and increasing user engagement. We hope to produce a powerful, user-friendly application that fulfills industry standards and enhances healthcare delivery by following user-centered design concepts and proven frameworks.

Chapter 3: Methodology

The methodology chapter shows the methods and strategies employed to gather and analyze data in our healthcare application project, providing readers with insight into the reliability of our findings and the robustness of our approach.

3.1 Overall Approach

- Our project's methodology involved using a mix of different techniques, tools, and materials to develop and implement the healthcare application.
- We used React Native for front-end mobile app development, due to its efficiency and cross-platform compatibility.
- The backend server, responsible for managing API interactions and business logic, was built using Node.js.
- MySQL served as the relational database management system for storing application data securely.
- Additionally, we used Dialogflow to implement an AI-powered mental health support chatbot, enhancing user engagement and accessibility.

3.2 Work Breakdown Structure

3.2.1 Project Phases and Deliverables

- Phase 1: Project Planning and Analysis (September)
  Deliverables:
    - Project Initiation and Scope Document
    - Requirement Gathering Report
    - Stakeholder Identification and Communication Plan
    - Initial Risk Assessment

- Phase 2: Design and Prototyping (October - November)
  Deliverables:
    - UI/UX Design
    - Database Design and Data Flow Diagrams
    - System Architecture Design
    - Prototype for Core Features (Registration, Medication Reminders, Appointment Booking)
    - Feedback and Approval from Stakeholders
- Phase 3: Development (December - February)
  Deliverables:
    - Front-end Development (React Native Implementation)
    - Back-end Development (Node.js and Express.js Implementation)
    - Database Integration (MySQL Setup and Configuration)
    - Integration of Dialogflow for Chatbot
    - Initial Testing of Core Features (Unit and Integration Testing)
    - Implementation of Security Measures (Encryption and Data Protection)

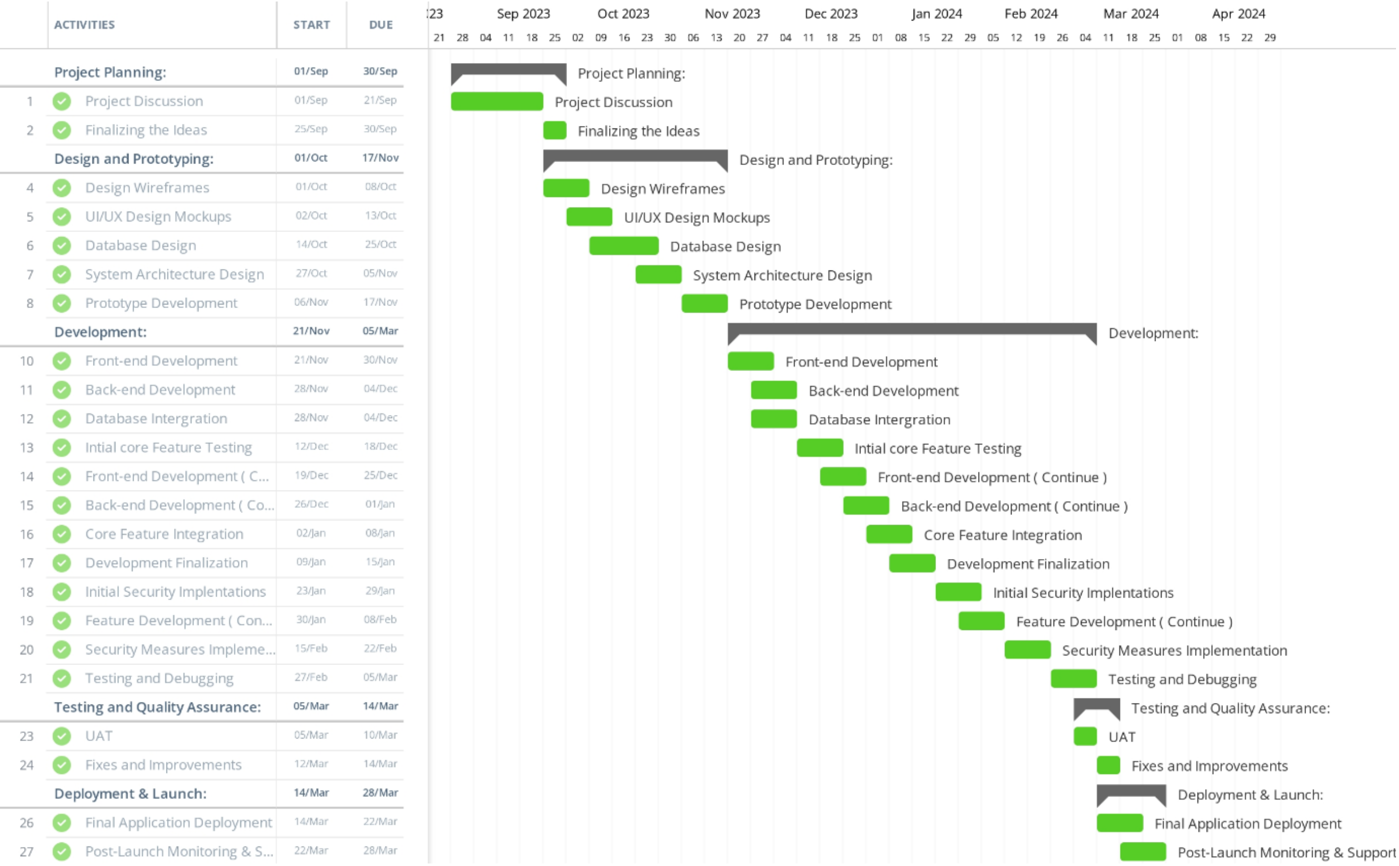
- Phase 4: Testing and Quality Assurance (March)
  Deliverables:
    - User Acceptance Testing (UAT)
    - Final Bug Fixes and Improvements

- Phase 5: Deployment and Launch (End of March)
  Deliverables:
    - Final Application Deployment
    - User Documentation and Support Materials
    - Post-Launch Monitoring and Support Plan
    - Project Closure Report and Lessons Learned

3.2.2 Project Timeline

Month				
September	Project Initiation	Requirement Gathering	Stakeholder Communication	Initial Risk Assessment
October	Design Wireframes	UI/UX Design Mockups	Database Design	System Architecture Design
November	Prototype Development	Prototype Feedback and Revisions	Finalize Design and Prototypes	Stakeholder Approval
December	Front-end Development	Begins Back-end Development	Begins Database Integration	Initial Core Feature Testing
January	Continue Front-end Development	Continue Back-end Development	Core Feature Integration	Initial Security Implementations
February	Development Finalization	Continued Feature Development	Final Testing and Debugging	Security Measures Implementation
March	End-to-End Testing Acceptance	Performance TestingUser	Testing (UAT) Final Bug	Fixes and Improvements
End of March	Final Deployment Preparation	Application Deployment	User Documentation and Support	Post-Launch Monitoring and Support

3.2.3 Gantt Chart



		ACTIVITIES	START	DUE
		Project Planning:	01/Sep	30/Sep
1	✓	Project Discussion	01/Sep	21/Sep
2	✓	Finalizing the Ideas	25/Sep	30/Sep
		Design and Prototyping:	01/Oct	17/Nov
4	✓	Design Wireframes	01/Oct	08/Oct
5	✓	UI/UX Design Mockups	02/Oct	13/Oct
6	✓	Database Design	14/Oct	25/Oct
7	✓	System Architecture Design	27/Oct	05/Nov
8	✓	Prototype Development	06/Nov	17/Nov
		Development:	21/Nov	05/Mar
10	✓	Front-end Development	21/Nov	30/Nov
11	✓	Back-end Development	28/Nov	04/Dec
12	✓	Database Intergration	28/Nov	04/Dec
13	✓	Intial core Feature Testing	12/Dec	18/Dec
14	✓	Front-end Development (C...	19/Dec	25/Dec
15	✓	Back-end Development (Co...	26/Dec	01/Jan
16	✓	Core Feature Integration	02/Jan	08/Jan
17	✓	Development Finalization	09/Jan	15/Jan
18	✓	Initial Security Implentations	23/Jan	29/Jan
19	✓	Feature Development (Con...	30/Jan	08/Feb
20	✓	Security Measures Impleme...	15/Feb	22/Feb
21	✓	Testing and Debugging	27/Feb	05/Mar
		Testing and Quality Assurance:	05/Mar	14/Mar
23	✓	UAT	05/Mar	10/Mar
24	✓	Fixes and Improvements	12/Mar	14/Mar
		Deployment & Launch:	14/Mar	28/Mar
26	✓	Final Application Deployment	14/Mar	22/Mar
27	✓	Post-Launch Monitoring & S...	22/Mar	28/Mar

3.2.4 Resource Allocation

(Work Distribution Among Members)

Group leader (Ranasinghe Ranasinghe) -

- In charge of overseeing the coordination of the entire project and making sure that all team members and leaders share the same objectives.
- manages the team's and stakeholders' communication, controls project development, and responds to any escalations.

Programming Leader (Priyantha Ranasinghe) -

- In charge of managing the front-end and back-end development and coding.
- guarantees the accuracy of the code, compliance with coding guidelines, and effective feature implementation.
- works in tandem with the Technical Leader to guarantee a flawless integration.
- Front-end Development (React Native)
- Back-end Development (Node.js and Express.js)
- Database Integration (MySQL)

Technical Leader (Rankira Kosgollage) -

- oversees the design and architecture of the system, making sure it is secure, scalable, and performs well.
- oversees the incorporation of external services, such as encryption libraries and Dialogflow.
- collaborates closely with the programming leader to ensure a smooth technical rollout.

- System Architecture Design
- Integration of Dialogflow for Chatbot
- Implementation of Security Measures (Encryption and Data Protection)

Quality Leader (Dodampe Nimna) -

- maintains the project's overall quality by adhering to standards and conducting regular evaluations.
- oversees the application of instruments and procedures for quality assurance.
- Code Reviews and Quality Checks
- Documentation Standards and Quality
- Feedback and Approval Processes

Testing and maintenance Leader (Ranasinghe Ranasinghe) -

- In charge of all testing tasks, including as user acceptability testing, end-to-end testing, integration testing, and unit testing.
- oversees the development and application of test cases, test results, and test strategies.

Planning Leader (Rakitha Gunawardena)-

- Oversees the scheduling, risk assessment, and planning of projects.
- Assures timely and scope-appropriate completion of all deliverables.
- Organises the team's and stakeholders' communication.
- Manages the project plan and timeline.

3.3 User Requirements

3.3.1 Identification of Users

- Patients and Caregivers: Primary users who will benefit from medication reminders, appointment scheduling, and health tracking.
- Mental Health Seekers: Users seeking emotional support through the AI-powered chatbot.
- Families: Users who may need to manage the health of multiple family members.
- Elderly People: Users who may need simplified interfaces and accessibility features.
- Healthcare Professionals: Users who will interact with the app for appointment management and telemedicine consultations.
- General Users: Individuals seeking health information and educational resources.

3.3.2 Requirements Prioritization

General Requirements:

- Medication Reminders - Ensure users take their medications on time with automated notifications based on the user’s medication schedule.
- Appointment Booking - Allows users to schedule and manage healthcare appointments easily with healthcare providers’ scheduling systems.
- Emergency Services - Enable users to quickly contact emergency services in case of a health crisis with a One-touch emergency call feature and location sharing.

Specific User Requirements:

- Personalized Health Profiles - Provide users with customized health management experience with user-specific data such as medical history, preferences, and health goals.
- Telemedicine Consultations - Offer virtual healthcare consultations to increase accessibility with secure audio calls with healthcare providers.

Additional Requirements:

- Health Education Resources - Provide users with information to improve their health literacy with articles, videos, and interactive content on various health topics.
- Emergency Assistance - Provide additional support in emergencies beyond basic services with features like location tracking for emergency responders.

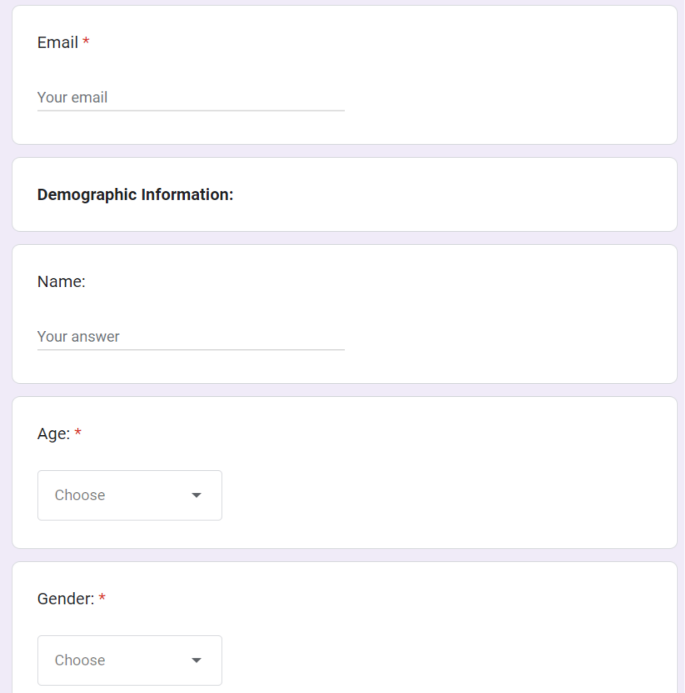
Continuous Prioritization:

- Stakeholder Needs - Regularly engage with stakeholders (users, healthcare providers, regulatory bodies) to gather feedback and identify emerging needs. This can be done by periodic surveys, focus groups, and usability testing sessions.
- Project Constraints - Evaluate constraints such as budget, timeline, and technical feasibility by regular project reviews and risk assessments.

3.3.3 User Interviews/Observations and Surveys

Google Forms was used for questionnaires, allowing users to respond at their own pace, providing us valuable insights and recommendations for further development of our project.

https://forms.gle/T52egF14Nr7cTS5fA



Email *

Your email

Demographic Information:

Name:

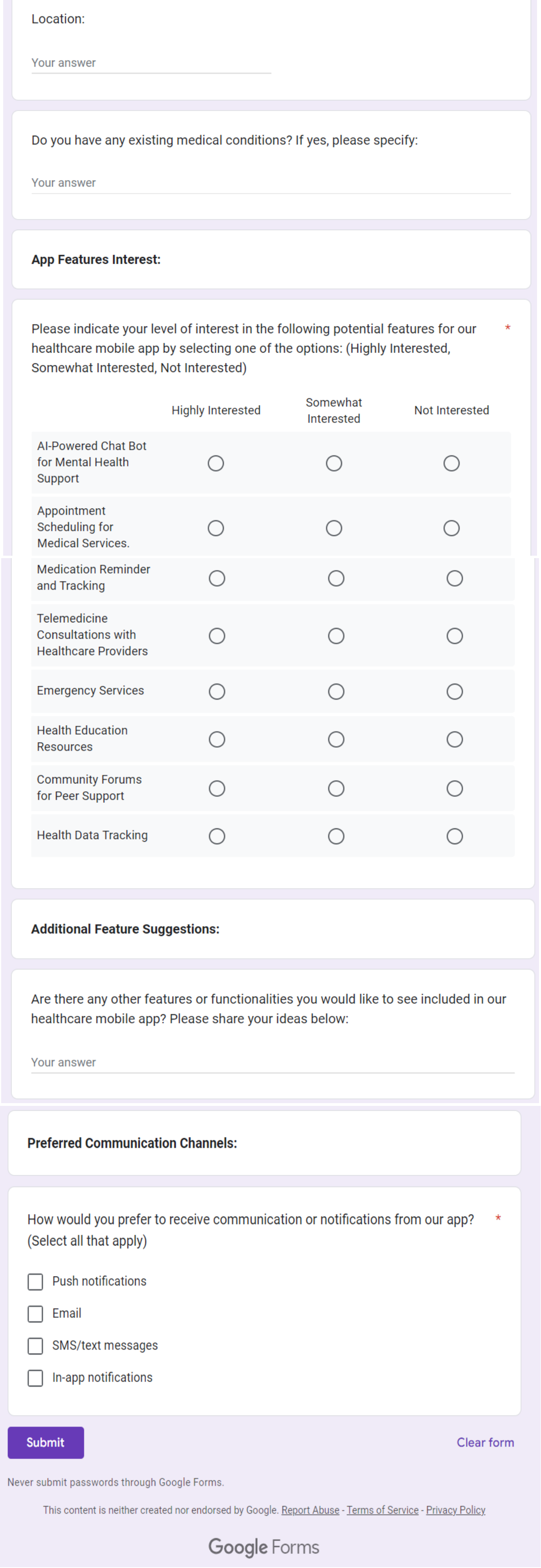
Your answer

Age: *

Choose

Gender: *

Choose



Location:

Your answer

Do you have any existing medical conditions? If yes, please specify:

Your answer

App Features Interest:

Please indicate your level of interest in the following potential features for our healthcare mobile app by selecting one of the options: (Highly Interested, Somewhat Interested, Not Interested) *

	Highly Interested	Somewhat Interested	Not Interested
AI-Powered Chat Bot for Mental Health Support	○	○	○
Appointment Scheduling for Medical Services.	○	○	○
Medication Reminder and Tracking	○	○	○
Telemedicine Consultations with Healthcare Providers	○	○	○
Emergency Services	○	○	○
Health Education Resources	○	○	○
Community Forums for Peer Support	○	○	○
Health Data Tracking	○	○	○

Additional Feature Suggestions:

Are there any other features or functionalities you would like to see included in our healthcare mobile app? Please share your ideas below:

Your answer

Preferred Communication Channels:

How would you prefer to receive communication or notifications from our app? (Select all that apply) *

□ Push notifications

□ Email

□ SMS/text messages

□ In-app notifications

Never submit passwords through Google Forms.

This content is neither created nor endorsed by Google. Report Abuse - Terms of Service - Privacy Policy

The above survey was used to gather user opinion on healthcare features, allowing for project enhancements based on user interest and feedback.

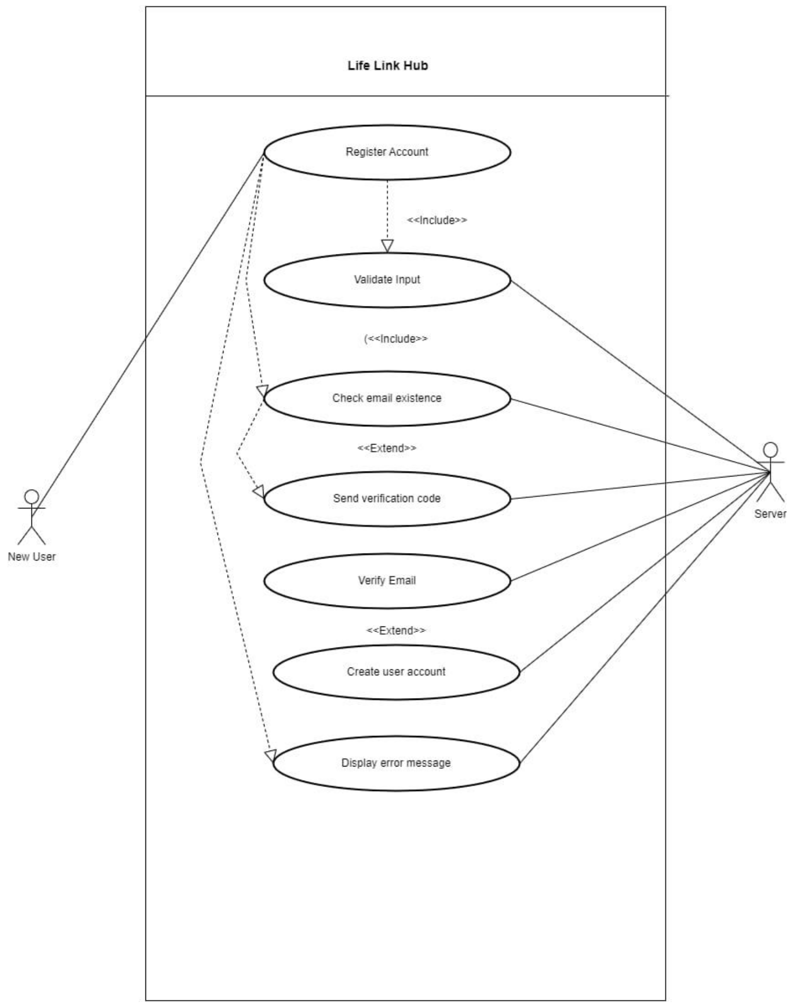
3.3.4 Use Case Analysis

1. Use Case 1: User Registration
   Actors: New User
   Description: A new user registers an account in the Life Link Hub app.
   Prerequisites:
   - The application is installed on the user's smartphone.
   - The internet is available to the user.

   Main Flow:
   - After launching the app, the user chooses the "Register" option.
   - The user inputs the necessary data (password, email, name, etc.).
   - The information entered is verified by the system.
   - A new user account is made by the system and is kept in the database.
   - After verifying the registration, the system takes the user to the login screen.

   Alternative flow
   - The system displays an error notice if the email entered is already registered.
   - The user is prompted by the system to amend any incorrect information entered.

1. User Registration

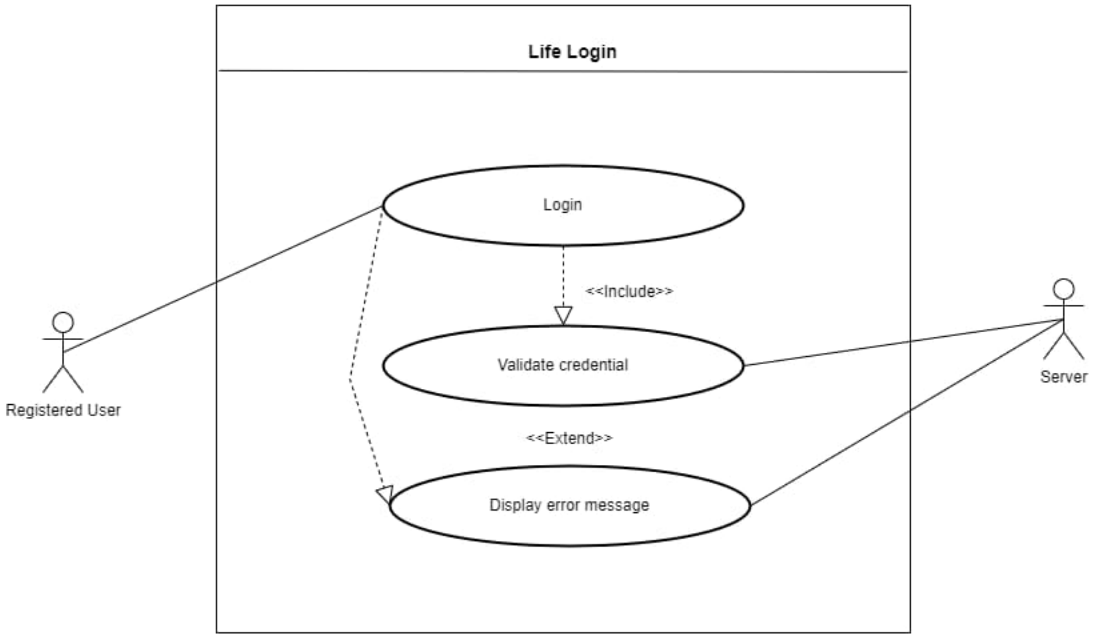


2. Use Case 2: User Login
   Actors: Registered User
   Description: A registered user logs into the Life Link Hub app.
   Prerequisites:
   - The user is in possession of an account.
   - The internet is available to the user.
   - Following conditions:
   - The application has the user logged in.
   - The home page is displayed to the user.

   Main Flow:
   - After opening the programme, the user chooses the "Login" option.
   - The user inputs their password and email address.
   - The credentials are verified by the system.
   - The user is redirected to the home page after being logged in by the system.

   Alternative Flow:
   - The system displays an error notice if the email address or password entered is invalid.

2. User Login



3. Use Case 3: Booking an E-Channeling Appointment
   Actors: User, Doctor
   Description: The e-channeling tool allows a user to schedule a doctor's appointment.
   Prerequisites:
   - The application has the user logged in.
   - The internet is available to the user.

   Following conditions:
   - The system records and books the appointment.
   - A notification of confirmation is sent to the user.

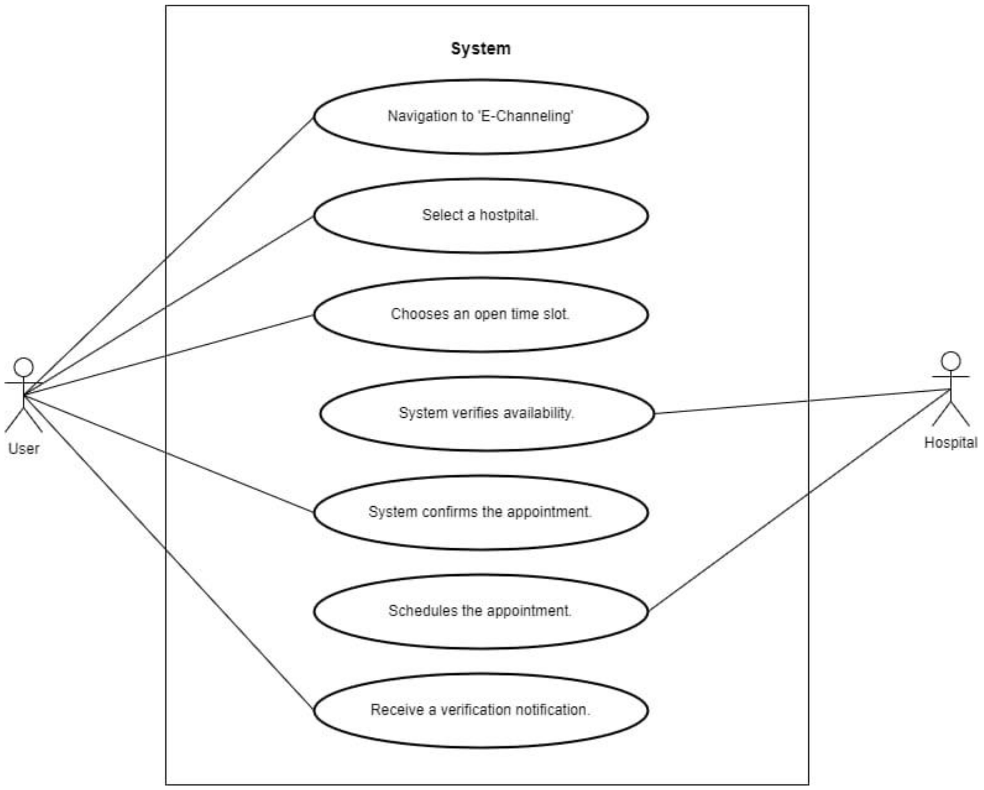
   Main Flow:
   - The "E-Channeling" component is reached by the user.
   - The user chooses a physician and a time slot that is open.
   - The chosen time window is verified to be available by the system.
   - The appointment is confirmed by the user.

- The appointment is scheduled by the system and is kept in the database.
- The user receives a confirmation notice from the system.

Alternative:

- The system notifies the user to choose an other time slot if the one they have selected isn't accessible anymore.

3. Booking an E - Channeling appoinment


System
Navigation to 'E-Channeling'
Select a hostpital.
Chooses an open time slot.
System verifies availability.
System confirms the appointment.
Schedules the appointment.
Receive a verification notification.
User
Hospital

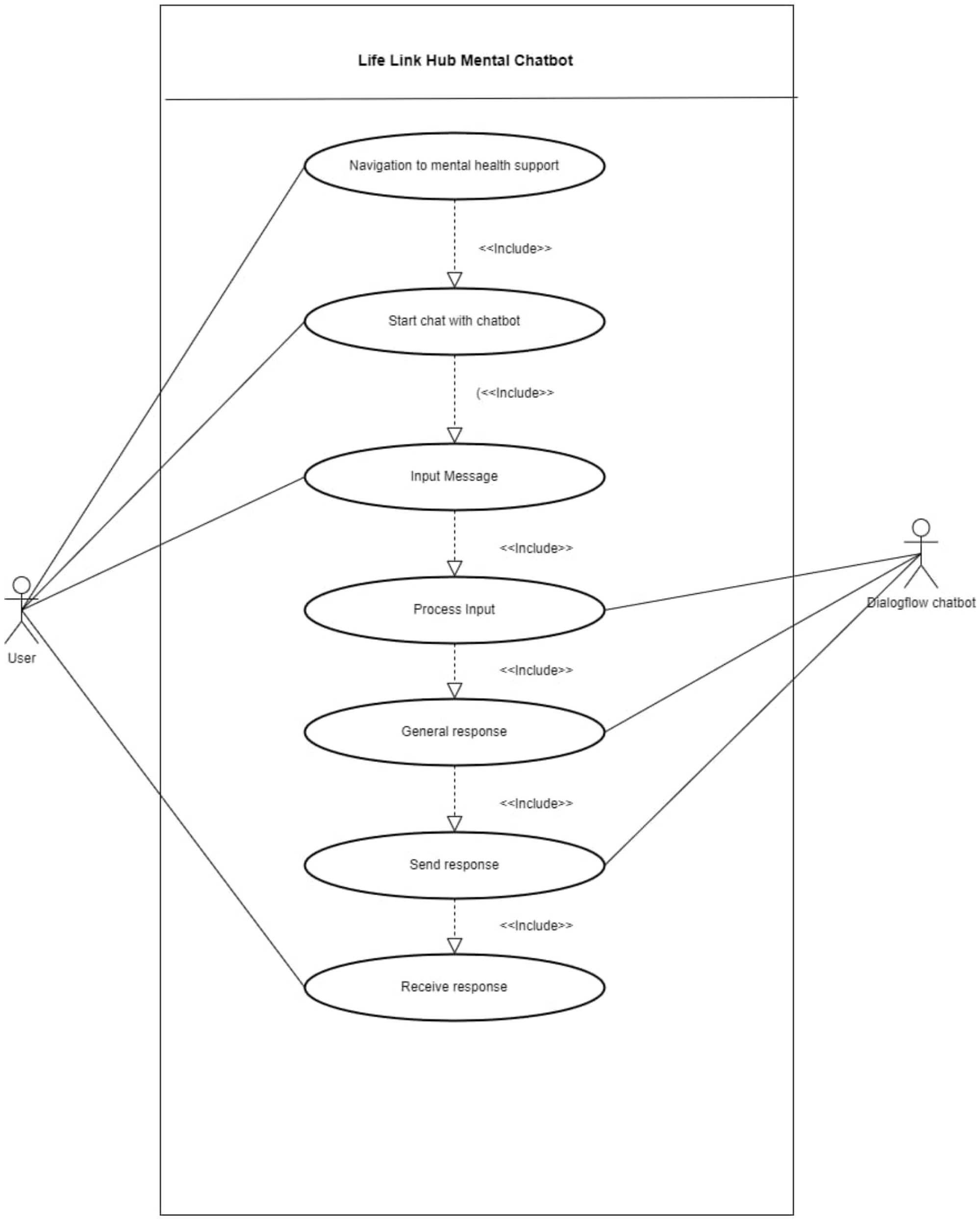
4. Use Case 4: Accessing Mental Support Chatbot
   Actors: User
   Description: To get advice and assistance, a user communicates with the chatbot for mental health.
   Prerequisites:
   - The application has the user logged in.
   - The internet is available to the user.

   Following conditions:
   - The chatbot responds to the user based on the information they provide.

   Main Flow:
   - The "Mental Support Chatbot" area is reached by the user navigating.
   - The user initiates a dialogue with the chatbot.
   - The user enters any queries or worries.
   - After analysing the information, the chatbot responds appropriately.
   - The user and the chatbot keep talking until the user closes the chatbot.

   Alternative Flow:
   - The user is prompted to restate the question or receives a generic response if the chatbot is unable to grasp the input.

4. Accessing Mental Support Chatbot



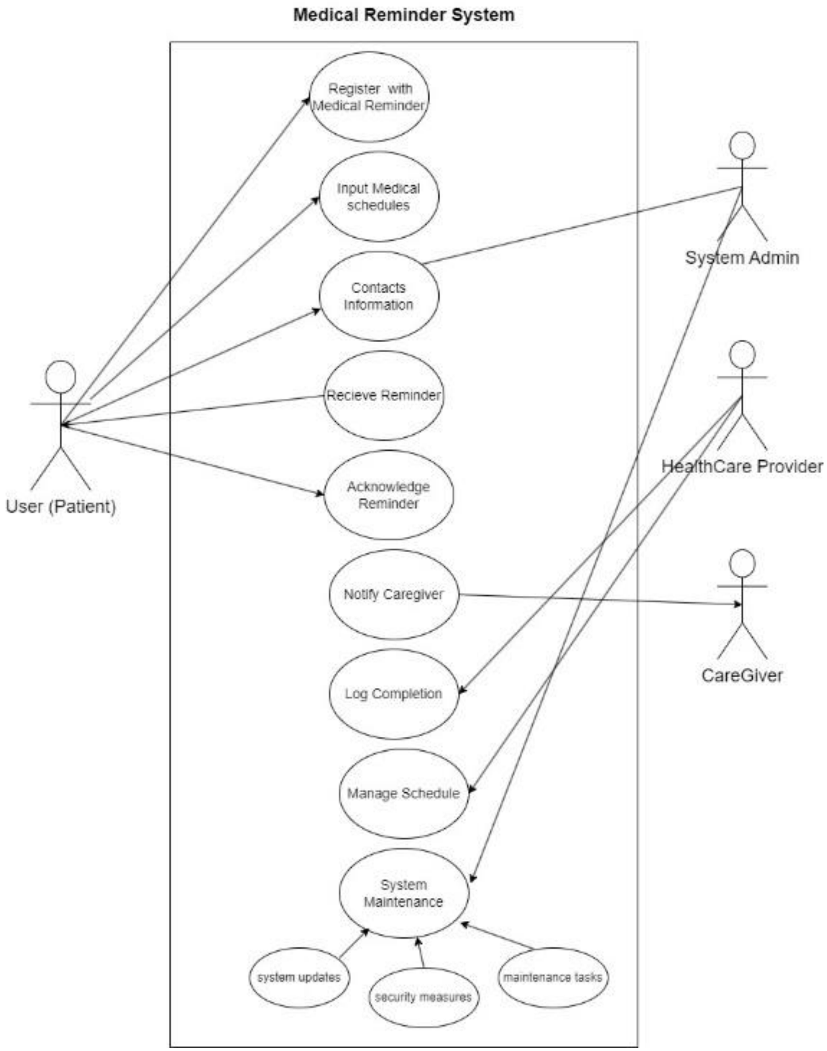
5. Use Case 5: Medication Reminder Setup
   Actors: User
   Description: A medicine reminder is put up by the user.
   Prerequisites:
   - The application has the user logged in.
   - The internet is available to the user.

   Following conditions:
   - The system is configured and contains the medication reminder.
   - Reminders are sent to the user at the designated times.

   Main Flow:
   - The "Medication Reminder" area is accessed by the user.
   - The schedule and medication data are entered by the user.
   - The information entered is verified by the system.
   - The reminder is set up by the system and is kept in the database.
   - Reminders are sent to the user by the system at the designated times.

   Alternative Flow:

- The system notifies the user to submit correct information if any is entered incorrectly.
- Detailed use cases and scenarios used for the project is shown below.



3.3.5 Funtional/ Non-Functional Requirements

Functional Requirements:

Functional requirements outline the behaviours and functionalities of the mobile application and define what it should be able to perform. The following are the primary functional requirements for our mobile application.

- User Registration and Authentication –
  Users should be able to securely create accounts and authenticate to access the application.
- Health Profile Management –
  Users should have the ability to input and update their health information, including medical conditions, allergies, and medications.
- Medication Reminders –
  The application should offer personalized reminders for users to take their medications at specific times.
- Appointment Booking –
  Users should be able to schedule appointments with specific doctors or healthcare providers based on their availability.
- AI-Powered Chatbot –
  providing emotional support, answering health-related questions, and offering guidance.
- Emergency Services Integration –

Users have access to emergency services features, including emergency call functionality, GPS location sharing, and health profile information sharing with emergency responders.

- Access to Medical Content –
  Users can access articles, videos, and educational materials relevant to different health conditions within the application.

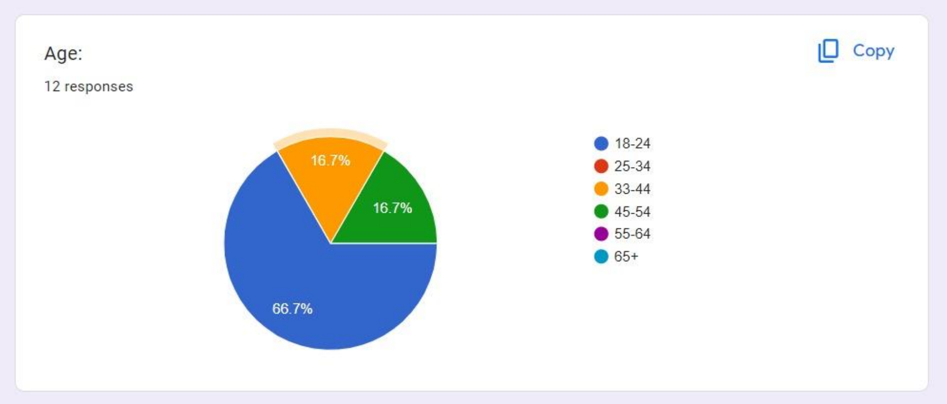
Non-Functional Requirements:

Non-functional requirements, which put the system's quality attributes front and centre, outline how the application carries out a certain task. The following are the primary non-functional criteria for our mobile application.

- Security and Privacy –
  Adhere to strict security measures to protect users' health information and ensure compliance with privacy regulations.
- Usability and Accessibility –
  User-friendly, with intuitive navigation and accessibility features to accommodate users with disabilities.
- Performance-
  Should perform efficiently, with minimal latency and downtime, to provide users with a smooth experience.
- Scalability –
  should be designed to handle a growing user base and increased data volume without compromising performance.
- Compatibility –
  should be compatible with various mobile devices and operating systems to reach a wide audience of users.
- Reliability –
  should be reliable, with robust error handling and data backup mechanisms to prevent data loss and ensure continuity of service.
- Interoperability –
  Integration with external systems, such as hospital databases or pharmacy networks, allows for seamless data interchange and service integration.

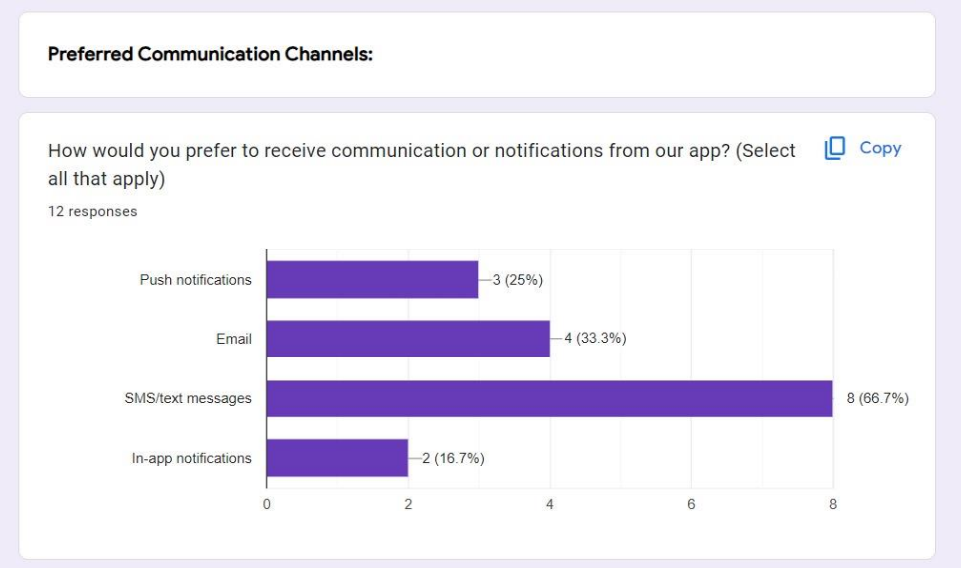
3.3.6 Validation and Verification of the Findings

We polled prospective users to get their opinions on the features, usability, and general satisfaction of the programme.
User experience, usability, functioning, and suggestions for improvement were all covered in the questionnaires.



Age:

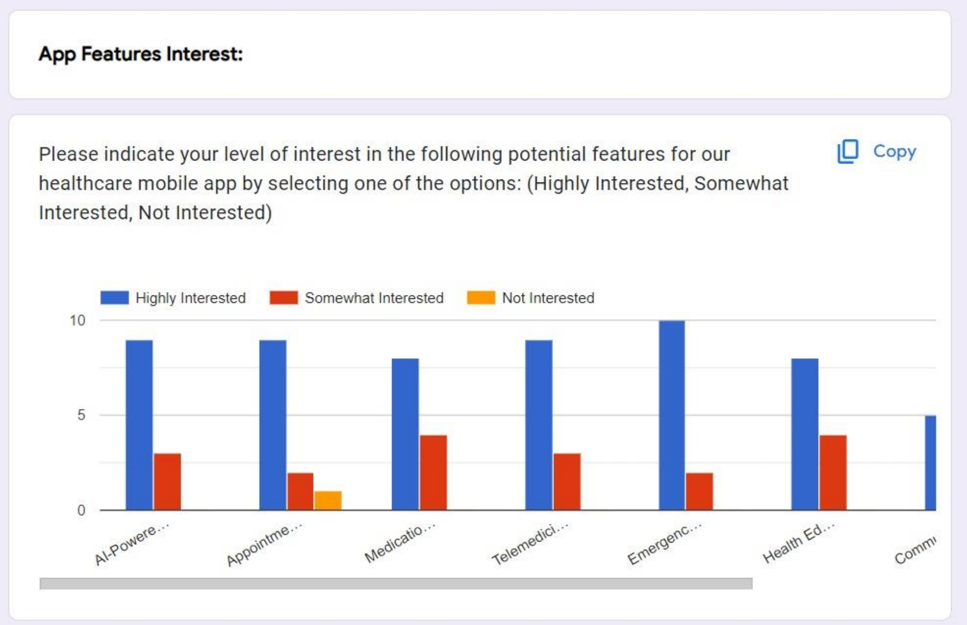
12 responses



Preferred Communication Channels:

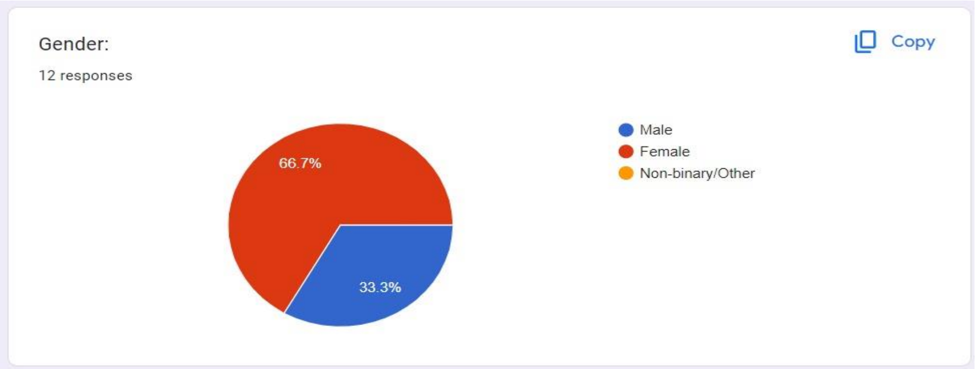
How would you prefer to receive communication or notifications from our app? (Select all that apply)

12 responses



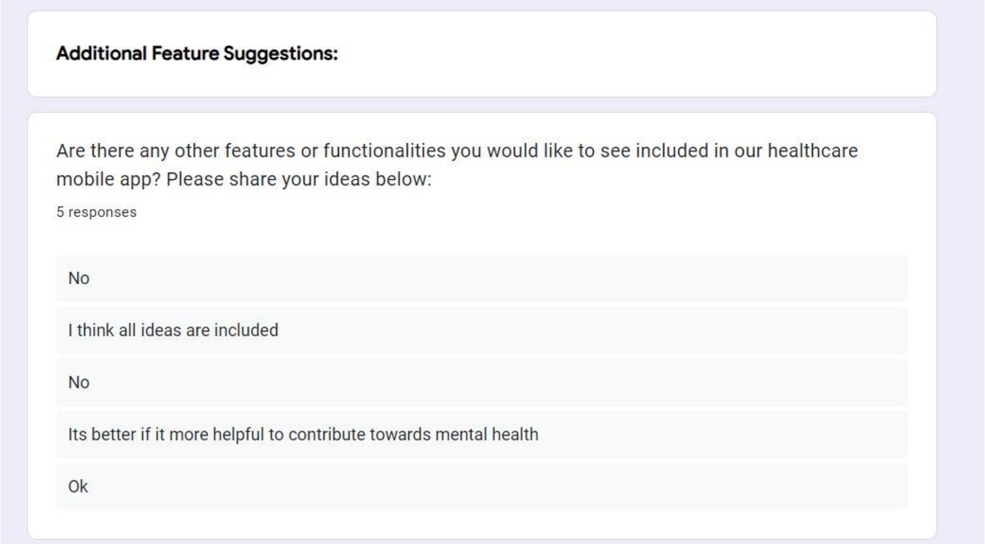
App Features Interest:

Please indicate your level of interest in the following potential features for our healthcare mobile app by selecting one of the options: (Highly Interested, Somewhat Interested, Not Interested)



Gender:

12 responses



Additional Feature Suggestions:

Are there any other features or functionalities you would like to see included in our healthcare mobile app? Please share your ideas below:

5 responses

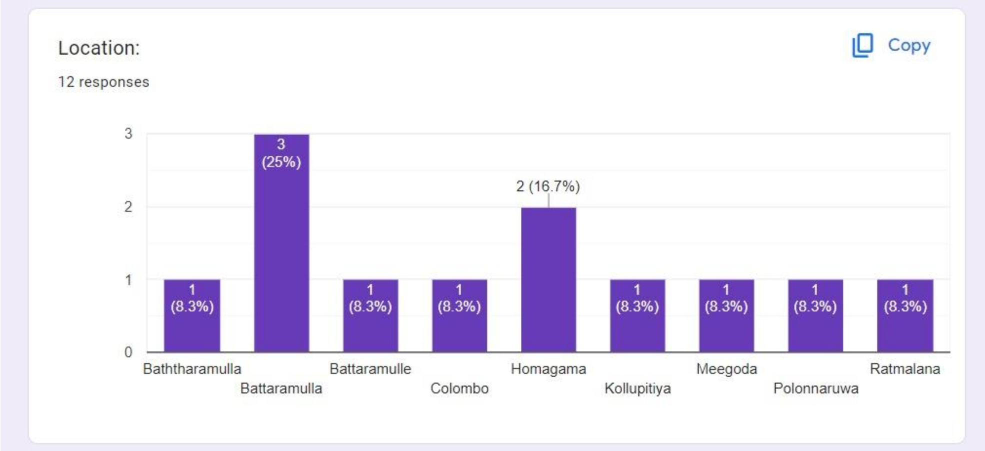
No

I think all ideas are included

No

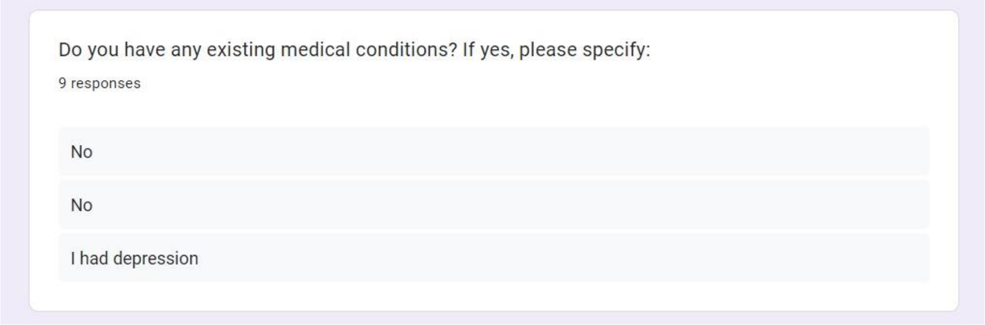
Its better if it more helpful to contribute towards mental health

Ok



Location:

12 responses



Do you have any existing medical conditions? If yes, please specify:

9 responses

No

No

I had depression

3.4 Functional Specifications

3.4.1 System Overview

The goal of the healthcare mobile application is to use contemporary technology to enhance accessibility and patient care. User registration, appointment scheduling, medication reminders, mental health support, emergency services, remote consultations, and medication management are some of the key features. MySQL is used for database management, React Native is used for the front end, Node.js is used for the back end, and Dialogflow is used to integrate chatbots.

3.4.2 Functional Requirements

1. Requirement ID: HR-001
   Requirement Description: Users must be able to register and manage their profiles, including personal information and notification preferences.

Dependencies: Integration with user authentication services and database management.
Acceptance Criteria:

- Users can register with personal details including name, email, phone number, and password.
- Upon registration, users receive an email containing a code.
- The account is activated, and the password is hashed using a secure algorithm (bycrypt) only after successful email verification.
- Users can log in with their credentials after successful registration and verification.
- Registration confirmation and login functionality are operational.
- Passwords are hashed using a cryptographic algorithm (bcrypt) before storing in the database.
- Users can update profile details and notification preferences.
- Changes are reflected immediately and accurately in the user profile.

Priority: High

2. Requirement ID: HR-002
Requirement Description: Implement an e-channelling feature for users to schedule, reschedule, or cancel appointments with healthcare professionals.
Dependencies:
    - Integration with doctor availability data from linked private hospitals.
    - Integration with notification systems for appointment confirmations and reminders.

   Acceptance Criteria:
    - Users can view doctor availability and schedules.
    - Users can book, reschedule, or cancel appointments.
    - Notifications are sent for confirmations and reminders.
    - Users can view and manage their appointments within the app.

   Priority: High

3. Requirement ID: HR-003
Requirement Description: Integrate a mental support chatbot using Dialogflow to provide mental health assistance and resources.
Dependencies: Integration with Dialogflow API and access to mental health resources.
Acceptance Criteria:
    - The chatbot conducts initial mental health assessments.
    - Provides appropriate responses and resources.
    - Escalates critical cases to professional help if necessary.

   Priority: High

4. Requirement ID: HR-004
Requirement Description: Provide automated medication reminders based on user schedules.
Dependencies: Database for storing medication schedules and integration with notification services.
Acceptance Criteria:
    - Users can input and manage medication details, including dosage and timing.
    - Automated reminders are sent according to the user’s schedule.
    - Users can view their medication history and upcoming reminders.

   Priority: High

5. Requirement ID: HR-005
Requirement Description: Include emergency services access with location sharing capabilities.
Dependencies: Integration with emergency contact systems and location services.
Acceptance Criteria:
    - Users can access emergency services with a single tap.
    - The app shares the user’s location with emergency responders.

- Confirmation of the emergency request is provided, including estimated response time.

Priority: High

6. Requirement ID: FR-006
   Requirement Description: Facilitate secure remote consultations with healthcare professionals via audio calls.
   Dependencies:Integration with a voice conferencing tool and secure data transmission protocols.
   Acceptance Criteria:
   - Users can schedule and initiate audio consultations.
   - Audio calls support high-quality sound.
   - Users can share medical documents during the consultation.

   Priority: Medium

7. Requirement ID: FR-007
   Requirement Description: Implement a medication management system to help users track and organize their prescriptions.
   Dependencies: Integration with the medication database and reminder services.
   Acceptance Criteria:
   - Users can add, edit, and delete medication records.
   - The system tracks medication schedules and generates refill reminders.
   - Users can view and manage their medication inventory and history.

   Priority: Medium

8. Requirement ID: HR-008
   Requirement Description: Implement data encryption for user data, both in transit and at rest, to ensure compliance with data protection regulations.
   Dependencies:Integration with encryption libraries and secure data transmission protocols.
   Acceptance Criteria:
   - Encryption at Rest - All sensitive data is encrypted using AES (Advanced Encryption Standard) with a key size of 256 bits before storing in the MySQL database.
   - Encryption in Transit - Data transmitted between the React Native app and the Node.js server is encrypted using TLS (Transport Layer Security) to ensure secure communication.
   - Regular Security Audits - Perform regular security audits using tools like Nmap to ensure encryption effectiveness and identify vulnerabilities.

   Priority: High

9. Requirement ID: HR-009
   Requirement Description: Implement a feedback system for users to rate and review their experiences with healthcare professionals and the app.
   Dependencies: Integration with feedback collection and reporting tools.
   Acceptance Criteria:
   - Users can submit ratings and reviews for healthcare professionals.
   - Feedback is collected and aggregated for analysis.
   - Users receive confirmation after submitting feedback.

   Priority: Low

3.4.3 Summary of Functional Specifications

- Hashing passwords during user registration and profile administration.
- Verification process.
- Scheduling appointments (e-channelling).
- Chatbot for mental health support.
- Reminders for medications.

- Access to emergency services.
- consultations conducted remotely.
- medication administration.
- features that make things accessible.
- Encrypting data.
- access to medical records.
- dashboard that can be altered.
- mechanism of feedback.

3.4.4 User Communication:

- Patient: Registering, maintaining profiles, setting up appointments, getting mental health help, getting medication reminders, using emergency services, taking part in remote consultations, and managing prescriptions are all tasks that patients can perform.
- Physicians: Schedule appointments, perform consultations, and access patient data.
- Administrators: Produce reports, manage users, and oversee system operations.

3.4.5. Integration of Systems:

- Front-end: User interface and interaction (React Native).
- Back-end (Node.js): Business logic and data processing.
- Database: Management and storage of data (MySQL).
- Dialogflow: Chatbot integration for mental wellness.
- Safe online consultations tool.
- Data protection with encryption libraries.
- JWT: Authorization and session management.

3.5 Technical Specifications

3.5.1 User Interface Design - UI/UX

Made sure the software is accessible, easy to use, and intuitive. To make an interface that is clear and responsive, apply contemporary design concepts.

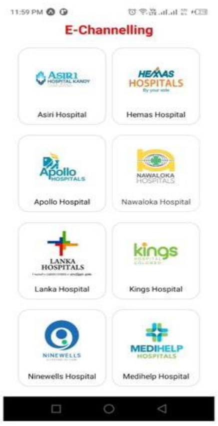
Components:

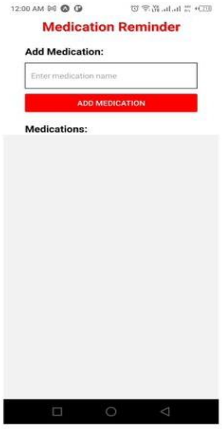
- Pages for logging in and registering: straightforward forms with feedback on validation and clear directions.
- Update your notification preferences and personal information in the profile management sections.
- Calendar display includes scheduling choices and availability for appointments.
- Chatbot Interface: Use the chat window to communicate with the chatbot that offers mental health support.
- Medication Management: Reminders and schedules for medications are tracked on this dashboard.
- Emergency Services: A location-sharing feature that allows for one-click access to emergency services.
- Remote Consultations: Document exchange possibilities and an audio call interface.
- Admin Dashboard: Interface for administrators to manage users, generate reports, and oversee system operations.

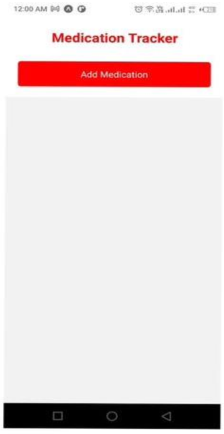
Tools & Technologies:

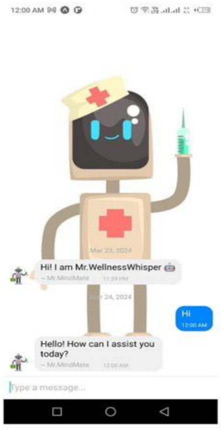
- React Native: For building cross-platform mobile application interfaces.
- Figma: For UI/UX design

Designing intuitive and user-friendly interfaces ensuring accessibility and ease of use.








Mar 23, 2024

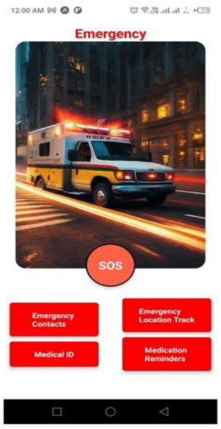
Hi! I am Mr.WellnessWhisper 🤖
~ Mr.MindMate 11:59 PM

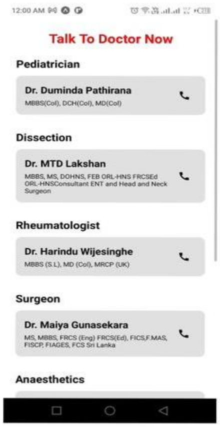
24, 2024

Hi
12:00 AM

Hello! How can I assist you today?
~ Mr.MindMate 12:00 AM

Type a message...





Pediatrician

Dr. Duminda Pathirana
MBBS(Col), DCH(Col), MD(Col)

Dissection

Dr. MTD Lakshan
MBBS, MS, DOHNS, FEB ORL-HNS FRCSEd ORL-HNSConsultant ENT and Head and Neck Surgeon

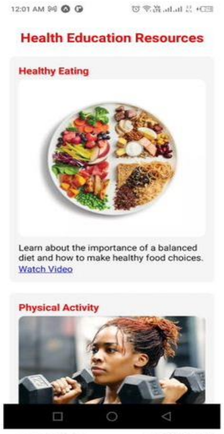
Rheumatologist

Dr. Harindu Wijesinghe
MBBS (S.L), MD (Col), MRCP (UK)

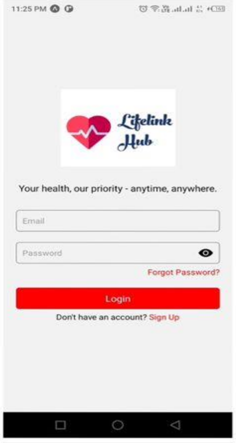
Surgeon

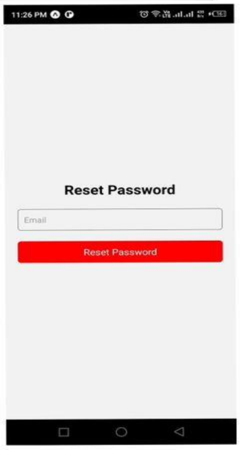
Dr. Maiya Gunasekara
MS, MBBS, FRCS (Eng) FRCS(Ed), FICS,F.MAS, FISCP, FIAGES, FCS Sri Lanka

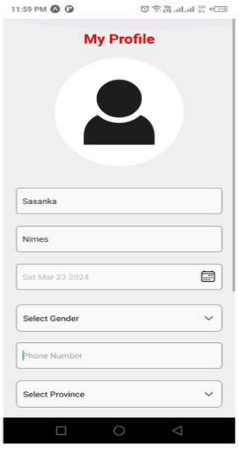
Anaesthetics



Learn about the importance of a balanced diet and how to make healthy food choices.
Watch Video


11:25 PM
Lifelink Hub
Your health, our priority - anytime, anywhere.
Email
Password
Forgot Password?
Login
Don't have an account? Sign Up





Sasanka

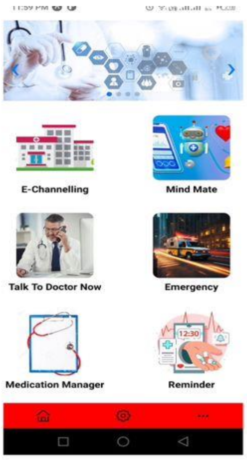
Nimes

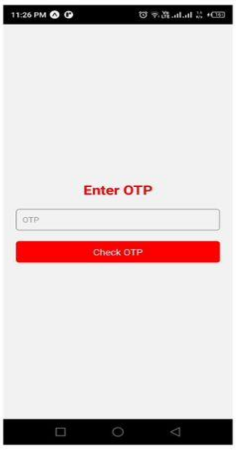
Sat Mar 23 2024

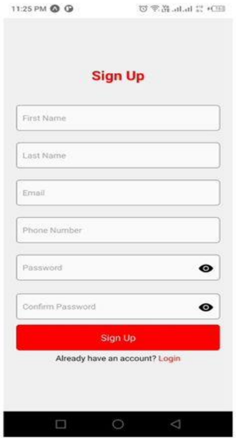
Select Gender

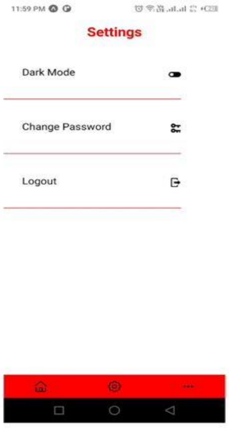
Phone Number

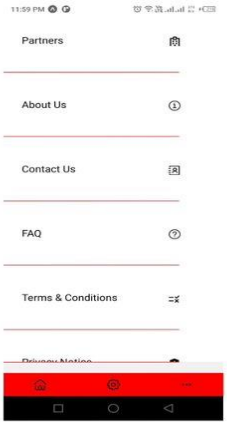
Select Province











3.5.2 Data Model - DFD/DB Design

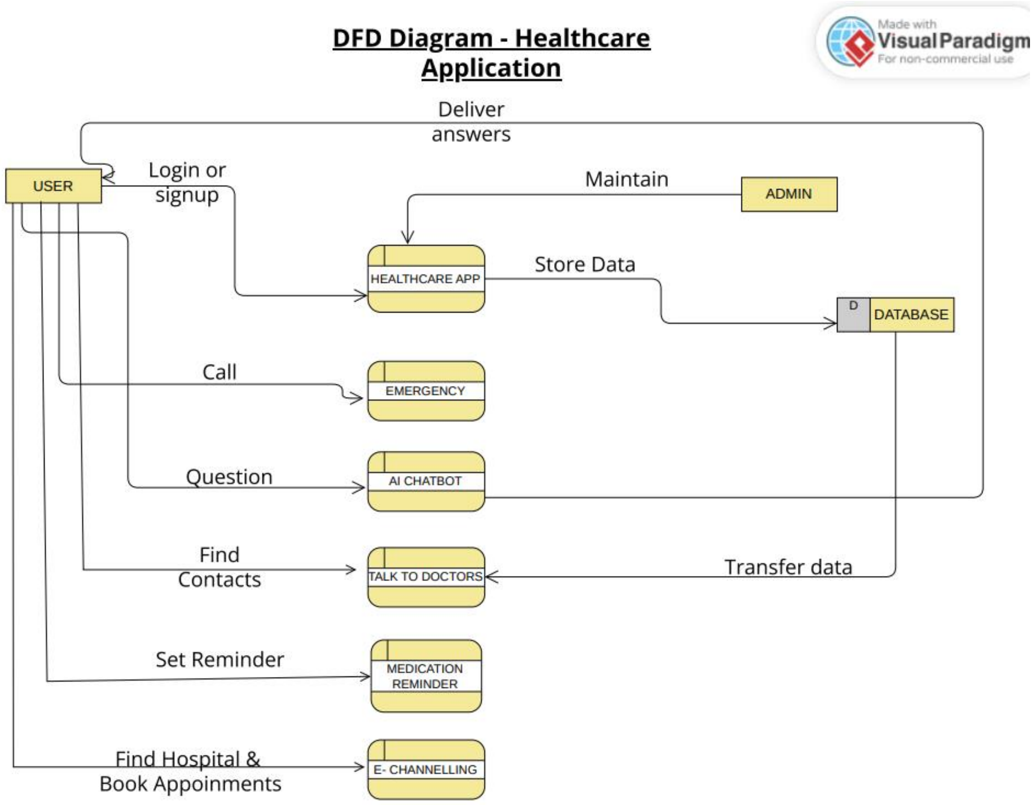
Explains the flow of data through the system, starting with user input and ending with database storage and retrieval.

Database Design:

- Tables - Users, Medications, Schedules, Requests for Emergencies, Comments, and Administrative Logs
- Connections - Establish the relationships between the tables, such as that users can make multiple appointments and administrators can provide reports.
- Normalisation - To cut down on redundancy and enhance data integrity, make sure the database design is normalized.

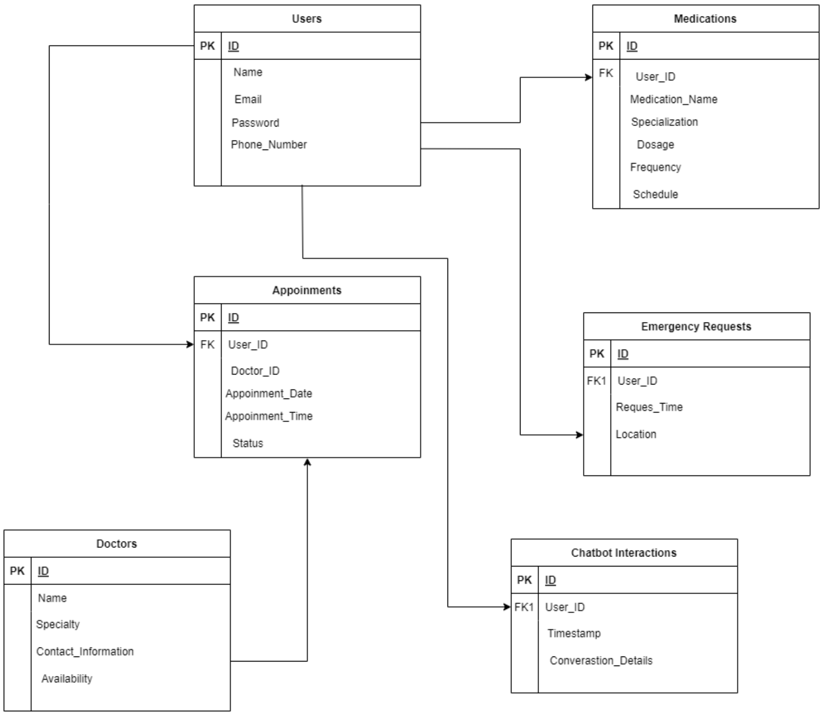
Tools & Technologies:

- MySQL - For managing database.
- MySQL Workbench - Used to create and visualize the database schemas.



3.5.3 Relational Schema

Below is the ER Diagram related to the project.



3.5.4 System Architecture

Description:

Overview - The architecture of the system is three-tiered:

- Presentation Layer - Utilising React Native, this layer provides the user interface for mobile apps.
- Application Layer - Node.js was used to implement server-side functionality and handle APIs.
- Data Layer - MySQL for storing and obtaining data.

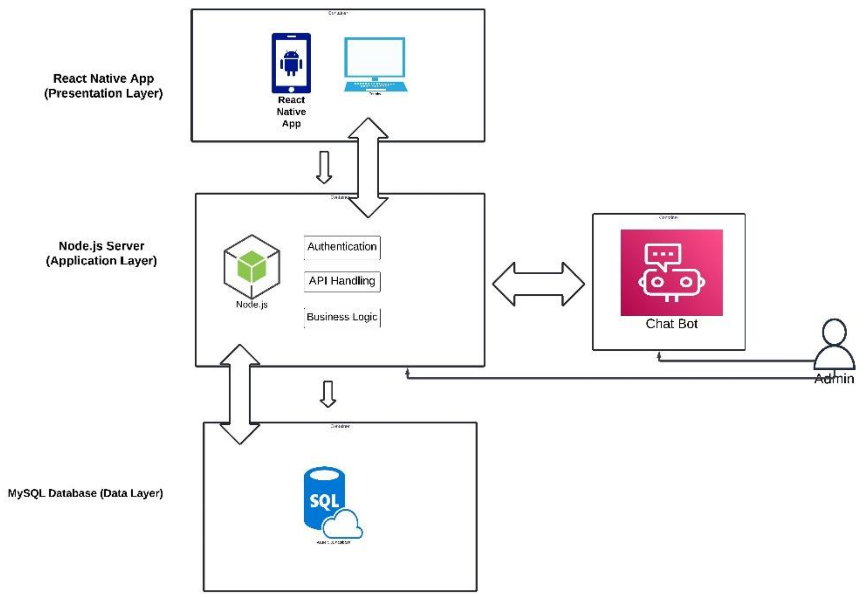
Parts:

- Front-end - For the user interface and experience, utilise React Native.
- Back-end - Node.js is used to manage user authentication, API interactions, and business logic.
- MySQL is the database used to store data.
- Dialogflow provides mental health support functionality through chatbot integration.
- Admin Module - For managing users and reporting, among other administrative duties.

Tools & Technologies:

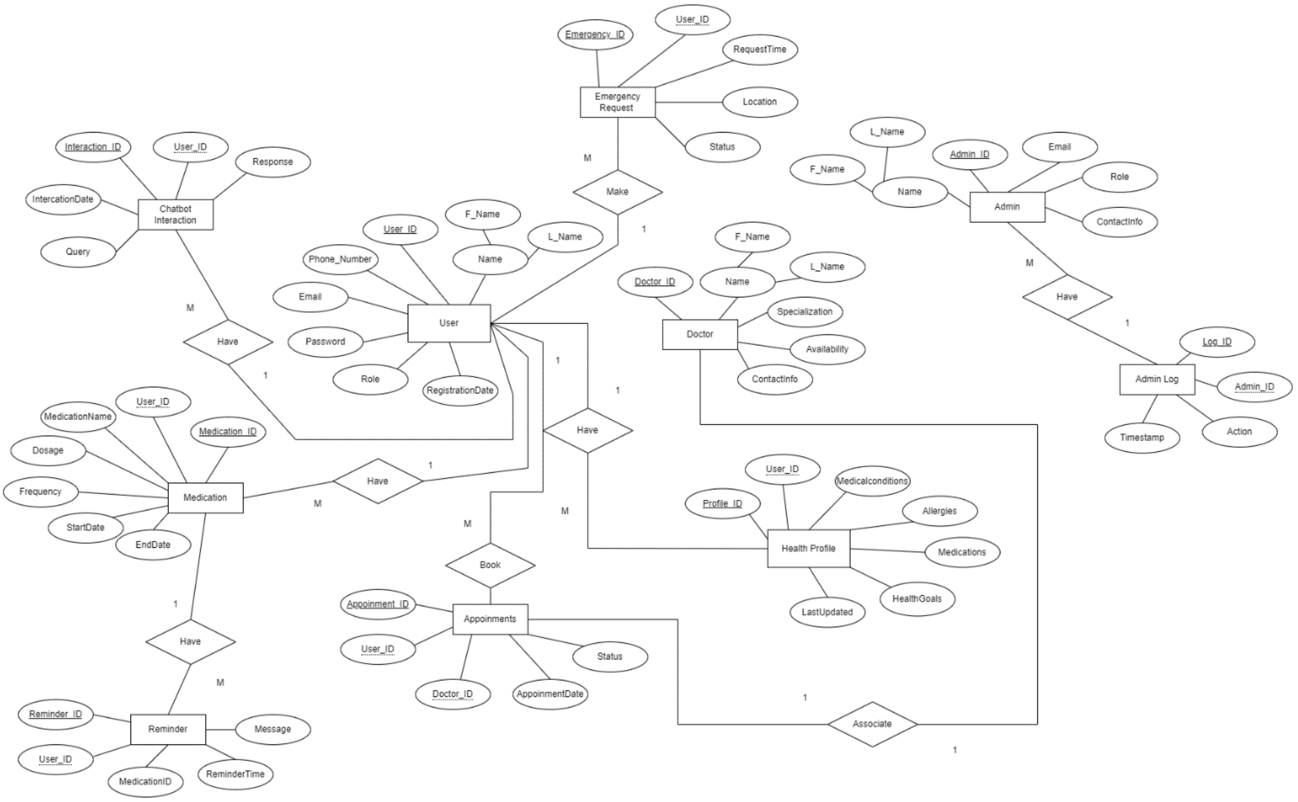
- Node.js - For server-side scripting.
- Express.js - For building RESTful APIs.
- Dialogflow - For chatbot integration.

system architecture



3.5.5 ER Diagram

Below is the Entity Relationship Diagram designed for our project



3.5.6 Deployment and Infrastructure

Hosting - Local infrastructure used on development machines to host the database and application.
Manual Deployment - For testing and demonstration reasons, manual deployment is carried out on local machines.
Scalability - In the event that a cloud-based infrastructure deployment becomes necessary, the system is built to be scalable.

Admin Access - Safe entry for administrators to control settings and supervise activities on the system.

Tools & Technologies -

- Local Development Server: Used to host the database and application locally.
- XAMPP: For configuring a local server environment in which the Node.js application and MySQL database are hosted.
- Git: For cooperation and version control.
- Procedure for Manual Deployment:
- Configure the development environment by installing the required programmes on local computers (e.g., MySQL, Node.js).
- Version Control: For collaboration and code change management, use Git.

Cloud Deployment in the Future - The architecture of the system is scalable, even if the current configuration relies on locally based infrastructure for testing and development. This guarantees that should the need arise in the future, a smooth transition to a cloud-based infrastructure, like Azure or AWS, may be completed. As a result, the programme will be able to manage growing user loads and data quantities effectively, offering stable and dependable performance as user needs expand.

3.5.7 Testing Strategies

A thorough testing approach was used to guarantee the healthcare mobile application's best quality and performance. This entails a variety of testing methods to cover every facet of the application, with a particular emphasis on cross-platform testing on both iOS and Android smartphones utilizing Expo Go.

- Cross-Platform Testing:
  Ensuring that the application runs well on Android and iOS gadgets.
  Tools: Expo Go
  How it was done - Expo Go streamlines the process of identifying and fixing platform-specific problems and offers a standardised testing environment.
  Using Expo Go's cross-platform features, it was confirmed that the program runs properly on a variety of screen sizes, resolutions, and operating system versions.

- Unit Testing :
  Testing certain parts or features to make sure they perform as intended.
  Tools: Expo Go
  How it was done- Create and run test cases to ensure that every component or function operates as intended.

- User Acceptance Testing (UAT):
  Test the application with actual users to make sure it satisfies their requirements and expectations.
  Tools: user feedback sessions.
  How it was done - Have in-person user sessions to get their input and determine whether the application satisfies their needs.

- Procedure for Testing:

  Planning Tests:
  - Specified the goals and parameters of the test.
  - Determined the instruments and resources needed.
  - Created thorough test cases and a test plan.

Conducting Tests:
- Followed the test plan when executing the test cases.
- Noted down any problems or errors that were discovered.

Adjusting:
- Made sure the bugs were fixed and prioritized before retesting to confirm the outcome.

Instruments & Technology:
- Sessions of User Feedback for UAT.
- Expo Go: To test on actual Android and iOS smartphones.

3.5.8 Dependencies

- Node.js - Used for server-side scripting and creating scalable network applications.
- Express.js - Used for building RESTful APIs.
- React Native - Used for developing cross-platform mobile applications for both iOS and Android.
- MySQL - Used to store and manage data such as user information, health records, and medication schedules.
- Dialogflow - Used to integrate mental health support functionality through a chatbot.
- Expo Go - Used to develop and test React Native applications on actual Android and iOS devices.
- bcrypt - Used for securely hashing user passwords before storing them in the database.
- TLS (Transport Layer Security) - Used to encrypt data transmitted between the React Native app and the Node.js server.
- AES (Advanced Encryption Standard) - Used to encrypt sensitive data before storing it in the MySQL database.
- Git - Used for version control to track changes in source code during development.
- XAMPP - Used to set up a local development environment with an Apache server and MySQL database.
- JWT (JSON Web Tokens) - Used for authorization and session management.
- MySQL Workbench - Used for designing and visualizing database schemas.

Chapter 4: Results & Discussion

4.1 Results

The healthcare application displayed strong performance and dependability. Key outcomes of testing are shown below

- User Access and Usability - Users were able to easily access medical services, make appointments, receive prescription reminders, and communicate with the AI-driven chatbot. This demonstrates the application's ability to meet its basic functionalities.

- Cross-Device Performance - thorough testing across several devices and operating systems (as described in Chapter 3, Section 3.5.6) verified that the application maintained high performance and reliability. This is critical for ensuring that users receive a consistent experience regardless of device.

4.2 Discussion

4.2.1 Achievements:

The initiative effectively attained its objectives.

- Functionality and Usability - By combining frontend and backend components, as well as an AI-powered chatbot, the program became more user-friendly and functional. This integration addressed a critical societal demand for accessible and individualized healthcare.
- Reliability - The application's performance across several platforms demonstrates its strength and dependability, which is critical for retaining user trust and happiness.
- User Engagement and Satisfaction - Updated methodologies, including usability testing and user feedback analysis, highlighted high levels of user satisfaction and engagement.

4.2.2 Challenges and Rectifications:

Several issues developed during the implementation phase and were appropriately addressed:

- Data Security and Privacy: - Keeping data secure was a huge concern. The application used powerful encryption mechanisms and secure authentication protocols to protect user data.
- Performance Optimization - To handle increased user traffic, approaches such as load balancing and efficient database administration were used. These measures ensured scalability and sustained performance under growing demand.

4.2.3 Future Improvements and Development Path:

- Enhance AI Chatbot - Improve conversational abilities and mental health support.
- Expand Device Integration - Support more wearables and health-monitoring devices.
- Optimize Performance - Implement advanced load balancing and database optimizations.
- Strengthen Data Security - Enhance encryption and stay updated with data protection regulations.
- Improve User Experience - Refine user interface and add personalized features.
- Increase Interoperability - Ensure compatibility with more healthcare systems and third-party services.
- Collaborate with Healthcare Professionals - Partner with experts for insights and validate features through clinical trials.
- Conduct Long-Term Impact Studies - Assess the app's long-term effectiveness and patient outcomes.
- Focus on User Feedback: Regularly incorporate user feedback to iteratively enhance features and address user concerns.

Chapter 5: Conclusion and Future Works

5.1 Conclusion

In conclusion, this study has demonstrated the successful development and implementation of the healthcare application aimed at improving healthcare accessibility and patient engagement. Through AI-driven chatbot technology and testing strategies, the application offers users convenient access to medical services, appointment scheduling, medication reminders, and mental health support.

The findings indicate the LifeLink Hub's potential to transform healthcare by offering personalized and efficient solutions to important societal needs. The AI chatbot, in particular, has demonstrated potential in providing mental health support, addressing an important facet of patient care. However, significant challenges, such as data security concerns and improved integration with existing systems, must be addressed in future releases.

Overall, the LifeLink Hub represents a major change in digital healthcare systems, displaying the ability to improve patient outcomes and accelerate healthcare delivery.

5.2 Future Works

There are also paths for future research and development based on this study.

1. Enhancing AI Chatbot Technology:

- Improving Conversational Abilities - Improve the AI chatbot's natural language processing ability to offer more accurate and sympathetic responses.
- Advanced Mental Health Support - Add more complex algorithms to detect and respond to mental health issues and consider collaborating with professional mental health providers for emergency assistance.
- Integration of Multilingual Capabilities - Implement multilingual support to cater to a more diverse user base.

2. Integrating With Wearable Devices:

- Remote Monitoring: Enable the application to support a broader choice of wearable devices and sensors for continuous health monitoring, such as fitness trackers, smartwatches, and medical sensors.
- Data Utilization: Use wearables data to deliver individualized health insights and proactive health management suggestions.

3. Collaboration With Healthcare Professionals:

- Clinical Validation: Collaborate with healthcare specialists to validate the application's features via clinical trials, confirming the app's efficacy and dependability.
- Expert Input: Continuously incorporate medical professionals in the development process to ensure the application stays to the most recent medical standards and practices.

4. Data Security and Privacy Enhancements:

- Advance Encryption Techniques: Implement more complex encryption techniques to improve data security.
- Compliance with regulations: Maintain continuing compliance with data protection rules such as GDPR and HIPAA and adapt to new privacy legislation as they arise.

5. Expanding Service Categories

- Telehealth Services: Include more comprehensive telehealth services such as video consultations, remote diagnostics, and electronic prescriptions.
- Health Education: Increase the number of educational options available through the app, such as webinars, interactive courses, and tailored health advice.

6. Long-Term Impact Studies:

- Patient Outcomes: Conduct long-term research to determine how the application affects patient health outcomes and overall healthcare quality.
- User Engagement: Analyse user engagement data to better understand usage patterns and find areas for improvement.

7. Performance Optimization:

- Scalability Enhancements: Use advanced load balancing and database optimizations to ensure that the application can handle additional user traffic effectively.

- User Experience Refinements: To increase user pleasure and engagement, continuously refine the user interface and add tailored features.

By exploring these additional initiatives, the LifeLink Hub will be able to grow and answer emerging healthcare requirements while also using technological advancements to improve healthcare delivery and patient outcomes. This study lays the groundwork for future innovation in digital healthcare, with the potential to have a long-term influence on global health and well-being.

Reference list

BARANWAL, S. (2020) 103.47.12.35, HEALTHCARE APPLICATION USING REACT-NATIVE. Available at: http://103.47.12.35/bitstream/handle/1/1914/1613101671_SHASHANK%20PAL_Project%20Report%20-%20SHASHANK%20PAL.pdf?sequence=1&amp;isAllowed=y (Accessed: 20 March 2024).

World Health Organization (WHO) World Health Organization. Available at: https://www.who.int/ (Accessed: 15 February 2024).

MITTAL, A. (2019) Build a chatbot: Dialogflow with react native, Jscrambler. Available at: https://jscrambler.com/blog/build-a-chatbot-with-dialogflow-and-react-native (Accessed: 18 March 2024).

Educative, Inc. (2024) How to create a new react native project, Educative. Available at: https://www.educative.io/answers/how-to-create-a-new-react-native-project (Accessed: 19 March 2024).

ASIRI HEALTH (2024) Enjoying life to the fullest, Asiri Health. Available at: https://asirihealth.com/ (Accessed: 10 March 2024).

Nawaloka Hospitals (2024) Nawaloka Hospitals plc, Nawaloka Hospitals PLC. Available at: https://www.nawaloka.com/ (Accessed: 10 March 2024).

Hemas Hospital (2024) Hemas hospitals, Sri Lanka - official website, Hemas Hospital -. Available at: https://hemashospitals.com/en/ (Accessed: 10 March 2024).

Apollo Hospitals - International Care Center Sri Lanka (2023) Apollo Hospitals - International Care Center Sri Lanka | Multispeciality Hospitals in India For all Your Healthcare Needs - Bringing healthcare of International standards within the reach of every individual. Available at: https://apollohospitals.lk/ (Accessed: 10 March 2024).

Lanka Hospital (2024) Lanka hospitals sri lanka - official website, Lanka Hospitals. Available at: https://www.lankahospitals.com/ (Accessed: 10 March 2024).

Kings Hospital, Kings Hospital colombo, Kings Hospital Colombo. Available at: https://www.kingshospital.lk/ (Accessed: 10 March 2024).

Ninewells Hospital (2020) NINEWELLS, Ninewells Hospital. Available at: https://www.ninewellshospital.lk/ (Accessed: 10 March 2024).

Medihelp Hospital, Medihelp Hospitals. Available at: https://www.medihelp.lk/ (Accessed: 10 March 2024).

Appendix

Contribution Matrix

Names	Plymouth ID	Work Distribution
Ranasinghe Ranasinghe	10903090	- Front-end Development - Report – Future works and conclusion (chapter 5), Diagrams
Priyantha Ranasinghe	**10899343**	- **Back-end development (Backend integration, Database management, API development, Server-side logic)** - **Front-end Development** - **UI/UX design** - **Report – Introduction (chapter 1), Methodology(chapter3)**
Dodampe Nimna	10899233	- Front-end development - Chatbot development - Report – Results and Discussions (Chapter 4), Diagrams
Rankira Kosgollage	10899228	- Front-end development - UI/UX design - Chatbot development - Report – Literature Review (chapter 2), Methodology (chapter 3), Results and Discussions (Chapter 4)
Rakitha Gunawardena	10899271	- Front-end development - UI/UX design - Report – Future works and conclusion (chapter 5), Diagrams

Meeting Minutes

Meeting date	Description of what members talked during the meeting
2023/09/05	Discussed project scope and objectives. Evaluated stakeholder needs and initial requirements. Assigned tasks for stakeholder communication and risk assessment.
2023/09/12	Reviewed project charter and detailed requirements. Assigned initial tasks for stakeholder communication and risk assessment. Updated each other on progress and insights.
2023/09/19	Discussed design concepts and UI/UX principles. Decided on wireframes and initial design mockups. Shared progress on individual design tasks.
2023/09/26	Evaluated feedback on design mockups. Finalized UI/UX design and identified necessary design changes. Updated each other on design revisions and feedback incorporation.
2023/10/03	Discussed database schema and data flow diagrams. Outlined system architecture and integration plans. Members updated on progress of their respective tasks.
2023/10/10	Finalized system architecture and database design. Discussed integration of Dialogflow for the chatbot. Shared updates on backend development and integration progress.
2023/10/17	Reviewed progress on the design and prototyping phase. Discussed challenges and adjustments needed. Team members updated each other on development and prototyping tasks.
2023/11/07	Discussed development progress for front-end and back-end. Identified integration issues and planned solutions. Each member provided updates on their development progress.
2023/11/21	Evaluated the implementation of core features. Planned next steps for security measures and continued development. Team members reported on feature implementation and integration status.
2023/12/05	Reviewed implementation of Dialogflow. Discussed improvements and features for chatbot integration. Members updated on progress of the chatbot and associated features.
2023/12/12	Discussed unit and integration testing results. Identified bugs and planned fixes. Team members shared individual testing results and coordinated on fixing identified issues.
2023/12/19	Reviewed security implementation. Discussed encryption methods and data protection strategies. Updated each other on security measures and encryption progress.
2024/01/09	Planned and conducted performance testing. Reviewed testing results and addressed performance issues. Team members provided updates on performance testing and optimization tasks.
2024/01/23	Prepared for end-to-end testing. Discussed scenarios and testing strategies. Members updated each other on their end-to-end testing preparations and scenarios.
2024/02/06	Conducted User Acceptance Testing (UAT). Collected feedback on usability and functionality from users. Discussed individual UAT results and plans for addressing feedback.
2024/02/20	Discussed feedback from UAT. Planned and prioritized final bug fixes and improvements. Team members updated on their respective bug fixes and enhancements.
2024/03/05	Reviewed deployment strategy and checklist. Finalized plans for launch and post-launch support. Members provided updates on deployment preparations and final checks.
2024/03/19	Discussed final preparations for launch. Reviewed project closure activities and final documentation. Team members updated on final tasks and project documentation status.