COMSATS University Islamabad,

Attock Campus, Attock Pakistan

Intellectual System For E-Health

By

Shahid Iqbal **CIIT/SP20-BCS-008/ATK**

Supervisor

Mr. Umar Zia

Bachelor of Science in Computer Science (2020-2024)

The candidate confirms that the work submitted is their own and appropriate credit has been given where reference has been made to the work of others.

COMSATS University Islamabad,
Attock Campus, Attock Pakistan

Intellectual System for E-Health

A project presented to
COMSATS University, Islamabad

In partial fulfillment
of the requirement for the degree of

Bachelors of Science in Computer Science (2020-2024)

By

Shahid Iqbal CIIT/SP20-BCS-008/ATK

DECLARATION

I hereby declare that this software, neither whole nor as a part has been copied out from any source. It is further declared that we have developed this software and accompanied report entirely based on our efforts. If any part of this project is proven to be copied out from any source or found to be a reproduction of some other, we will stand by the consequences. Portioning of the work presented has been submitted of any application for any other degree or qualification of this or any other university or institute of learning.

Shahid Iqbal

CERTIFICATE OF APPROVAL

It is to certify that the final year project of BS (CS) **“INTELLECTUAL SYSTEM FOR E-HEATH”** developed by **SHAHID IQBAL (CIIT/SP20-BCS-008)** under the supervision of **MR. UMAR ZIA** and co-supervisor **MR. WASEEM KHAN** and that in his opinion; it is fully adequate, in scope and quality for the degree of Bachelor of Science in Computer Sciences (BSCS).

Supervisor

External Examiner

Head of Department

(Department of Computer Science)

Executive Summary

Chronic disorders including diabetes, Symptom Based disease, and heart disease are becoming more common, which has a substantial effect on public health. Early detection and prompt action might be extremely important to properly manage chronic conditions. Machine learning models and algorithms have demonstrated considerable promise in estimating the likelihood of developing certain diseases, allowing for early diagnosis and care. The intended smartphone app promises to offer users a simple and convenient platform to evaluate their risk of getting diabetes, Symptom based disease, or heart disease. The app analyses multiple health data and offers tailored risk evaluations using machine learning algorithms and models.

The software takes variables including age, gender, blood pressure, cholesterol levels, and smoking history when predicting heart disease. The software takes variables including age, body mass index (BMI), blood pressure, and fasting glucose levels when predicting diabetes. Accurate risk projections are made possible by the machine learning models utilized in the app, which were trained on enormous datasets of health and medical data. The software offers users personalized risk assessments. Using machine learning algorithms and models, the suggested smartphone app offers a viable method for predicting the likelihood of getting diabetes, Symptom based disease, and heart disease. The software has the potential to enhance the early detection and management of many chronic diseases, improving patient quality of life and health outcomes.

DEDICATION

This all work which is done by us is dedicated to our parents and Respected Teachers, without their prayers and effort we could not able to do anything.

Acknowledgment

In the name of ALLAH, the most merciful and beneficent. I'm grateful to the ALLAH Almighty who provides all the resources of every kind to me, so that I make their proper use for the benefit of mankind. May He keep providing me with all the resources, and the guidance to keep helping the humanity.

I would like to thank our parents who kept backing us up in all the times, both financially and morally. I'm thankful to them for their prayers.

I would also like to thank my supervisor **Mr. Umar Zia** for his guidance and encouraging me to work hard and smart and for his moral support to accomplish our project. His critical comments on my work made me think of new ideas and techniques in the field of optimization and software development. I have found him very helpful while discussing the optimization issues.

I'm very thankful to COMSATS University Islamabad, Attock Campus for giving me enough knowledge and skills that made me too innovatively.

Shahid Iqbal

Abbreviations

GDPR	General Data Protection Regulation
PHI	Personal Health Information

Table of Contents

List of Tables

List of Figures

1. Introduction

Nowadays, everything is available through the Internet. The purpose of constructing this project called “Disease Prediction Using Machine Learning” is to predict the accurate disease of the patient using all their general information the symptoms. The system will predict the following Diseases i.e. (Diabetes Disease, Heart Disease). This method will predict the foremost possible disease supported by the symptoms system will predict diseases for using based on symptoms. System’s great concern is to send the necessary information to patients at the right time while ensuring the accuracy and trustworthiness. Healthcare issues can be solved efficiently by using Machine Learning Technology.

The Algorithm will used are i.e. (Decision Tree, SVM, Logistic regression, Random Forest). ML model allows us to build models to get quickly cleaned and processed data and deliver results faster. The Datasets used for prediction of diseases are Disease-Symptom Knowledge Database, Disease Prediction Using Machine Learning, Heart Disease Dataset, Diabetes Dataset, Symptom based Disease prediction datasets, also Medicine Information system. This system is supposed to give information of any medicine/drug on the basis of the search result. By using this system doctors will make good decisions related to patient diagnoses and according to that.

1.1. Vision Statement

The keyword template below is useful for creating a product vision statement:

- **For** [all people, offices, schools, and homes]
- **Who** [patients or normal]
- **The** [Smart E-Health]
- **Is** [Prediction system]
- **That** [Predict Diseases on the base of symptoms and tests]
- **In contrast to** [current Health care websites and applications, which provide services, but do not give any recommendation OR prediction],
- **Our system** [user can use for prediction of diseases and get information about medicine and treatment also Health, Diet, Exercise information]

1.2. Related System Analysis/Literature Review

There are many existing systems to resolve the issue based on the data privacy, medicine and relevant recommendations, but all the previous approaches suffer from heavy computation as well as communication complexity. The existing system predicts the chronic diseases which are for a particular region and for the particular community. In this System, Big Data & CNN Algorithm is used for Disease risk prediction.

Table 1 Related System

Applications Name's	**Weakness/flaws**	**Solution of the proposed project**
The name of the associated application (s).	Weaknesses also can furthermore encompass a loss of abilities and low-outstanding capabilities and processes.	How the proposed challenge addresses the flaws.
National library of Medicine Administrative and medical records Health Care Data Standards	In these systems there are some weakness and flaws. • User and professional of Heath have some kind of difficulties. • In these system there is no recommendation /prediction. • There is no prediction for user or the health professionals.	We will use some techniques for the solution of these flaws. • An App having proper interface, easy to use. • Prediction of multiple Disease. • Heart Disease prediction. • Symptom based prediction • Diabetes Disease prediction.

	• medicine related no recommendation. Information. • No proper App	• Search all type of medicine with all about information/ uses. • User profile building and maintaining. • Health, Diet, Exercise Information.

1.3. Project Deliverables

There are some general deliverables, and the specific deliverables may vary based on the project's requirements and scope.

- **Project Plan:** A thorough project plan that details the project's goals, deadlines, and scope.
- **Data Gathering**: Gather pertinent information about the diseases from a variety of sources, including medical background, lifestyle choices, genetic data, and other crucial parameters.
- **Data Preprocessing:** Prepare the gathered data for machine learning algorithms by cleaning, filtering, normalizing, and transforming it.
- **Feature Selection:** Choose the preprocessed data's most pertinent characteristics that can have a big impact on disease prediction.
- **Model Building:** Create machine learning models to forecast the occurrence of diseases using a variety of algorithms, including logistic regression, decision trees, neural networks, etc.
- **Validate:** The developed models using a variety of performance metrics, including recall, accuracy, precision, F1-score, and others, and compare the outcomes with those of current models.
- **Deploy**: The developed models to a user-friendly interface, such as a smartphone application.
- **Testing and Debugging:** Thoroughly test the developed system and fix any bugs or mistakes that are discovered.
- **Project documentation:** Include data sources, preprocessing procedures, feature selection, model construction, validation, and deployment in the documentation.

- **User Manual:** Write a user guide that describes how to operate the system and decipher the outcomes.
- **Upkeep:** Maintain and assist the system, including updating the models and resolving any problems that may occur.

1.4. System Limitations/Constraints

Disease forecast systems are no different from other machine learning systems in that they all have restrictions or limitations. The diabetes, heart disease, and **Symptom based** disease prediction algorithms commonly have the following drawbacks and restrictions:

- **Limited predictive power:** Disease prediction systems have a range of limitations that can reduce their prognostic power. There is always a chance that a person could develop the disease even if it is not predicted to, and they can only make educated guesses about the likelihood of a disease based on the facts at hand.
- **Limited data availability:** Disease prediction systems depend on data to train their models, but data isn't always readily available. However, in some instances, the data may be scarce, resulting in a model that is less precise.
- **Variability in symptoms**: Diabetes, heart disease, and symptom-based disease prediction can differ greatly between people, making it difficult to create a model that correctly forecasts the course of the disease.
- **Limited demographic representation:** Age, gender, and race may not be taken into consideration by some disease prediction models. For some populations, this may lead to inaccurate forecasts.
- **Ethical issues**: The possibility for discrimination based on predicted disease risk is one ethical issue with disease prediction systems.

Overall, it is important to keep in mind that disease prediction algorithms have constraints and limitations when creating and using them. To create more precise and dependable prediction models, it is crucial to be aware of these constraints and strive to overcome them.

1.5. Tools and Technologies

We are currently planning to with following tools and technologies.

Table 2. Tools and Technologies

	Tool’s	Version’s	Rationale
Tools Technologies	Visual Studio VS Code	17.3 2022	IDE
	Firebase	8.0 2022	DBMS
	React Native	18	Development
	Technology	**Version**	**Rationale**
	React	18	Development
	Python	3.10	AI/ML
	Java Script	ES2015	Back-end Development

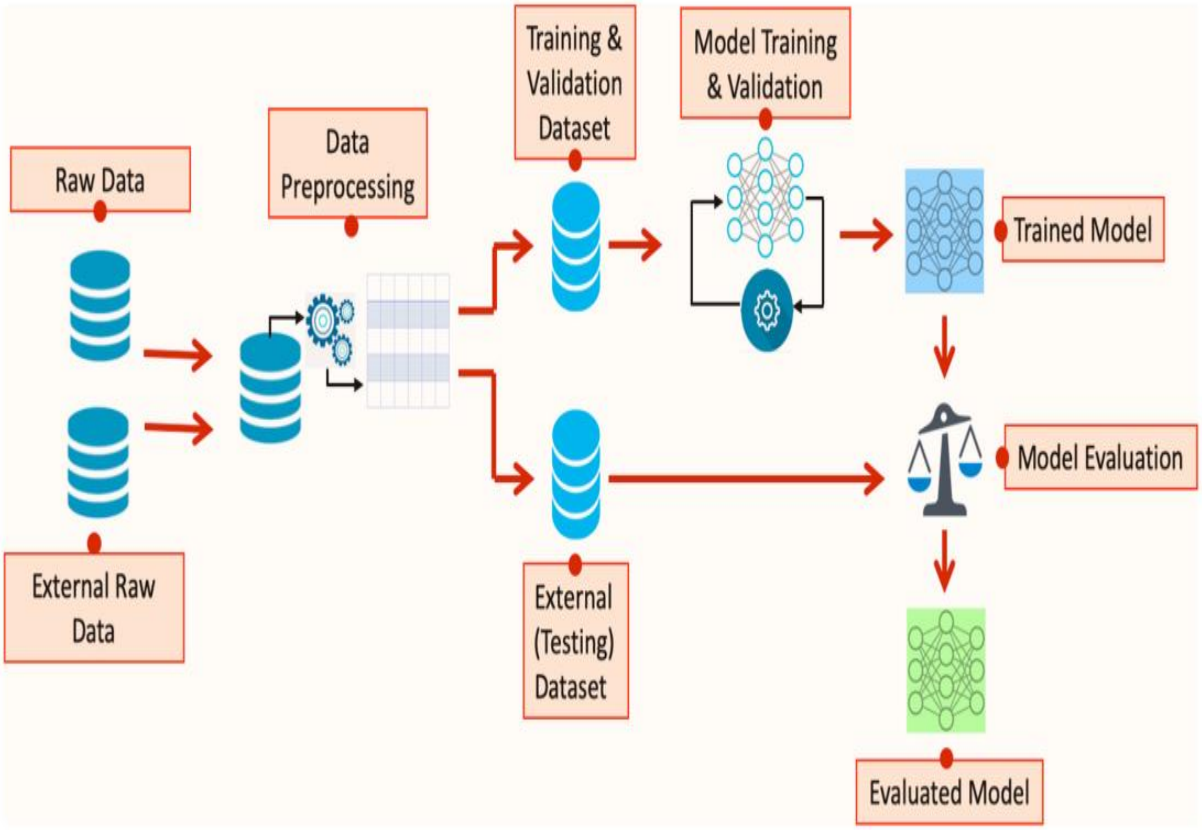
1.6. Relevance to Course Modules

My project is relevant to our courses i.e., Programming Fundamentals, Object Oriented Programming, Software Engineering, Machine Learning, Mobile Development. I am using JavaScript for the front-end development of the application and for back-end implementation I use a python language for Prediction.

I represent the state of work of the application by drawing different diagram i.e., Architecture diagram, Activity Diagram, Sequence Diagram and Use Case Diagram the courses that boost our designing skills are Software Engineering, Introduction to Software Engineering, Human computer Interaction, Software Design and Architecture.

The writing skill of the document about project skills is develop by English Comprehension and Composition, Report writing Skills, Communication skills and Professional Practices for IT.

1.7. Methodology:



2 Problem Definition

One of the most important aspects of the economy and of human life is medicine and healthcare. The world we live in today and the world from a few weeks ago have seen a great deal of change. Everything has changed to be horrifying and bizarre. This is not practicable for everyone who is in bed in this circumstance, including during in-person appointments. Diabetes is one of the deadliest diseases in the world, among other ailments. Additionally, it is the creator of a number of different types of illnesses, including cardiac failure, blindness, diseases of the urinary system, etc. In this situation, the patient must go to a diagnostic facility to receive their reports following consultation. They have to devote their time and money each time because of this. Unknowingly or not, this serious sickness affects people all around the world. Diabetes can also lead to other conditions like heart attacks, paralysis, kidney problems, blindness, etc.

To create a predictive model that can precisely identify those who are at risk of contracting certain diseases is the challenge facing the disease prediction system. As a result, patients would receive early detection and intervention, which would improve the management and treatment of their ailments. In order to create a predictive model based on different risk factors, including age, gender, medical history, lifestyle, and genetic predisposition, the system will employ machine learning algorithms to analyses patient data. The risk that a person would get one of these diseases will then be predicted using the model. The Diabetes, Heart, and Symptom Based Disease prediction system has the potential to revolutionize how we approach the diagnosis and treatment of these widespread and debilitating disorders by utilizing the power of machine learning and sophisticated analytics.

2.1. Problem Statement

The goal of creating a diabetes, heart, and Symptom based disease prediction system is to give medical practitioners a tool to help them identify those who are most at risk of contracting these diseases. The software system will forecast a patient's risk of getting certain diseases based on their medical history, way of life, and genetic characteristics using a variety of data analytic approaches. In order to analyses patient data and make highly accurate predictions, this software system will use machine learning algorithms. This will allow healthcare professionals to intervene early and adopt efficient preventative measures. To help patients effectively manage their diseases and minimize future consequences, the goal is to provide early detection and preventative techniques. Early identification and prevention are essential to improve patient outcomes and lower healthcare costs because of the increasing prevalence of these diseases and the high cost of treatment.

2.2. Proposed Solution

A computerized programme called the Disease Prediction System attempts to forecast the risk that a person will get one of these three diseases. In order to estimate a person's likelihood of getting one of these diseases, the system uses machine learning algorithms to analyses numerous medical and lifestyle data inputs, such as blood sugar levels, blood pressure, and physical activity.

- **Early Detection:** The main goal of this software is to identify these diseases as they first manifest themselves. Early identification is essential to halting the disease's progression and the body's irreparable harm.
- **Improved Health Outcomes:** By giving users early warning of probable health issues, this programme hopes to enhance the health outcomes of specific individuals.
- **Preventive measures:** The programme attempts to assist users in taking preventive actions to lower their risk of contracting certain diseases. People can lower their chance of getting chronic diseases by receiving personalised suggestions for lifestyle changes, such as dietary adjustments and exercise regimens.
- **Simple to utilize:** A wide spectrum of users, including healthcare experts and those with little to no technological expertise, can easily access and utilize the software.

The Disease prediction system has the potential to have a substantial impact on public health by fulfilling these objectives and goals. For those who are at risk of developing certain diseases, early detection, better health outcomes.

2.3. Objectives of the Proposed System

Our long-term objective is to develop a system that can address every issue that people encounter. Early detection, prevention, better patient outcomes, cost-effective healthcare, and research are the key goals of a disease prediction system.

A disease prediction system's goals can be summed up as follows:

- **Early Detection:** Finding the disease early on is one of a disease prediction system's key goal. The likelihood of the patient recovering and the danger of complications can both be considerably increased by early identification of these disorders.
- **Prevention:** Patients who are at a high risk of contracting a disease can be identified using disease prediction systems. Preventative steps can be made to lower the risk by identifying these patients at an early stage.
- **Improved Patient Outcomes:** Disease prediction systems can aid in improving patient outcomes by identifying diseases early and offering individualised treatment approaches. Early intervention and individualised care improve the chances of recovery and quality of life for patients.
- **Affordable healthcare:** By lowering hospital readmissions, avoiding complications, and improving treatment regimens, disease prediction systems can help lower the overall cost of healthcare.
- **Research:** illness prediction systems can also be used for research. This enables medical experts to examine patient information and spot patterns and risk factors that may aid in the future with illness prevention and treatment.

2.3.1 Goals:

- To create a system that will bring a better platform for patients and health professionals.
- To provide the easiness to health professionals by predictions of diseases.
- To provide the users easily get information about diseases.

2.4. Scope

Depending on the ailment in question, a disease prediction system's scope can change. In this scenario, diabetes, heart disease, and other disease are common chronic diseases for which we will explore the breadth of a disease prediction system.

- **Scope of the diabetes prediction system:** Based on a variety of variables, including age, family history, and lifestyle choices, a diabetes prediction system can be used to identify those who are at risk of acquiring diabetes. A diabetes prediction system’s capability can be expanded to include the early identification of diabetes-related problems such diabetic retinopathy, neuropathy, and nephropathy.

- **Heart Disease Prediction System Scope:** Based on a variety of variables, including age, family history, lifestyle choices, and medical history, a heart disease prediction system can be used to identify people who are at risk of developing heart disease. A method for predicting heart disease can be expanded to include early identification of heart disease complications such coronary artery disease, heart failure, and stroke.
- **Scope of the Symptom Disease Prediction System:** Based on different criteria like age, family history, and lifestyle choices, disease prediction system can be used to identify those who are at risk of developing the condition. A disease prediction system can be expanded to include early identification of the condition's side effects, such as depression, sleep issues, and cognitive decline.

A disease prediction system's capability can extend to risk assessment, disease management, and early detection of consequences. By putting such systems into place, people can actively control their health and stop or postpone the emergence of chronic diseases.

2.5. Modules

2.5.1 Module 1: User Authentication

- Allows the user to register and log in to the app securely
- Keeps user data private and secure

2.5.2 Module 2: User Profile Creation

- Allows the user to create a profile with personal information
- Asks for relevant medical history and current symptoms

2.5.3 Module 3: Data Collection

- Collects user data like glucose levels, blood pressure, heart rate, etc.

2.5.4 Module 5: Diabetes Prediction

- Uses machine learning algorithms to predict the likelihood of diabetes
- Analyzes user data to generate a personalized risk score

2.5.5 Module 6: Heart Disease Prediction

- Uses machine learning algorithms to predict the likelihood of heart disease
- Analyzes user data to generate a personalized risk score

2.5.6 Module 7: Symptom based Disease Prediction

- Uses machine learning algorithms to predict the **different** disease**s**
- Analyzes user data to generate a personalized risk score

3 Requirement Analysis

Any software project, including the mobile app for the Disease prediction system, must go through the requirement analysis phase. The project team identifies and records the system's functional and non-functional requirements at this phase. The design and development of the app will be based on this data. The gathering, analysis, validation, and management of requirements, along with other crucial phases of requirement analysis, will all be covered in this chapter. The precise needs for the Disease prediction system mobile app will also be highlighted, along with how they will be addressed during the design and development phases.

3.1. User classes and characteristics

User Classes and Characteristics for Disease Prediction System Mobile App:

- **Patient:** Patient are those who have been given a diagnosis of diabetes, a heart condition, or Parkinson's disease. Their knowledge of their ailment and how to handle it could vary. Additionally, their technological proficiency and comfort using mobile apps may vary.
- **Caregivers:** Caregivers are people who give assistance and care to people who have diabetes, heart disease, or other disease. They can be relatives or medical specialists. They might have a fundamental understanding of the illness.
- **Health industry personnel:** Healthcare professionals are educated, credentialed persons who provide treatment and assistance to those with diabetes, heart disease, other disease, and other illnesses. They might be highly knowledgeable and skilled in treating the sickness.
- **Researchers:** People who study diabetes, heart disease, or other disease are known as researchers. They might be experts in the study of the disease.
- **General Public:** The general public is any member of society who could be curious to learn more about conditions like diabetes, heart disease, or other disease. They can know very little or nothing about the illness and how to treat it.

They all require a system that can give them precise and understandable information about the illness and how to treat it. They require a user-friendly, educational, and entertaining system. We may create a system that satisfies the requirements and expectations of each user group by defining the various user classes and their important characteristics.

3.2. Requirement Identifying Technique

For Disease prediction system project, which is a mobile app, a combination of techniques can be used to identify the requirements. Some of the techniques that can be used are:

- **Use case technique:** This method can be used to determine the many situations or use cases that the app's users might run into. This method works especially well for applications that include interactive end users. The use cases in this instance could include entering personal health data, seeing the prediction results.
- **Use the event-response:** table technique for real-time systems where the majority of operations are handled at the backend. The method can be applied to this project to determine the various situations that could cause the prediction system to provide a result and the accompanying answers that the system should provide.
- **Storyboarding method:** For applications that require a lot of graphics, this strategy is very helpful. The method can be applied to this project to determine the various screens and user interfaces the app should feature. This comprises elements like the dashboard, the login page, and the screen that shows the prediction results.

The project team may determine every requirement for the Disease prediction system mobile app by combining these strategies.

The system's requirements must be determined before a mobile app for disease prediction can be developed. we can use the following methods to find requirements:

- **Brainstorming:** To determine the needs, brainstorming sessions can be performed with subject-matter experts, medical professionals, and potential app users.
- **Surveys:** To learn more about the requirements and preferences of potential users, surveys might be conducted.
- **Use case analysis:** This technique can be used to pinpoint the various interactions and scenarios that the app must handle.
- **Scenario analysis:** To pinpoint the many app use cases and requirements, scenarios can be built.
- **Requirements:** workshops can be held to gather stakeholders and subject-matter experts and establish the requirements.

Based on these techniques, the following requirements can be identified for Disease prediction system mobile app:

- User login and registration capabilities
- The ability to manage user profiles.
- Functionality for diabetes, heart disease, and Symptom based disease data analysis and prediction.
- Materials that provide knowledge and instruction about the disorders.
- Security and privacy safeguards to safeguard user data.
- An intuitive and user-friendly user interface.
- Support for many mobile operating systems and devices.

These specifications will aid in the creation of an extensive and straightforward mobile app for disease prediction systems.

3.3. Functional Requirements

The main product features or functions that must implement to enable users to accomplish the tasks. So, it is important to make clear both for the development team and stockholders.

3.3.1 User Login and Registration:

Table 3 Login and Registration

Name	FR-1: User Login and Registration
Summary	The app must have a user registration feature to create user accounts and manage user data.
Rationale	To enable personalized disease risk assessment and recommendations
Requirement	User account creation User profile management User data management
Source	User Stories
Dependencies	Database, User Interface

Priority	High

3.3.2 Health Data Collection:

Table 4 Health Data Collection

Name	FR-2: Health Data Collection
Summary	The app must be able to collect and store user health data such as blood sugar levels, heart rate, and tremors.
Rationale	To allow users to monitor their symptoms and receive early warning of potential health issues
Source	User Stories
Dependencies	User Registration, Database, User Interface
Priority	High

3.3.3 Data Analysis:

Table 5 Data Analysis

Name	FR-3: Data Analysis
Summary	The app must analyze the user's health data to predict the likelihood of developing a particular disease.
Rationale	To provide users with accurate and up-to-date information about disease prevention and management
Source	User Stories
Dependencies	User Registration, Database, User Interface

Priority	High

3.3.4 Disease Prediction:

Table 6 Disease Prediction

Name	FR-4: Disease Prediction
Summary	The app must be able to predict the likelihood of developing. Disease based on the user's health data.
Rationale	To provide users with an understanding of their risk of developing certain diseases
Source	User Stories
Dependencies	Symptom Input, Machine Learning Algorithm
Priority	High

3.4. Non-Functional Requirements

Requirements for the Disease Prediction System Mobile App That Are Not Functional:

3.4.1 Performance:

- The application must react to user input in under two seconds.
- The application shouldn't freeze or crash at the time of using.
- Multiple concurrent users should not cause the programme to significantly slow down.

3.4.2 Usability:

- The app should be simple to use and navigate for users of all ages.
- The user interface of the app should be straightforward and easy to use.

3.4.3 Security:

- The app must securely store all user data and guard against unauthorized access.
- The app should transfer data to the server using secure communication protocols.
- To avoid unauthorized access, the app should have strict authentication and authorization requirements.

3.4.4 Compatibility:

- The app should be compatible with both the iOS and Android operating systems.
- The app ought to work with a variety of mobile devices.
- The app ought to be flexible enough to work with various screen resolutions and sizes.

3.4.5 Reliability:

- The app must run continuously, without interruption for maintenance.
- Any faults or crashes should be swiftly fixed by the application.
- To avoid data loss.

3.4.6 Scalability:

- The programme should be able to accommodate an expanding user and data load without experiencing noticeable slowdowns.
- The application needs to be flexible enough to adjust to shifting hardware and software environments.

3.4.7 Performance Efficiency:

- The programme should use the least amount of CPU, memory, and battery power on the device.
- To reduce user, wait times, the app should load and display data rapidly.
- To speed up data transfers, the app should employ caching and compression techniques.

3.4.8 Maintainability:

- The app needs to be simple to update and maintain.
- The app's architecture ought to be modular to make adding and removing features simple.
- The app's codebase should be well-documented and simple to debug.

3.4.9 Data privacy:

- The app should adhere to all applicable data privacy laws and rules, including GDPR and HIPAA.
- The app should make users aware of how data is collected and used.
- The app should enable users to manage their data and, if wanted, erase it.

3.5. External Interface Requirements

The goal of the illness prediction system project is to create a mobile application that can precisely forecast a person's risk of contracting one of these diseases based on their lifestyle and medical history.

3.5.1 User Interfaces Requirements

The application's user interface should be simple and easy to use, with instructions that are straightforward and to the point to help users submit their medical and lifestyle data. With scalable font sizes and an intuitive style, the programme should be made to accommodate users of all ages. Additionally, the user interface needs to be responsively designed so that it may change to fit the screen size and orientation of any mobile device.

3.5.2 Input Requirements

A range of data input methods should be supported by the application, including manual data entry. If there are mistakes or inconsistencies in the data, the application should be able to detect a wide variety of data types and should let users know.

3.5.3 Output Requirements

Users should receive clear and straightforward output from the program, including a forecast of how likely it is that they will contract one of the targeted diseases as well as suggestions for dietary modifications and medical procedures that might assist to avoid or treat the disease. When necessary, visual aids like graphs or charts should be presented together with the output in an easy-to-understand style.

3.5.4 External Interfaces

The software should be developed such that it can efficiently interface with other hardware and software components. The software should be able to transport data securely and effectively and should be able to recognise a broad variety of data formats. Additionally, the programme should be built to reference content from other publications.

3.5.5 Software interfaces

The programme should be able to recognise a variety of data types, be built to connect with other software systems, and be capable of sending data safely and effectively. The following software components will interact with the app:

- **Database:** To store and retrieve user information, the app will communicate with a database.
- **Social media accounts:** In order to allow users to log in and register using their social media accounts, the app must connect with social media accounts.

3.5.6 Hardware interfaces

The programme must be able to recognise the data and send data effectively and securely. The mobile app for the prediction system shall not have any hardware requirements. It must be made to work with iOS and Android smartphones' hardware requirements.

3.5.7 Communications interfaces

The programme needs to be built to interact with other systems via a number of communication interface. Each interface's proper communication protocol should be recognised by the program, which should also be able to convey data effectively and securely. The following external components must be in communication with the app:

Internet: In order for the app to communicate with the database and social media accounts, it must have access to the internet.

Email: The app will notify users via email, in case forget password etc.

A well-designed user interface that can receive information from a number of sources and give consumers clear, concise output is required for the illness prediction system project. Additionally, the application needs to be built to work well with other software components. The programme can deliver precise and trustworthy disease forecasts that can help people take charge of their health and well-being by implementing these external interface criteria.

4 Design and Architecture

The introduction should give a general overview of the software design and architecture of the mobile app project for the diabetes, heart, and Symptom baased disease prediction systems. The function and objectives of the software design should also be discussed in this part, along with any pertinent context or background data.

4.1. Design Overview

A high-level explanation of the system architecture and design should be provided in the design overview. The primary system elements, their interactions with one another, and the structure of the system should all be covered in this section.

4.2. Architectural Design

The system's overall structure should be described in the architecture section, together with its many hardware and software components, interfaces between them, and connections between them. The design choices chosen for the system architecture should also be discussed in this part, along with any trade-offs that were taken into account. It ought to additionally have illustrations of the system architecture in the form of diagrams.

4.3. Design Models

Use Case Model: The use case model explains how users and the system interact and how the system reacts to user activities. Some possible use cases for a disease prediction system would be:

- User sign in and enters some basic data. (age, gender, medical history, etc.)
- When the user enters particular symptoms and chooses a disease to predict, a prediction is returned.
- The user can view all three predictions.

Activity Diagram: The activity diagram shows how the system's activities flow. The activity diagram in a disease prediction system could comprise the following:

- Opens the app and logs in the user
- The user chooses a disease to forecast
- User provides symptoms and medical background

- Based on the inputs, the system provides a prediction.
- The programmer stores the forecast and give result.

Sequence Diagram: The sequence diagram depicts how various system components interact with one another over time. The sequence diagram for a disease prediction system might show:

- The user chooses a disease to forecast
- User provides symptoms and medical background
- The system communicates the input information to the prediction algorithm.
- The system receives a prediction from the algorithm and acts upon it.
- The programmer stores the forecast.
- The system shows the user the prediction.

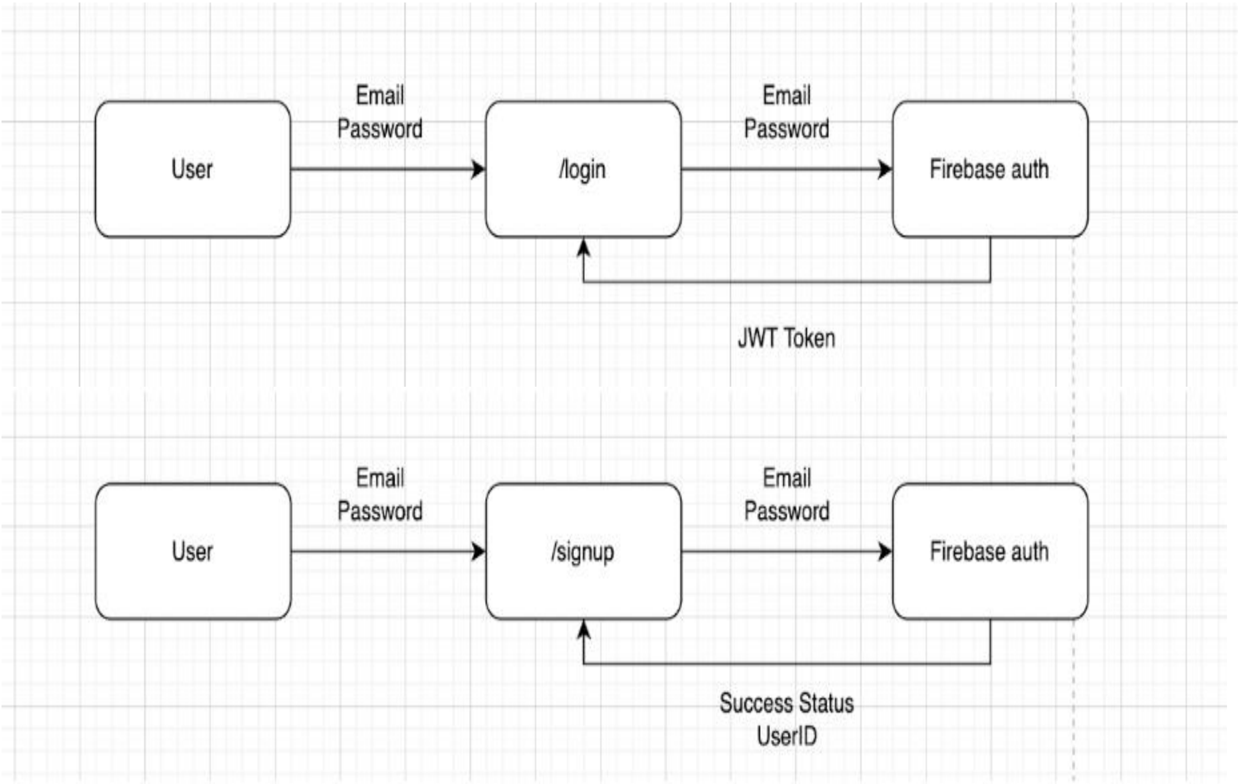
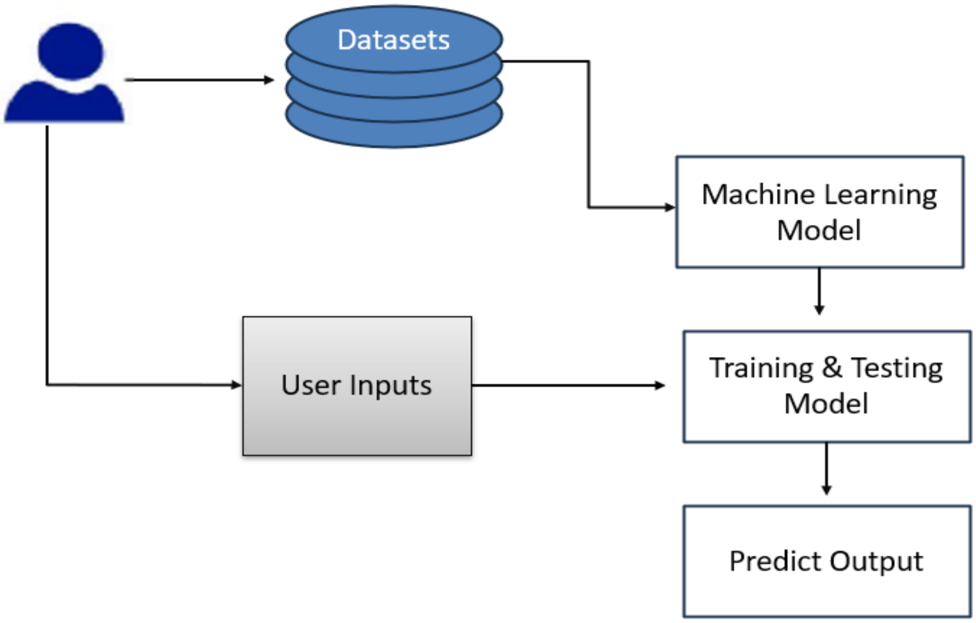


Figure 1 Block Diagram 01



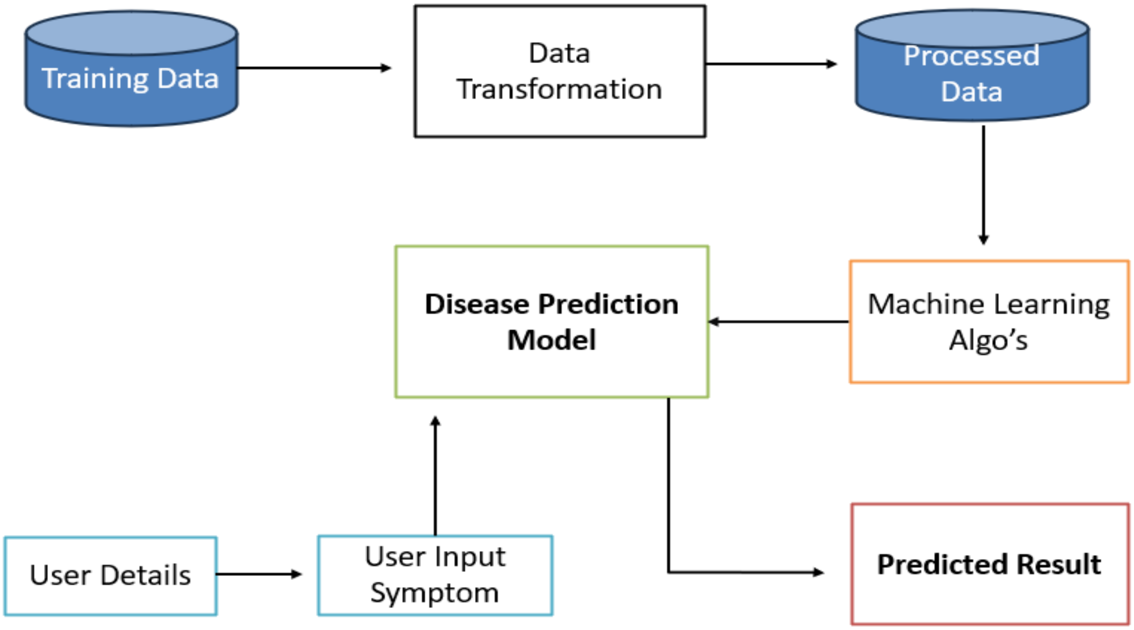


Figure 2 Block Diagram 02

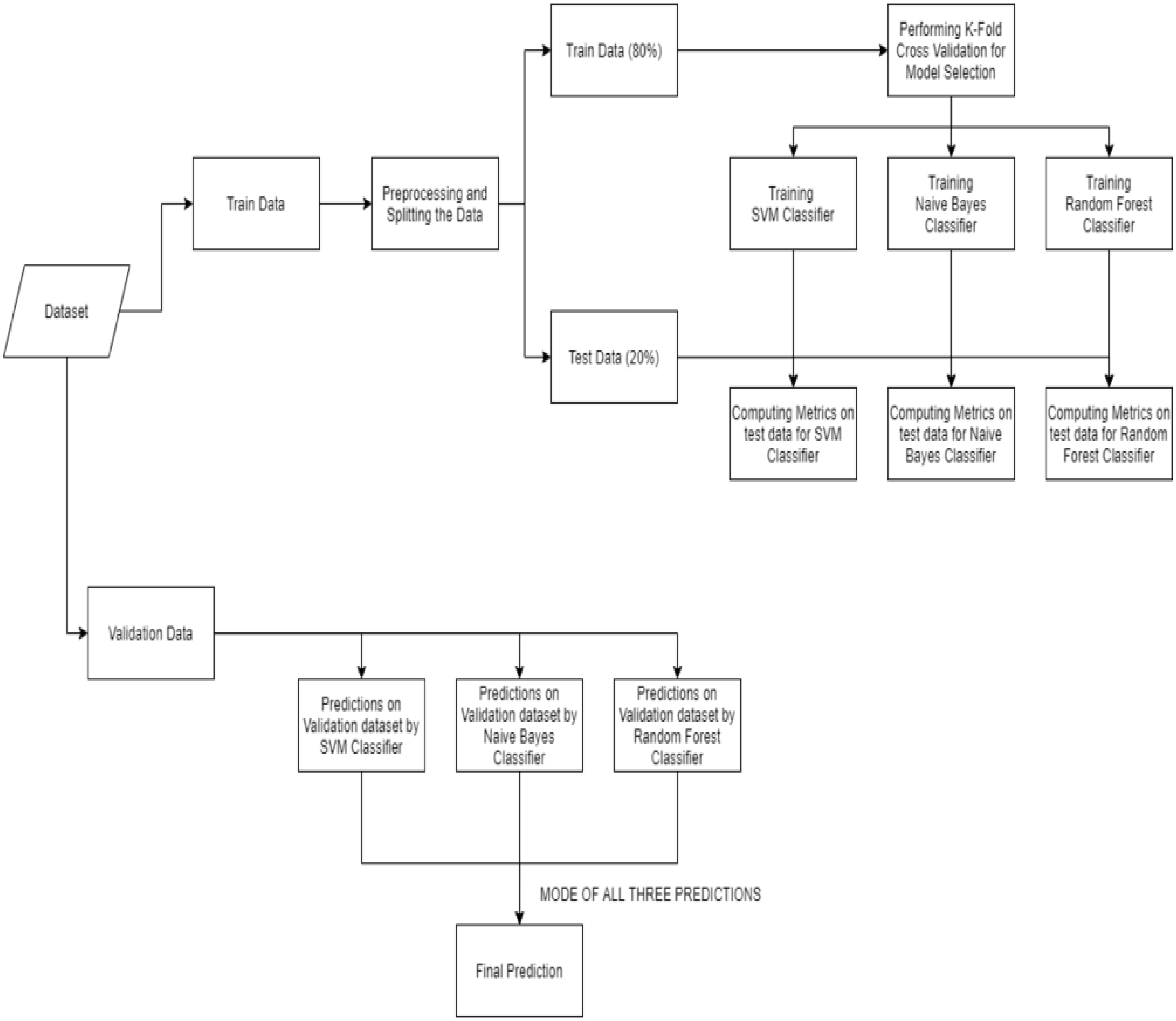

Dataset
Train Data
Preprocessing and Splitting the Data
Train Data (80%)
Performing K-Fold Cross Validation for Model Selection
Training SVM Classifier
Training Naive Bayes Classifier
Training Random Forest Classifier
Test Data (20%)
Computing Metrics on test data for SVM Classifier
Computing Metrics on test data for Naive Bayes Classifier
Computing Metrics on test data for Random Forest Classifier
Validation Data
Predictions on Validation dataset by SVM Classifier
Predictions on Validation dataset by Naive Bayes Classifier
Predictions on Validation dataset by Random Forest Classifier
MODE OF ALL THREE PREDICTIONS
Final Prediction

Figure 3 Database Diagram

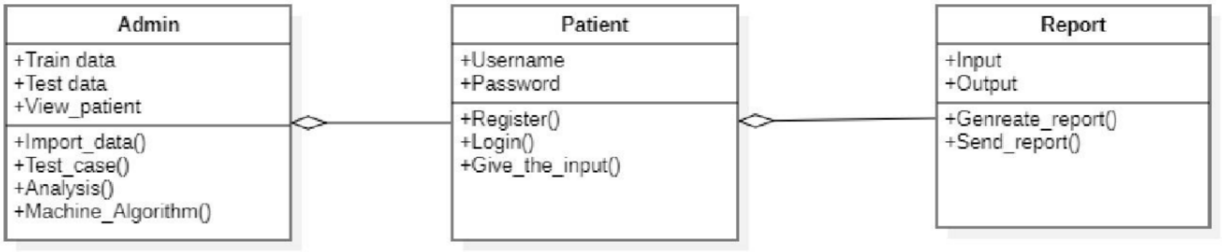


Figure 4 Class Diagram

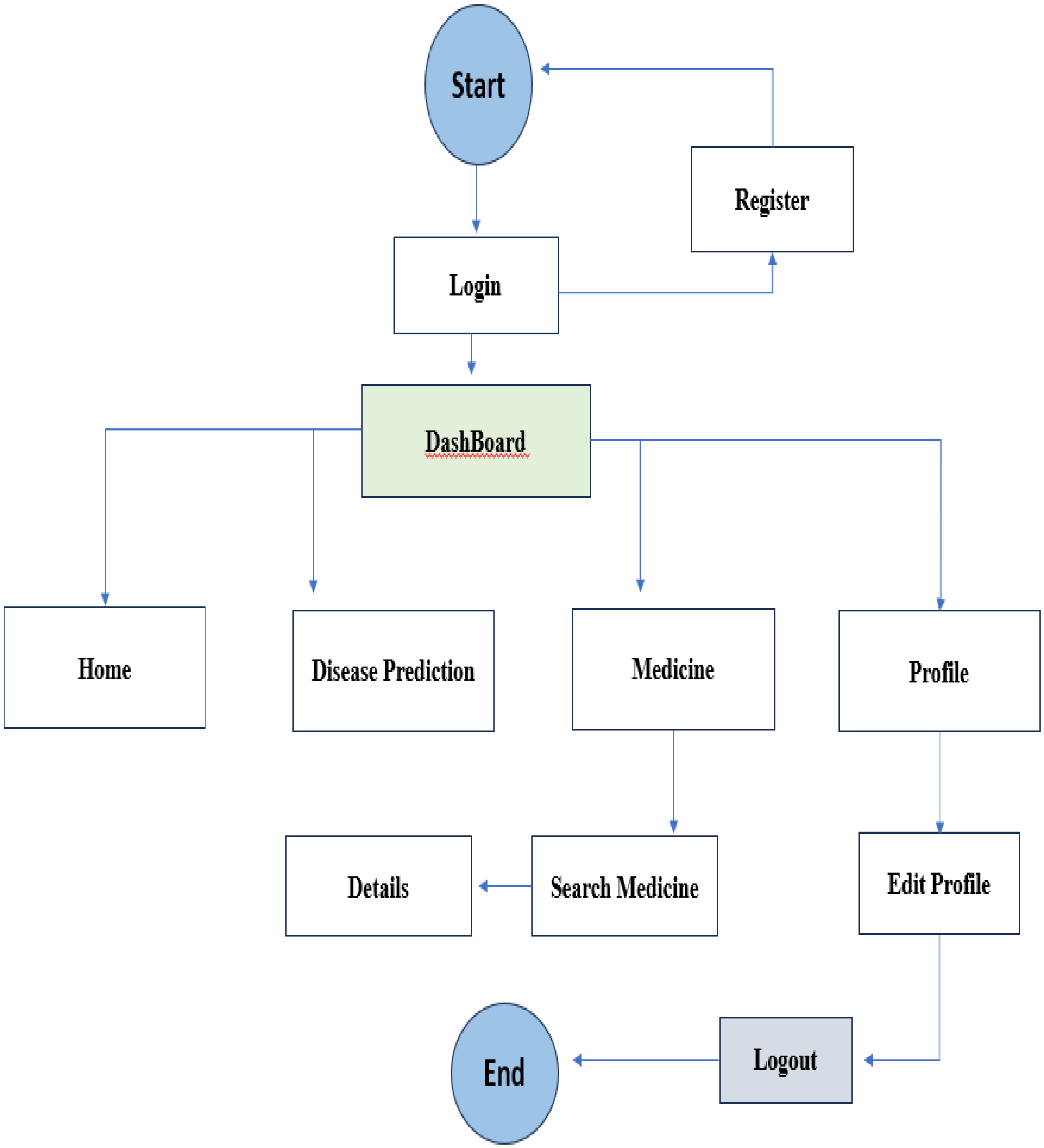
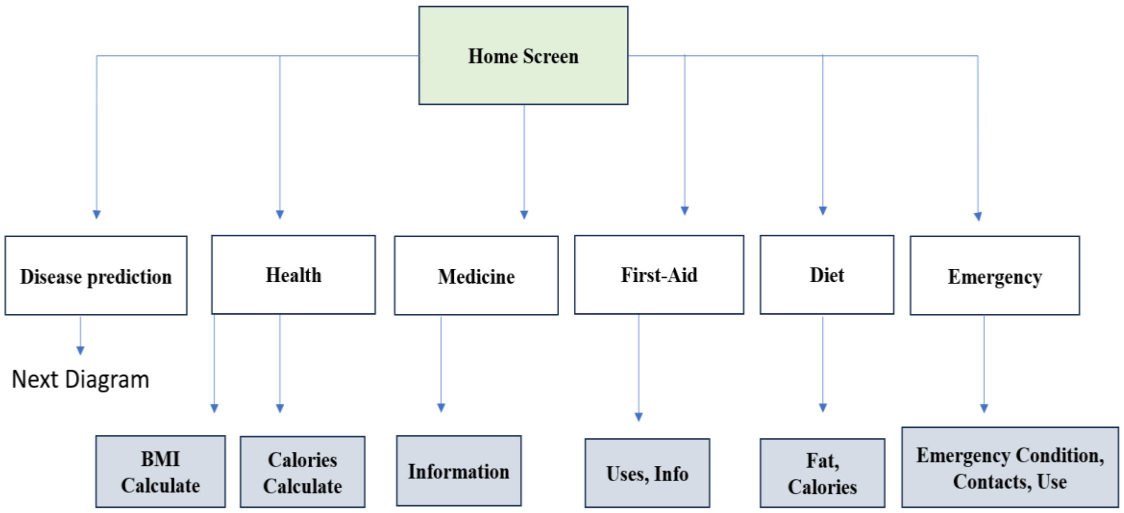
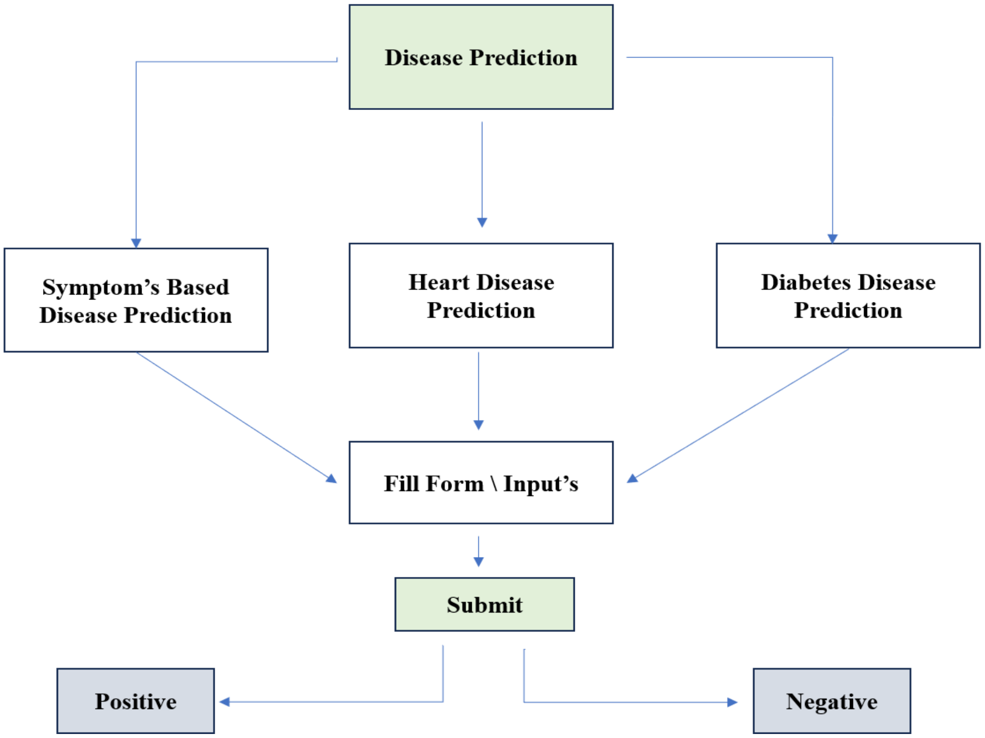


Figure 5 Activity Diagram


Home Screen
Disease prediction
Health
Medicine
First-Aid
Diet
Emergency
Next Diagram
BMI Calculate
Calories Calculate
Information
Uses, Info
Fat, Calories
Emergency Condition, Contacts, Use


Disease Prediction
Symptom’s Based Disease Prediction
Heart Disease Prediction
Diabetes Disease Prediction
Fill Form \ Input’s
Submit
Positive
Negative

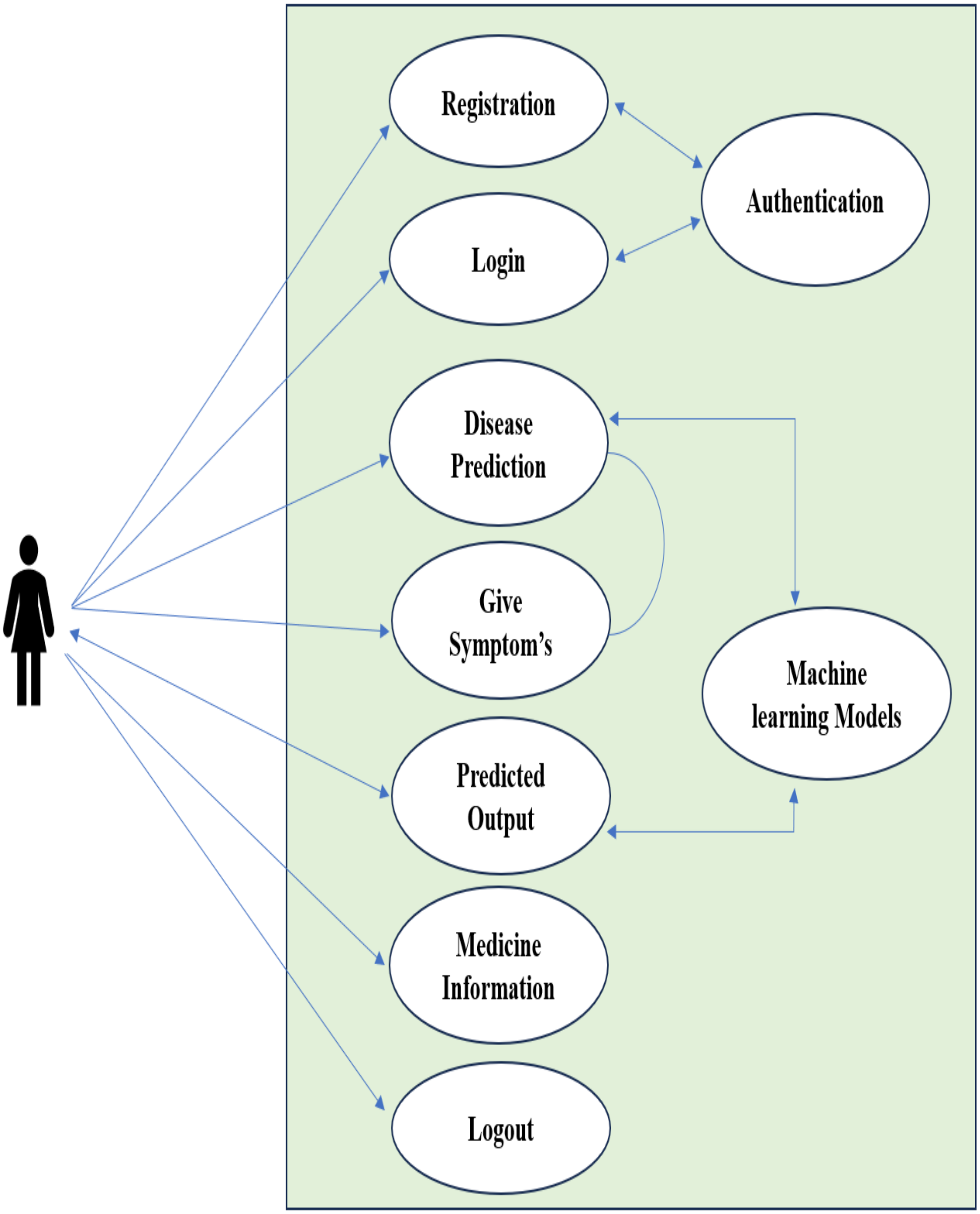


Figure 6 Use case Diagram

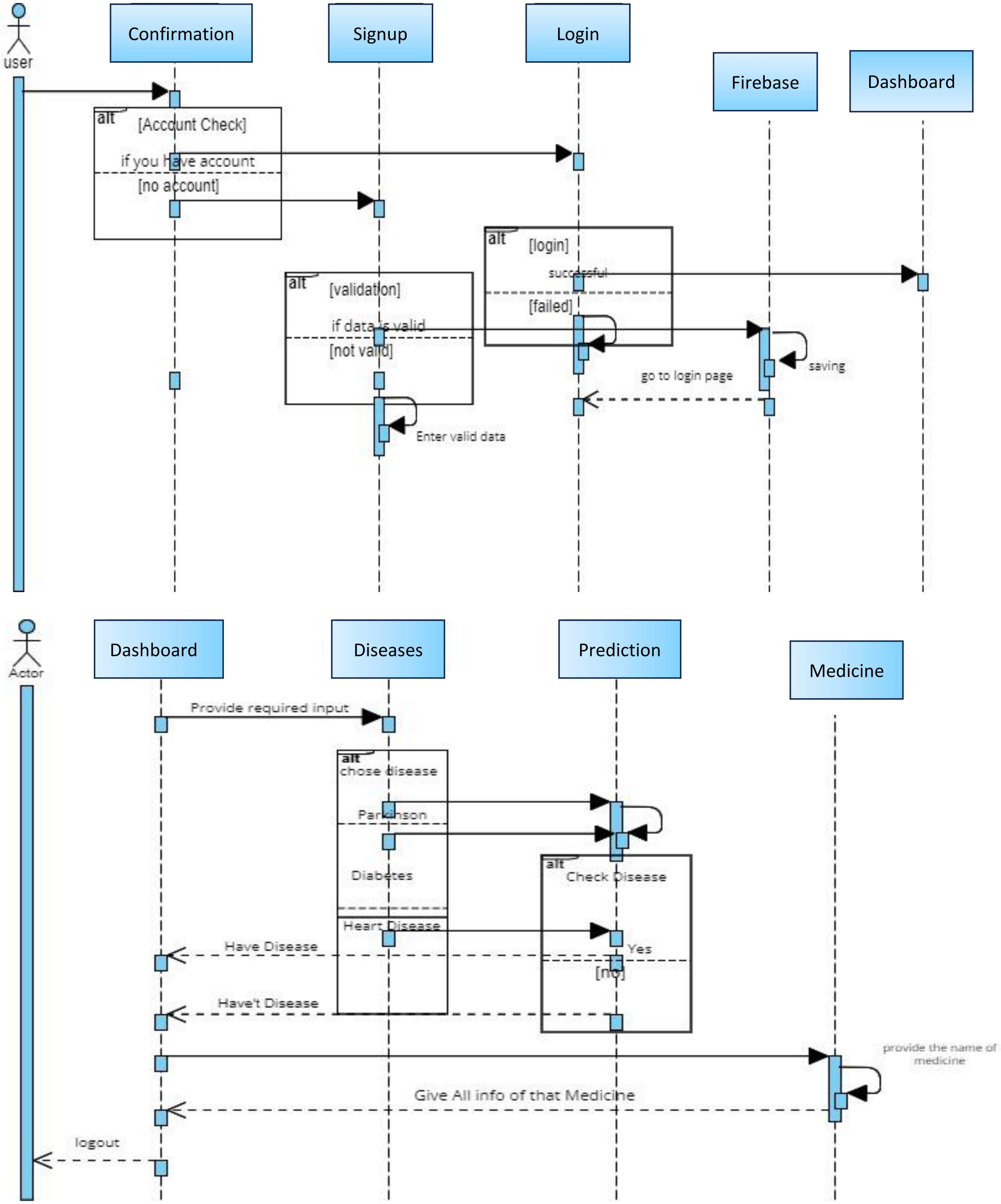


Figure 7 Sequence diagram

4.4. Data Design

The data structures and entities utilised in the system, as well as how the data is stored, retrieved, and modified, should be described in the data model section. The connections and hierarchy of the various data entities should also be discussed in this section. The database schema and any necessary data validation rules should also be included.

4.5. Human Interface Design

The section on user interface design should go into detail on the layout, navigation, and visual appeal of the mobile app's user interface. Any usability considerations and user experience design guidelines should be discussed in this section as well. Mockups or wireframes of the user interface should be included to show the design.

4.6. Software Components

Software Components This section should outline the various software elements of the system, such as the frameworks, libraries, and programming languages that were utilised to create it. The interactions and integration of the various software components into the system should also be covered in this section.

5 Implementation

This chapter describe the implementation of the whole application. The implementation is the most important phase of software development life cycle in which we implement the whole idea and requirements. If we are able to implement the functionalities successfully so we will definitely be able to satisfy the users. Each and every functionality as well as implementation described in this chapter.

5.1. Tools & Technologies

Tool and Technologies used in our system are given below:

5.1.1 React Native (version 0.66.0)

React Native is a powerful open-source framework for building cross-platform mobile applications using JavaScript and React. Developed by Facebook, React Native enables developers to create high-performance mobile apps with a single codebase that can run on both iOS and Android platforms. With a focus on efficiency and a declarative component-based approach, React Native streamlines the development process, allowing for rapid iteration and easy maintenance. The framework leverages native components, providing a native look and feel while still allowing for the flexibility of web development. React Native continues to evolve with regular updates and a vibrant community, making it a popular choice for mobile app development.

5.1.1.1 Features

1. **Cross-Platform Development:** Build applications that run seamlessly on both iOS and Android platforms, saving development time and resources.
2. **Reusable Components:** React Native's component-based architecture enables the creation of reusable UI components, enhancing code maintainability and reducing redundancy.
3. **Hot Reloading**: Experience faster development cycles with hot reloading, allowing developers to instantly see the result of the latest changes without rebuilding the entire app.
4. **Native Performance:** Leverage the performance of native controls and modules, ensuring that React Native applications deliver a smooth and responsive user experience**.**
5. **Third-Party Plugin Integration:** Easily integrate third-party plugins and libraries, expanding the functionality of your app and accelerating development**.**

6. **React Ecosystem Integration:** Seamlessly integrate with the broader React ecosystem, making it easy to incorporate popular libraries, tools, and patterns into your mobile app development.
7. **Community Support:** Benefit from a large and active community of developers, ensuring access to a wealth of resources, tutorials, and support for troubleshooting.
8. **Live Updates:** Deploy over-the-air updates to your React Native apps without going through the app store review process, enabling quick bug fixes and feature enhancements.
9. **Offline Support:** Build offline-capable applications using libraries like Redux, AsyncStorage, and other state management solutions to ensure a reliable user experience even without an internet connection.
10. **Built-in Developer Tools:** Utilize built-in debugging and profiling tools to troubleshoot and optimize your React Native applications during development.

5.1.1.2 Firebase Connectivity

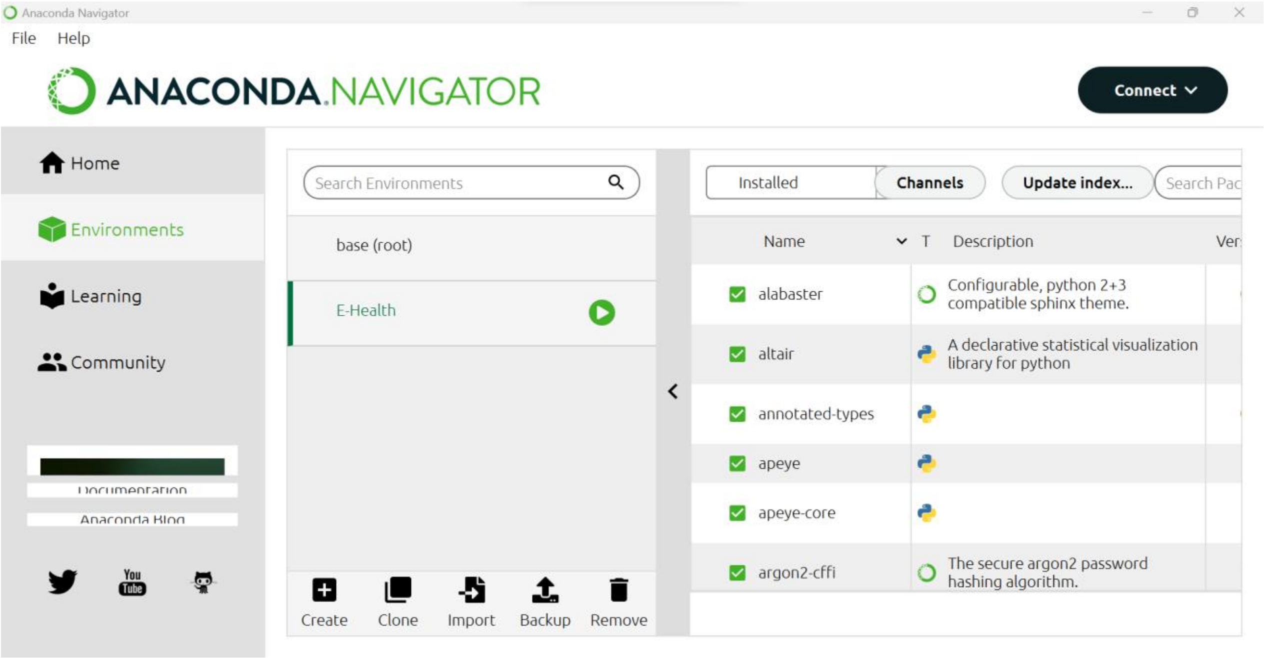
The firebase is cloud-hosted database. Data is stored as JSON and synchronized in real time to every connected client.

5.1.1.3 Node.js

Node.js is a powerful, open-source, server-side JavaScript runtime environment that enables developers to build scalable and high-performance applications. It utilizes the V8 JavaScript engine from Google Chrome, providing a fast and efficient execution environment for server-side and network applications. Node.js is event-driven and non-blocking, making it particularly well-suited for building real-time applications and APIs.

5.1.2 Anaconda

Anaconda is a distribution of open-source programming and data science tools that simplifies the process of deploying and managing Python and R environments. It is designed to streamline the development and deployment of data science, machine learning, and artificial intelligence applications. Anaconda includes a comprehensive collection of pre-built packages, libraries, and tools, making it a go-to choose for individuals and organizations in the data science community.

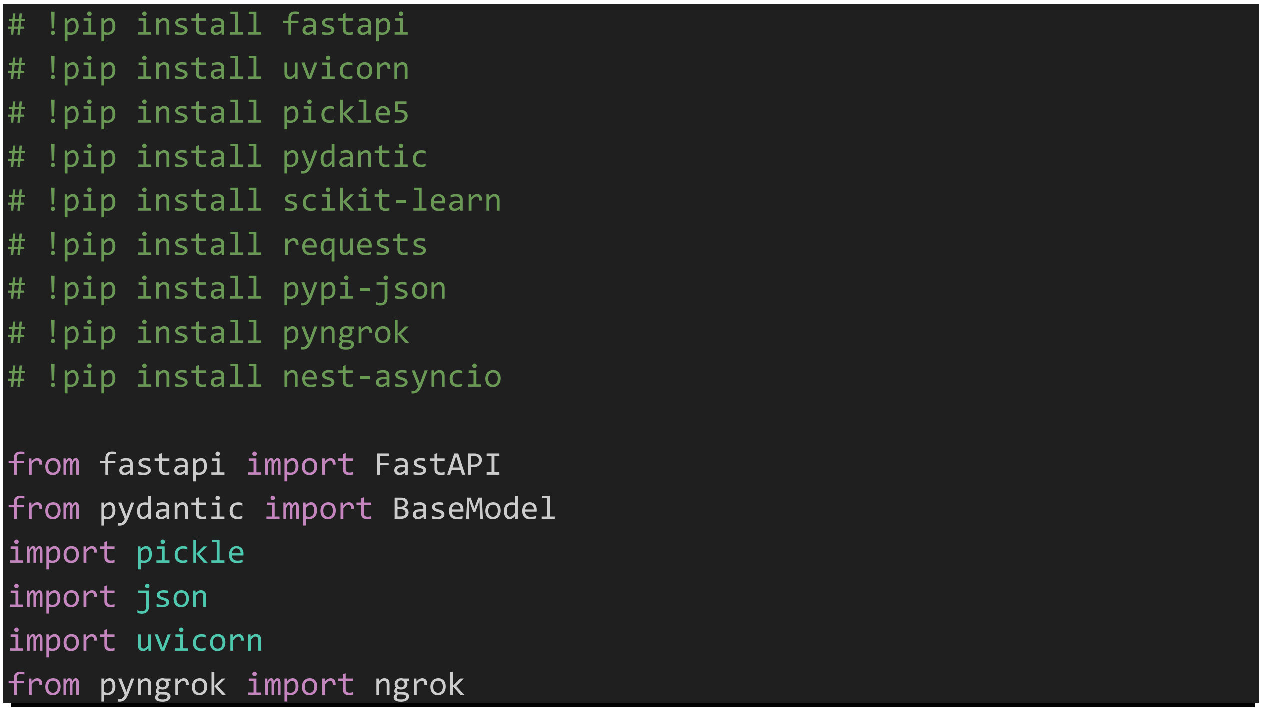


	Name	T	Description	Ver
☑	alabaster	○	Configurable, python 2+3 compatible sphinx theme.	
☑	altair		A declarative statistical visualization library for python	
☑	annotated-types			
☑	apeye			
☑	apeye-core			
☑	argon2-cffi	○	The secure argon2 password hashing algorithm.	

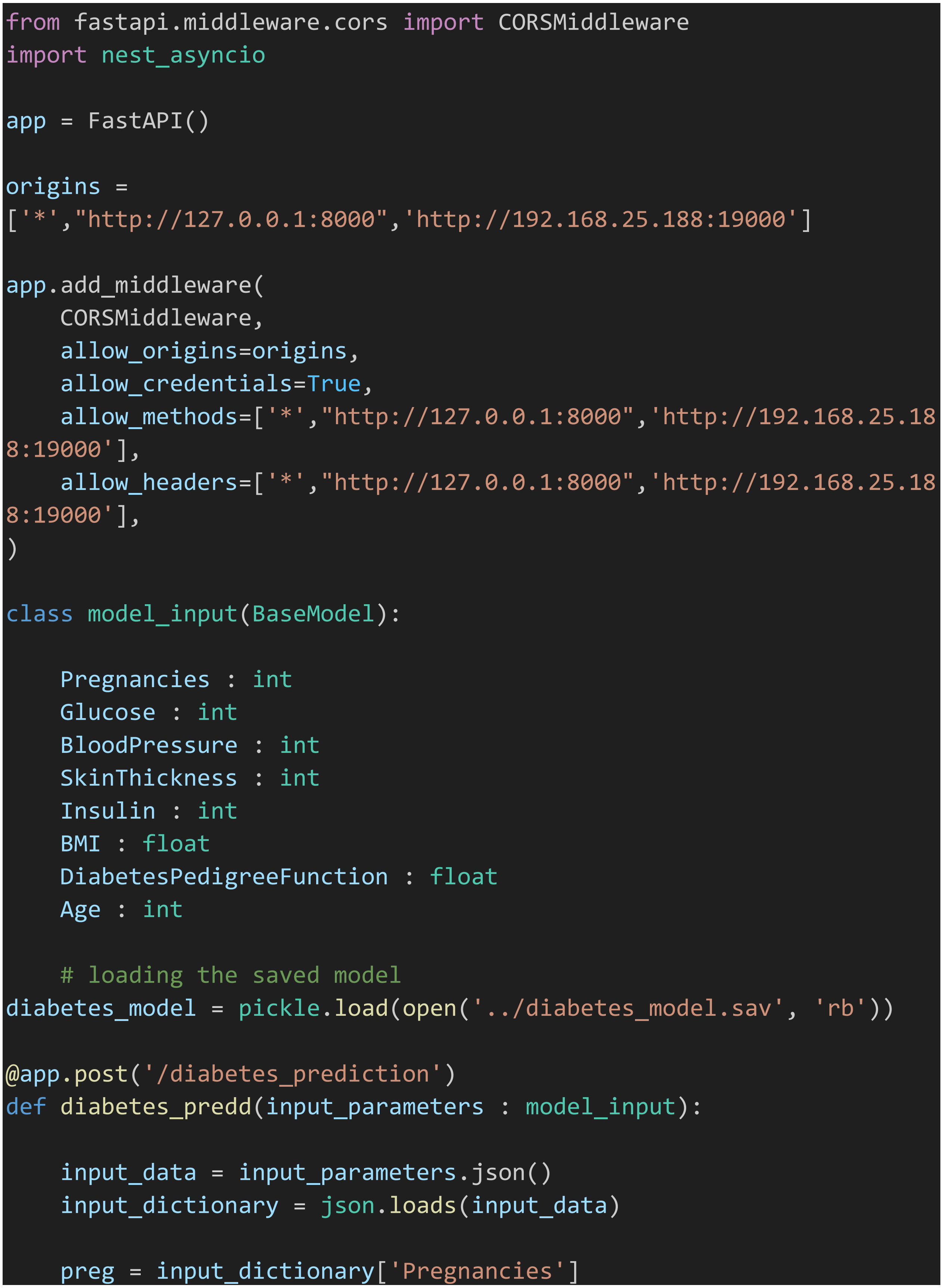
5.2. Algorithm

Machine learning models are increasingly employed for disease prediction, leveraging patterns and insights from data to aid in early diagnosis and intervention. Various algorithms are applied based on the nature of the data and the specific requirements of the healthcare application.

5.2.1 Diabetes Prediction Model:



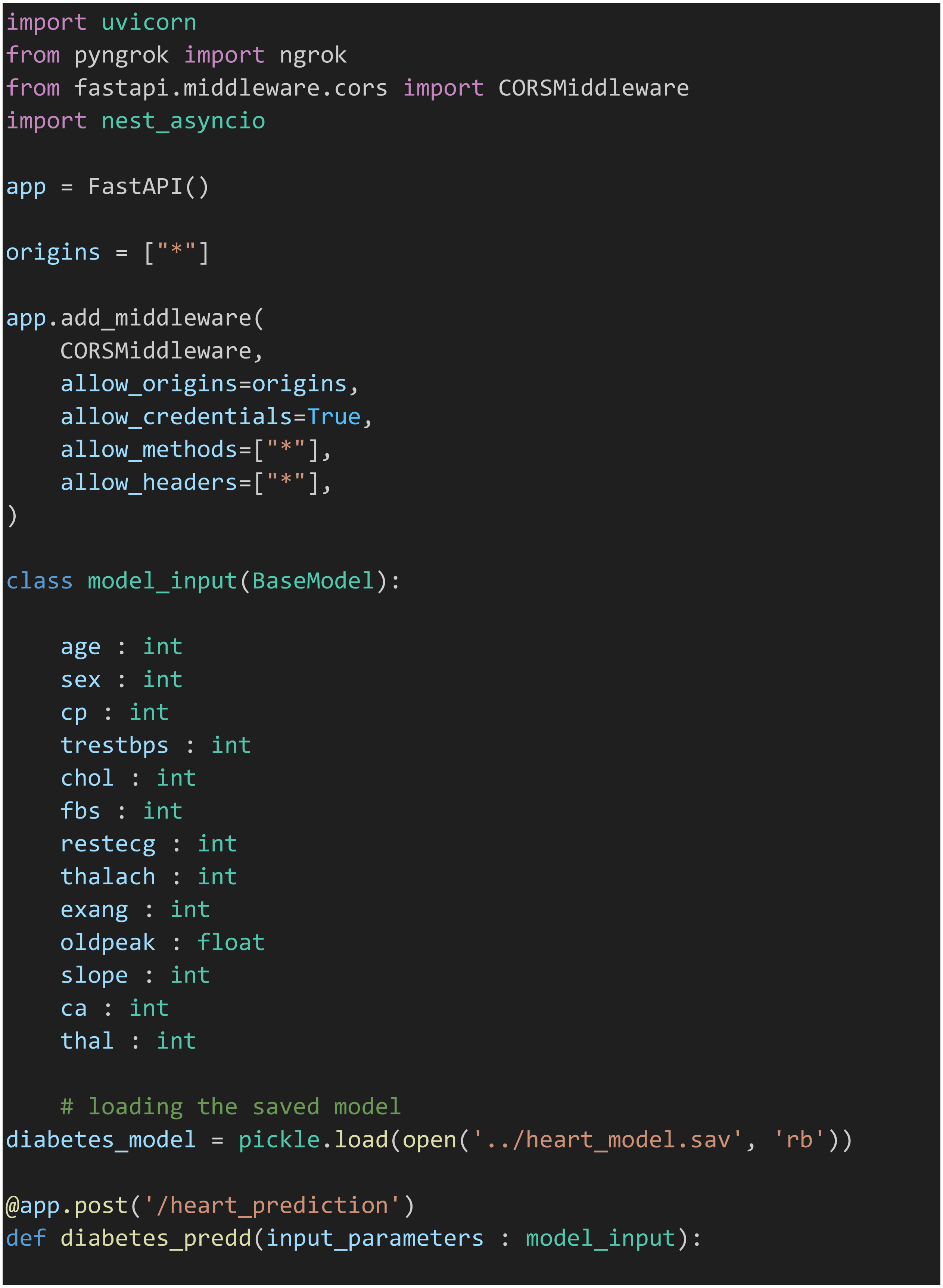
```
# !pip install fastapi
# !pip install uvicorn
# !pip install pickle5
# !pip install pydantic
# !pip install scikit-learn
# !pip install requests
# !pip install pypi-json
# !pip install pyngrok
# !pip install nest-asyncio

from fastapi import FastAPI
from pydantic import BaseModel
import pickle
import json
import uvicorn
from pyngrok import ngrok
```



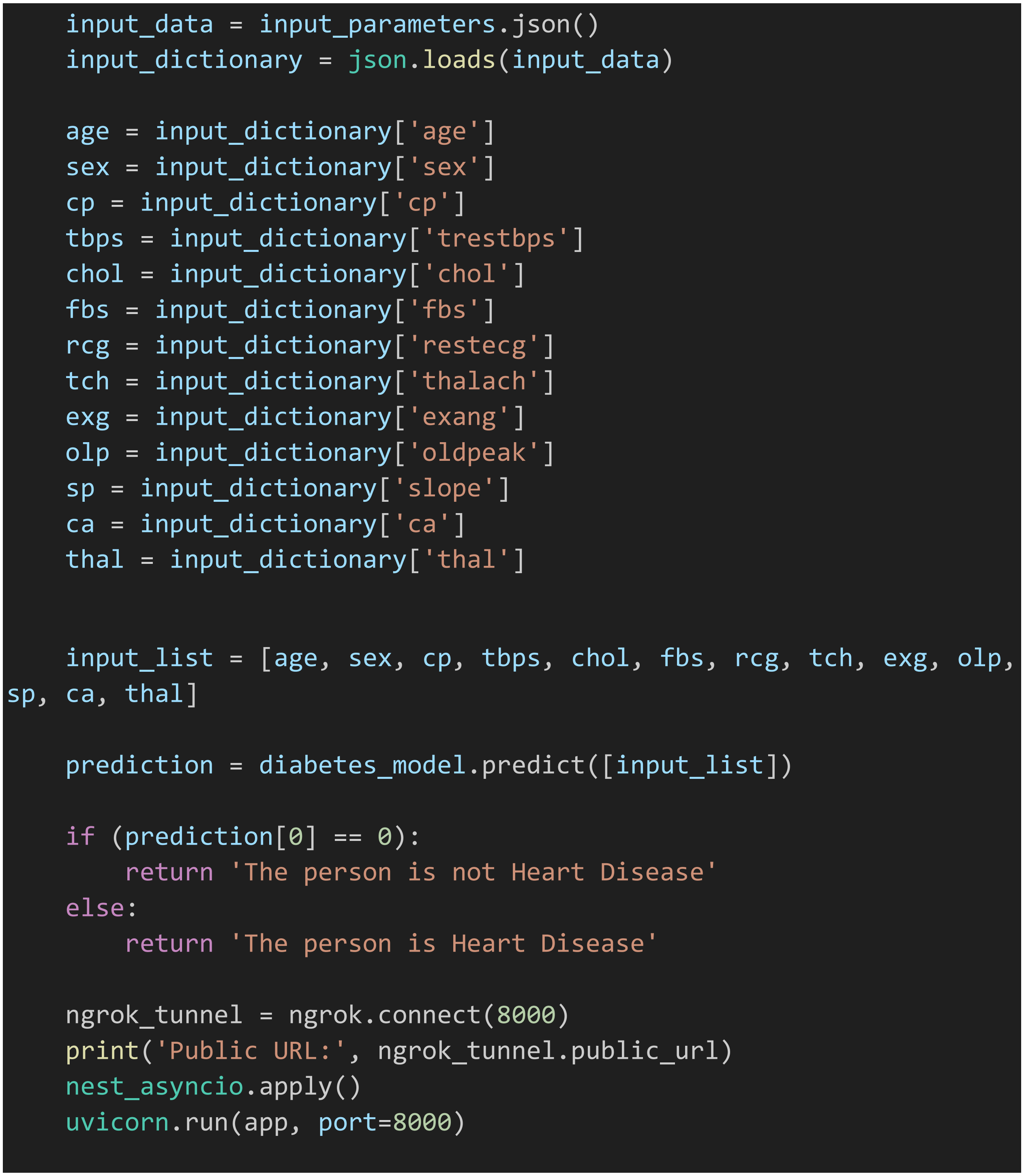
```
from fastapi.middleware.cors import CORSMiddleware
import nest_asyncio

app = FastAPI()

origins =
['*',"http://127.0.0.1:8000",'http://192.168.25.188:19000']

app.add_middleware(
    CORSMiddleware,
    allow_origins=origins,
    allow_credentials=True,
    allow_methods=['*',"http://127.0.0.1:8000",'http://192.168.25.188:19000'],
    allow_headers=['*',"http://127.0.0.1:8000",'http://192.168.25.188:19000'],
)

class model_input(BaseModel):

    Pregnancies : int
    Glucose : int
    BloodPressure : int
    SkinThickness : int
    Insulin : int
    BMI : float
    DiabetesPedigreeFunction : float
    Age : int

    # loading the saved model
diabetes_model = pickle.load(open('../diabetes_model.sav', 'rb'))

@app.post('/diabetes_prediction')
def diabetes_predd(input_parameters : model_input):

    input_data = input_parameters.json()
    input_dictionary = json.loads(input_data)

    preg = input_dictionary['Pregnancies']
    glu = input_dictionary['Glucose']
    bp = input_dictionary['BloodPressure']
    skin = input_dictionary['SkinThickness']
    insulin = input_dictionary['Insulin']
    bmi = input_dictionary['BMI']
    dpf = input_dictionary['DiabetesPedigreeFunction']
    age = input_dictionary['Age']


    input_list = [preg, glu, bp, skin, insulin, bmi, dpf, age]

    prediction = diabetes_model.predict([input_list])

    if (prediction[0] == 0):
        return 'The person is not diabetic'
    else:
        return 'The person is diabetic'

    ngrok_tunnel = ngrok.connect(8000)
    print('Public URL:', ngrok_tunnel.public_url)
    nest_asyncio.apply()
    uvicorn.run(app, port=8000)
```

5.2.2 Heart Disease prediction:

```
# !pip install fastapi
# !pip install uvicorn
# !pip install pickle5
# !pip install pydantic
# !pip install scikit-learn
# !pip install requests
# !pip install pypi-json
# !pip install pyngrok
# !pip install nest-asyncio

from fastapi import FastAPI
from pydantic import BaseModel
import pickle
import json
```

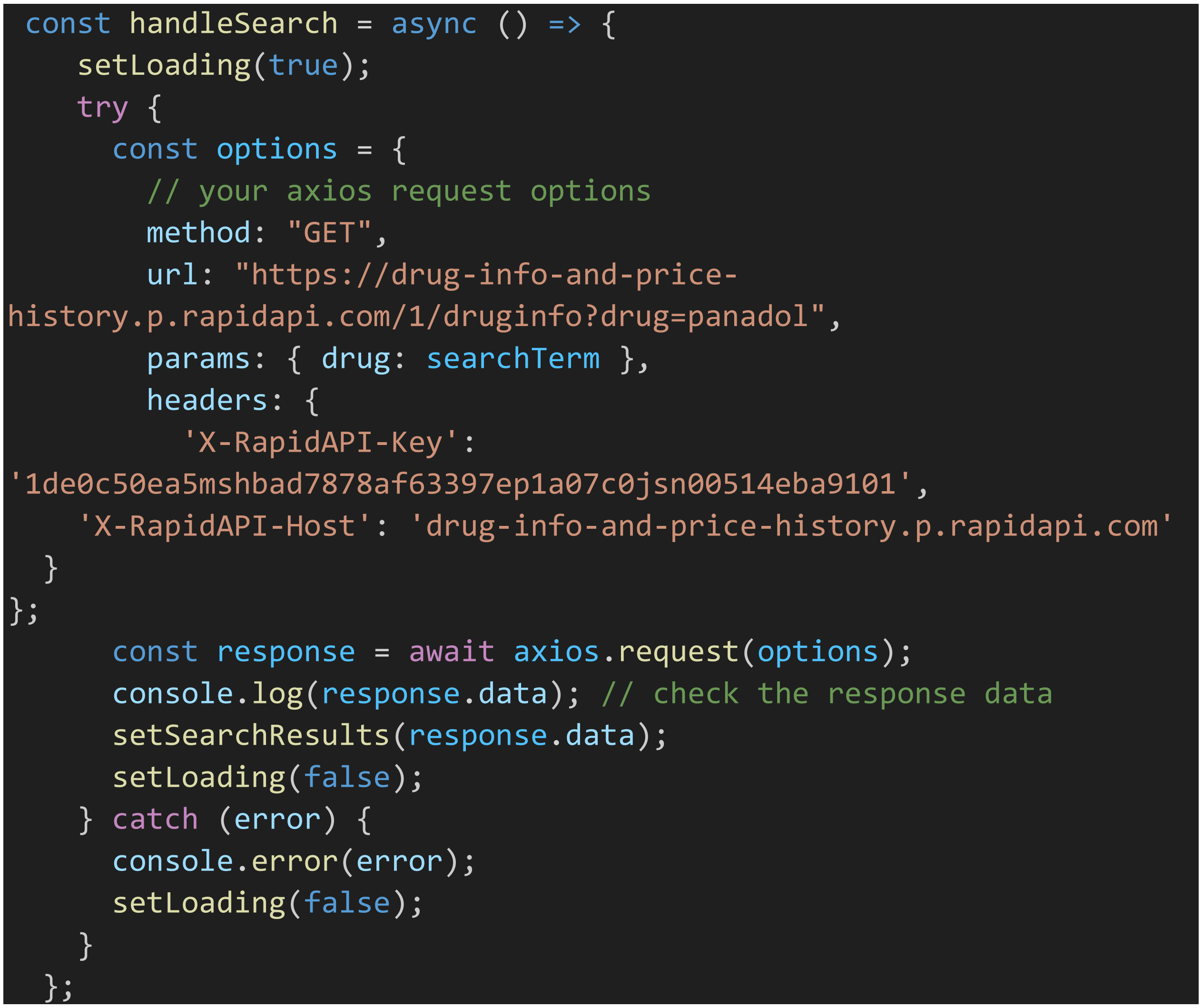


```python
import uvicorn
from pyngrok import ngrok
from fastapi.middleware.cors import CORSMiddleware
import nest_asyncio

app = FastAPI()

origins = ["*"]

app.add_middleware(
    CORSMiddleware,
    allow_origins=origins,
    allow_credentials=True,
    allow_methods=["*"],
    allow_headers=["*"],
)

class model_input(BaseModel):

    age : int
    sex : int
    cp : int
    trestbps : int
    chol : int
    fbs : int
    restecg : int
    thalach : int
    exang : int
    oldpeak : float
    slope : int
    ca : int
    thal : int

    # loading the saved model
diabetes_model = pickle.load(open('../heart_model.sav', 'rb'))

@app.post('/heart_prediction')
def diabetes_predd(input_parameters : model_input):
```



```
    input_data = input_parameters.json()
    input_dictionary = json.loads(input_data)

    age = input_dictionary['age']
    sex = input_dictionary['sex']
    cp = input_dictionary['cp']
    tbps = input_dictionary['trestbps']
    chol = input_dictionary['chol']
    fbs = input_dictionary['fbs']
    rcg = input_dictionary['restecg']
    tch = input_dictionary['thalach']
    exg = input_dictionary['exang']
    olp = input_dictionary['oldpeak']
    sp = input_dictionary['slope']
    ca = input_dictionary['ca']
    thal = input_dictionary['thal']


    input_list = [age, sex, cp, tbps, chol, fbs, rcg, tch, exg, olp,
sp, ca, thal]

    prediction = diabetes_model.predict([input_list])

    if (prediction[0] == 0):
        return 'The person is not Heart Disease'
    else:
        return 'The person is Heart Disease'

    ngrok_tunnel = ngrok.connect(8000)
    print('Public URL:', ngrok_tunnel.public_url)
    nest_asyncio.apply()
    uvicorn.run(app, port=8000)
```

5.3. External APIs/SDKs



```
 const handleSearch = async () => {
    setLoading(true);
    try {
      const options = {
        // your axios request options
        method: "GET",
        url: "https://drug-info-and-price-
history.p.rapidapi.com/1/druginfo?drug=panadol",
        params: { drug: searchTerm },
        headers: {
          'X-RapidAPI-Key':
'1de0c50ea5mshbad7878af63397ep1a07c0jsn00514eba9101',
    'X-RapidAPI-Host': 'drug-info-and-price-history.p.rapidapi.com'
  }
};
      const response = await axios.request(options);
      console.log(response.data); // check the response data
      setSearchResults(response.data);
      setLoading(false);
    } catch (error) {
      console.error(error);
      setLoading(false);
    }
  };
```

5.4. User Interface

By incorporating machine learning models to forecast diseases based on user-provided symptoms and test results, the E-Health smartphone application transforms healthcare. By entering pertinent health data, users may take an active role in their own well-being. The program uses sophisticated algorithms to evaluate the information and generate forecasts in a timely manner.

Following are few examples of User Interfaces:

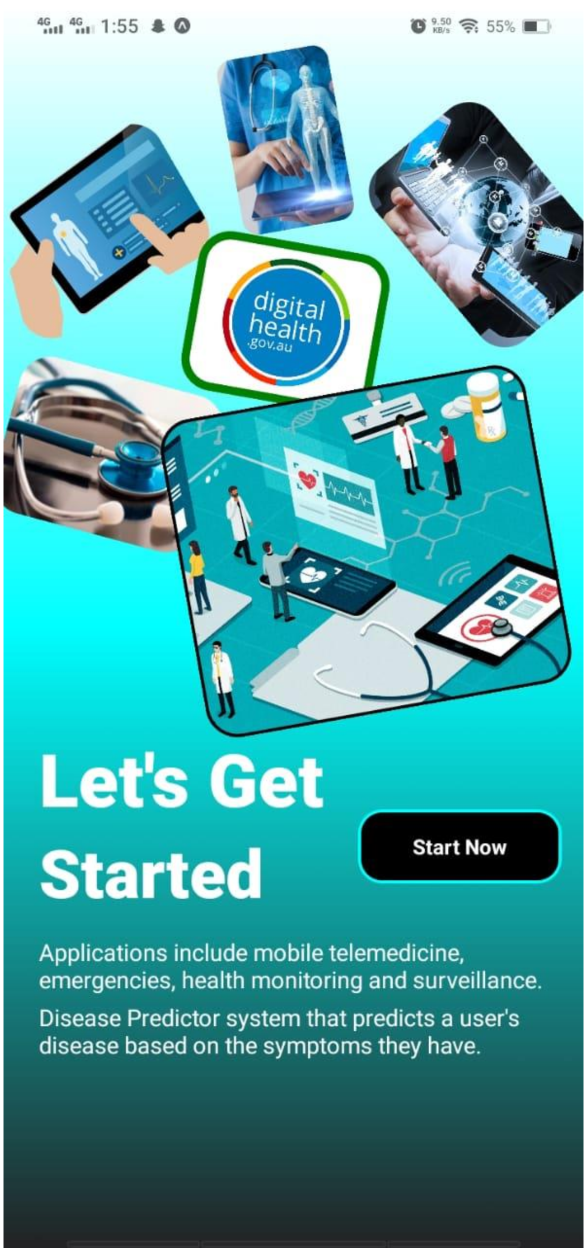
5.4.1 Get Start Screen

- When user open app interface will be show.
- By click Start Now user can start the App

Login Screen

This is Login Screen of Application,

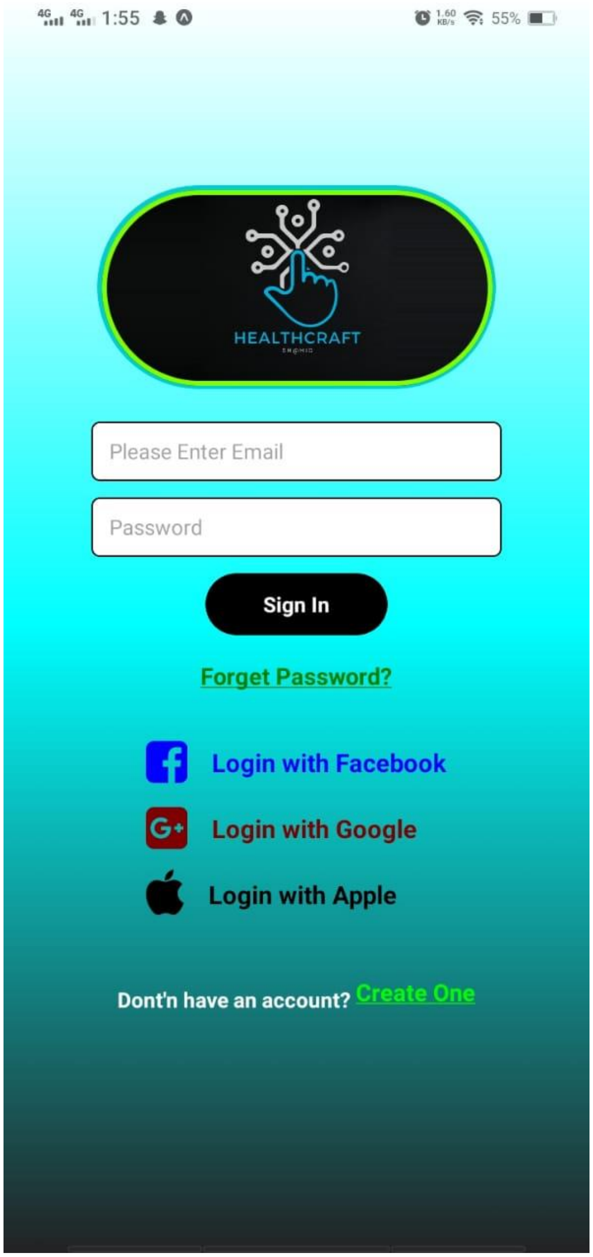
Give Email, Password For login



Let's Get Started

Applications include mobile telemedicine, emergencies, health monitoring and surveillance.

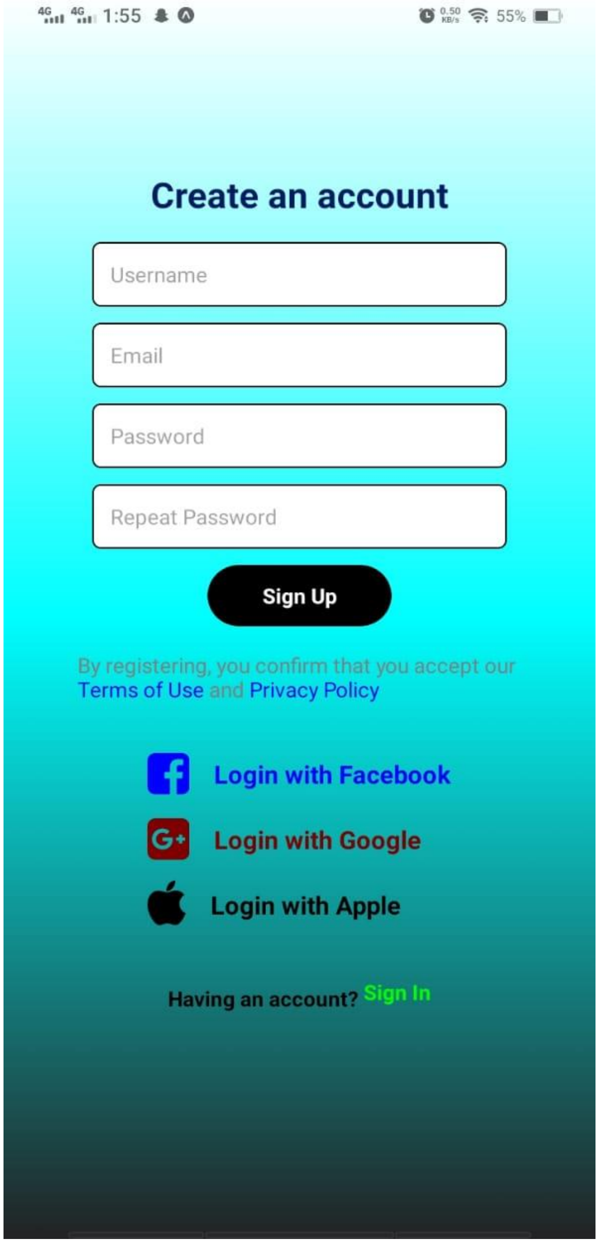
Disease Predictor system that predicts a user's disease based on the symptoms they have.



Dont'n have an account? Create One

5.4.2 SignUp Screen

- For register in firebase.
- User name, email, password.
- Click Signup Button.



4G 4G 1:55 55%

Create an account

Username

Email

Password

Repeat Password

Sign Up

By registering, you confirm that you accept our Terms of Use and Privacy Policy

Login with Facebook

Login with Google

Login with Apple

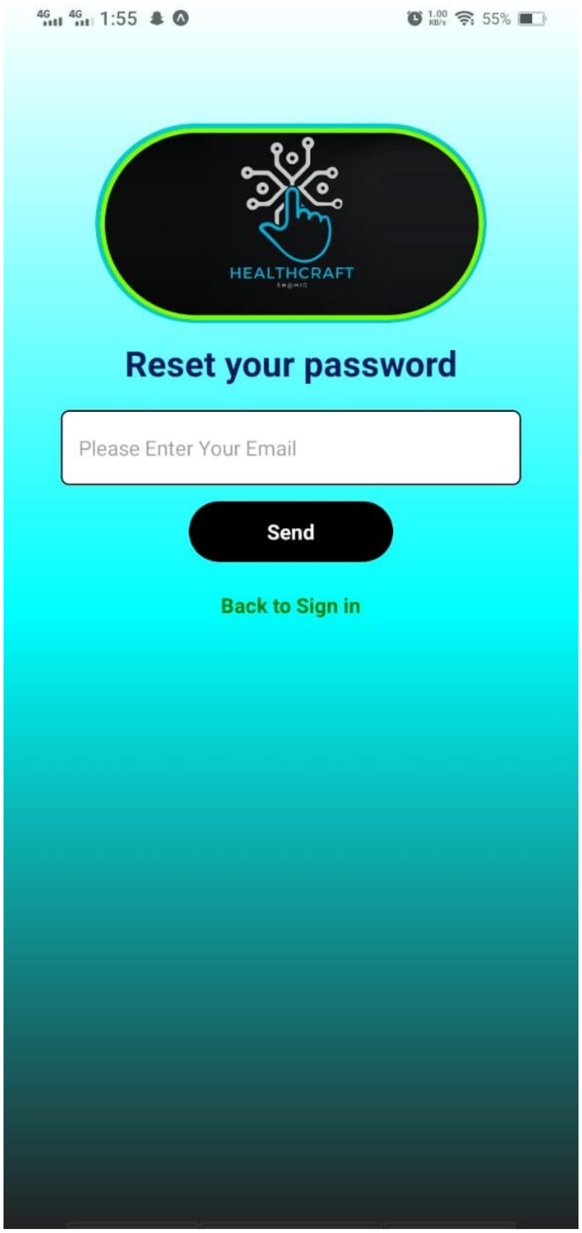
Having an account? Sign In

Forget Password Screen

For Forget password.

Required Email.

Press Send Button.



4G 4G 1:55 55%

HEALTHCRAFT

Reset your password

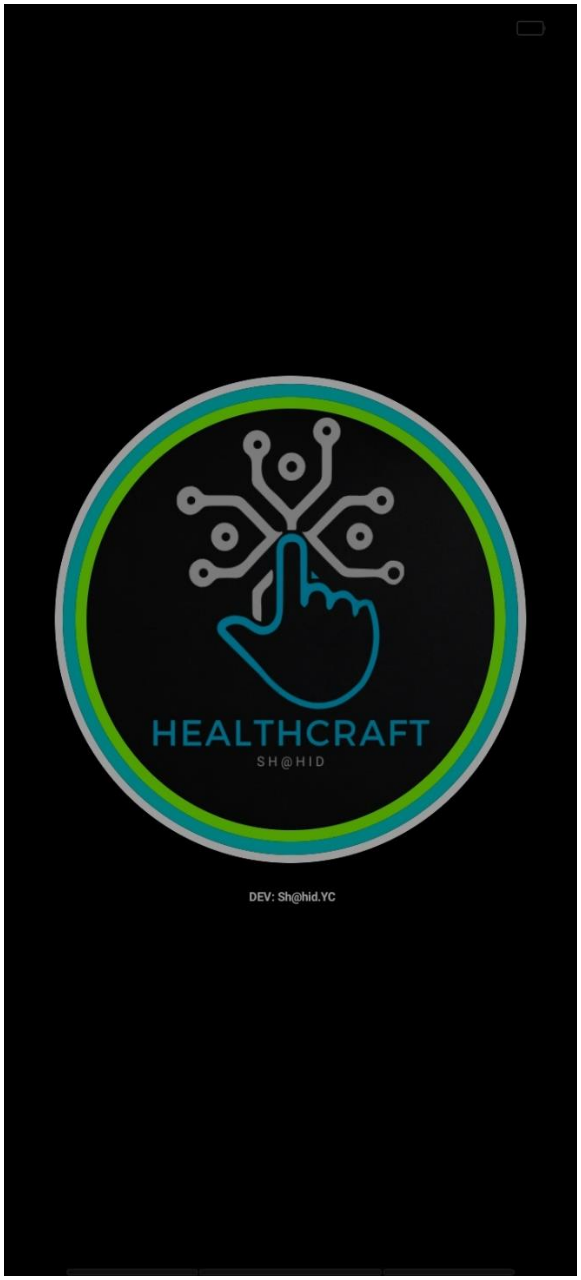
Please Enter Your Email

Send

Back to Sign in

5.4.3 Splash Screen

- After Successful Login.
- Screen will Appear atuo.
- Go Next after Some Seconds

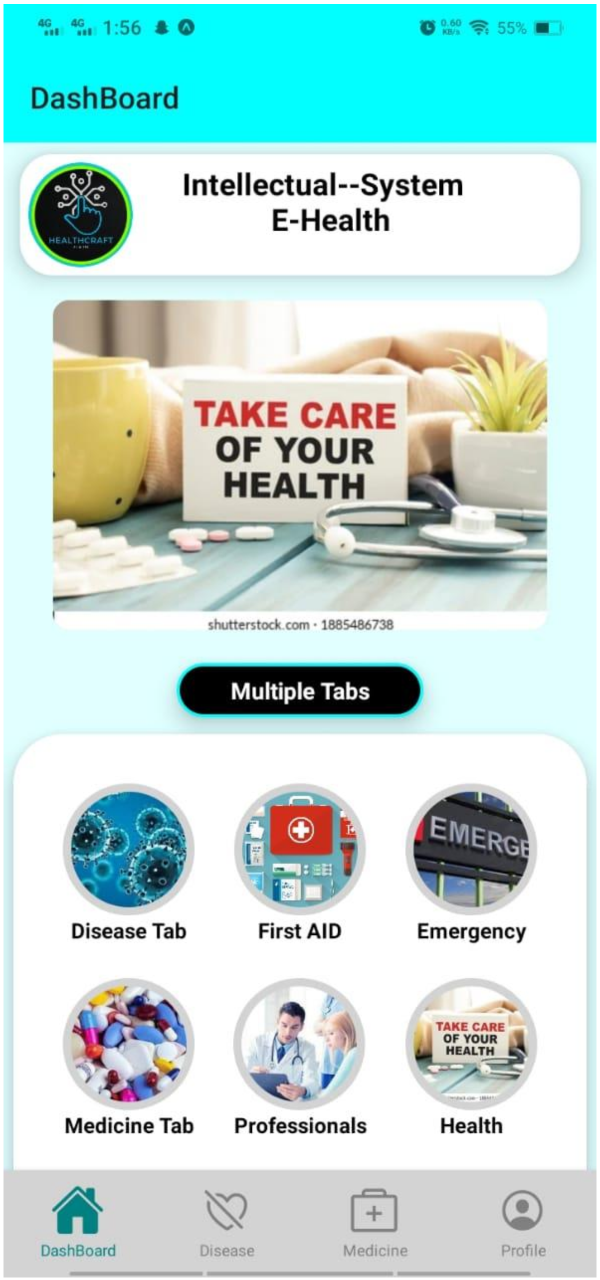


Dashboard Screen

BashBoard Screen After Login,

Perform Different operation,

Different Tabs.



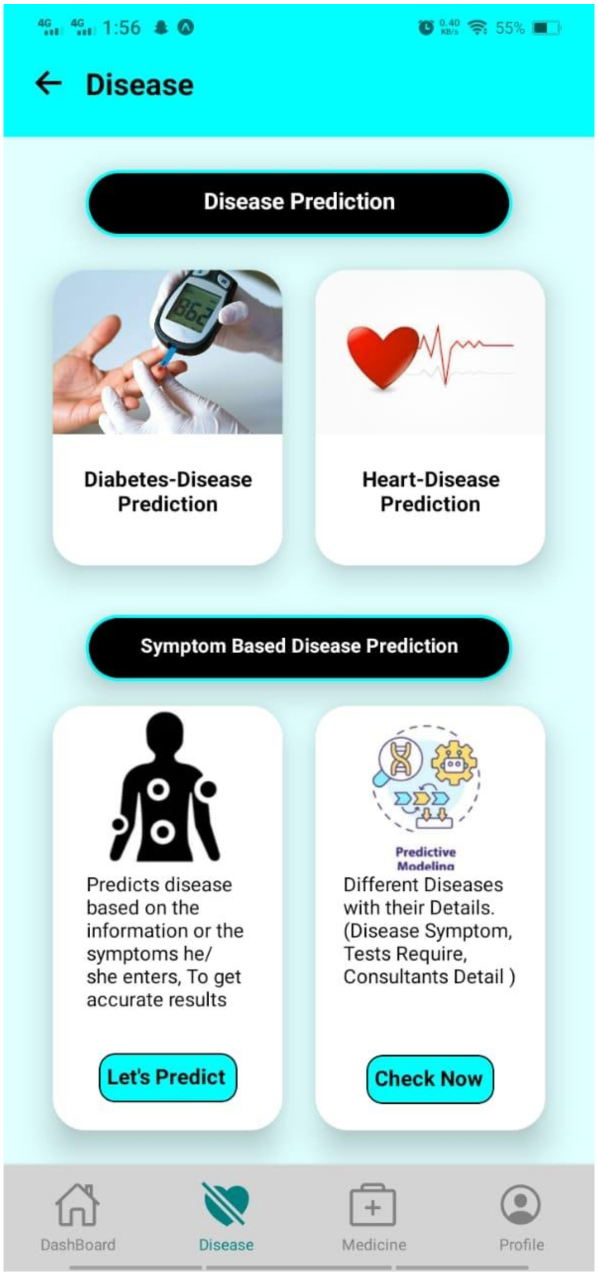
5.4.4 Disease Screen

- Different Prediction will be show,
- For checking Diseases
- Click Let’s Predict Button

Symptom Form

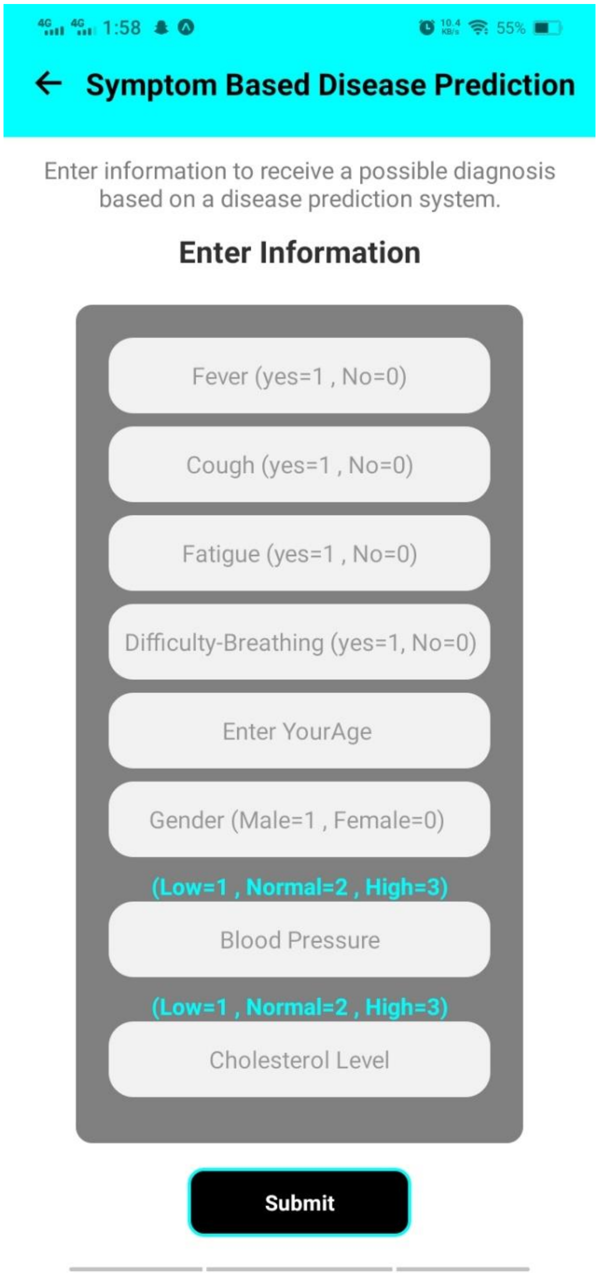
Must give symptoms. ,

Press Submit Button.



Predicts disease based on the information or the symptoms he/ she enters, To get accurate results

Different Diseases with their Details. (Disease Symptom, Tests Require, Consultants Detail)



Enter information to receive a possible diagnosis based on a disease prediction system.

Fever (yes=1 , No=0)

Cough (yes=1 , No=0)

Fatigue (yes=1 , No=0)

Difficulty-Breathing (yes=1, No=0)

Enter YourAge

Gender (Male=1 , Female=0)

(Low=1 , Normal=2 , High=3)

Blood Pressure

(Low=1 , Normal=2 , High=3)

Cholesterol Level

5.4.5 Medicine’s Tab Screen

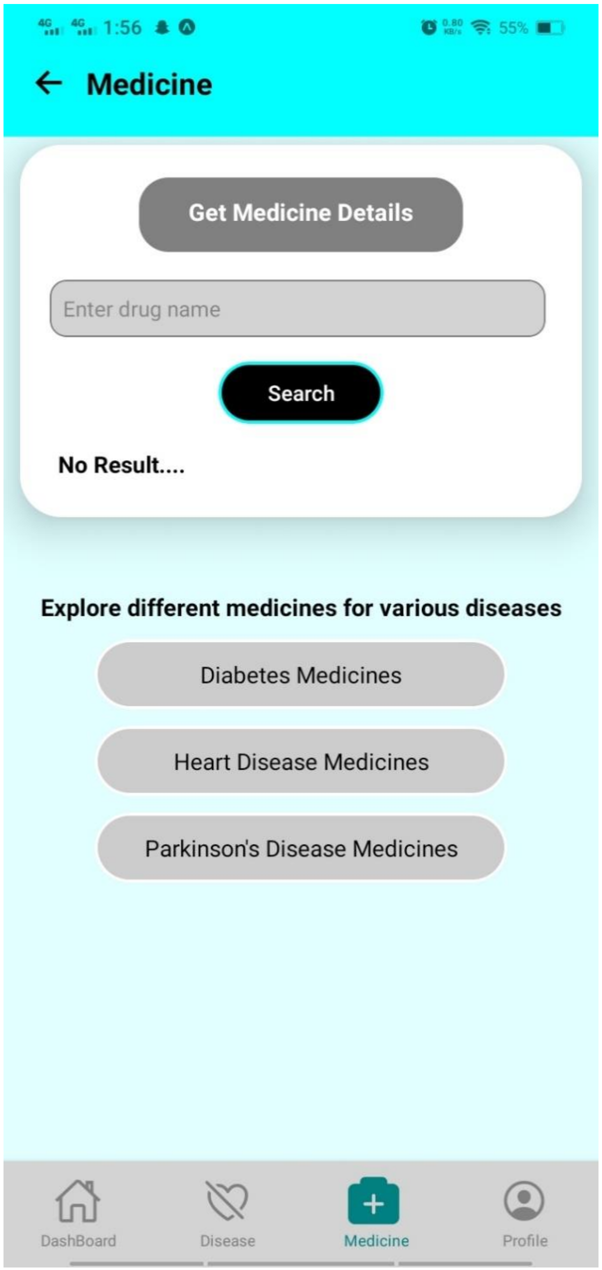
- Search Medicine by name.
- System Give all info about that
- Also Check medicine for disease

Profile

User Email and Name will be shown,

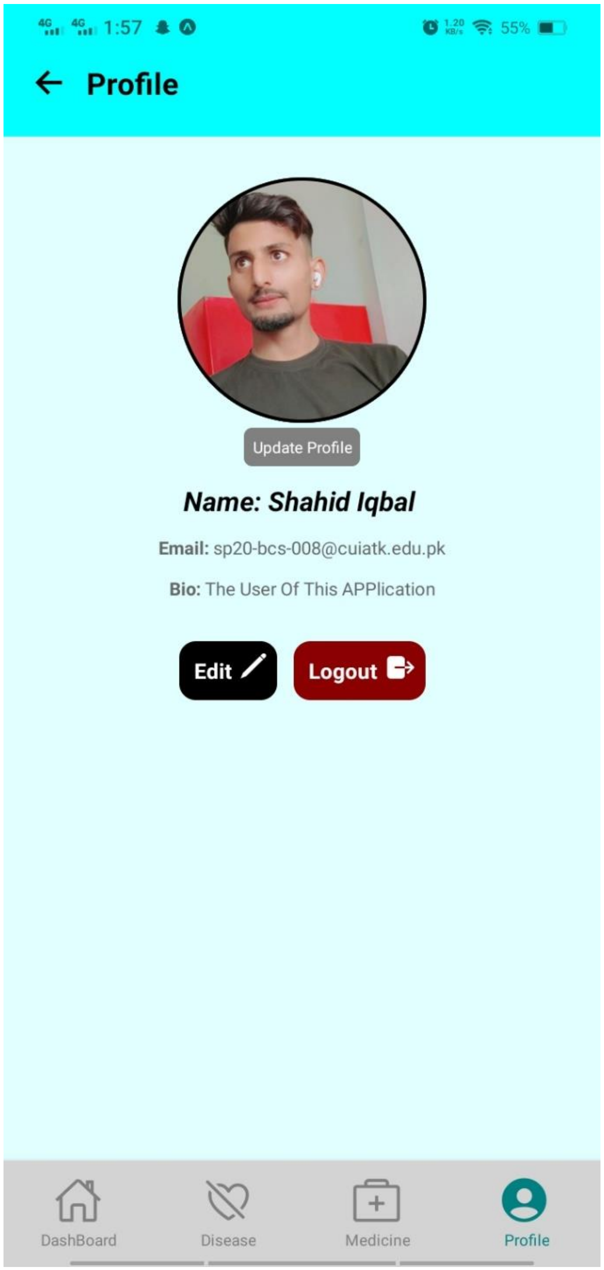
user can update Profile,

also Logout from Here.



No Result....

Explore different medicines for various diseases



Name: Shahid Iqbal

Email: sp20-bcs-008@cuiatk.edu.pk

Bio: The User Of This APPlication

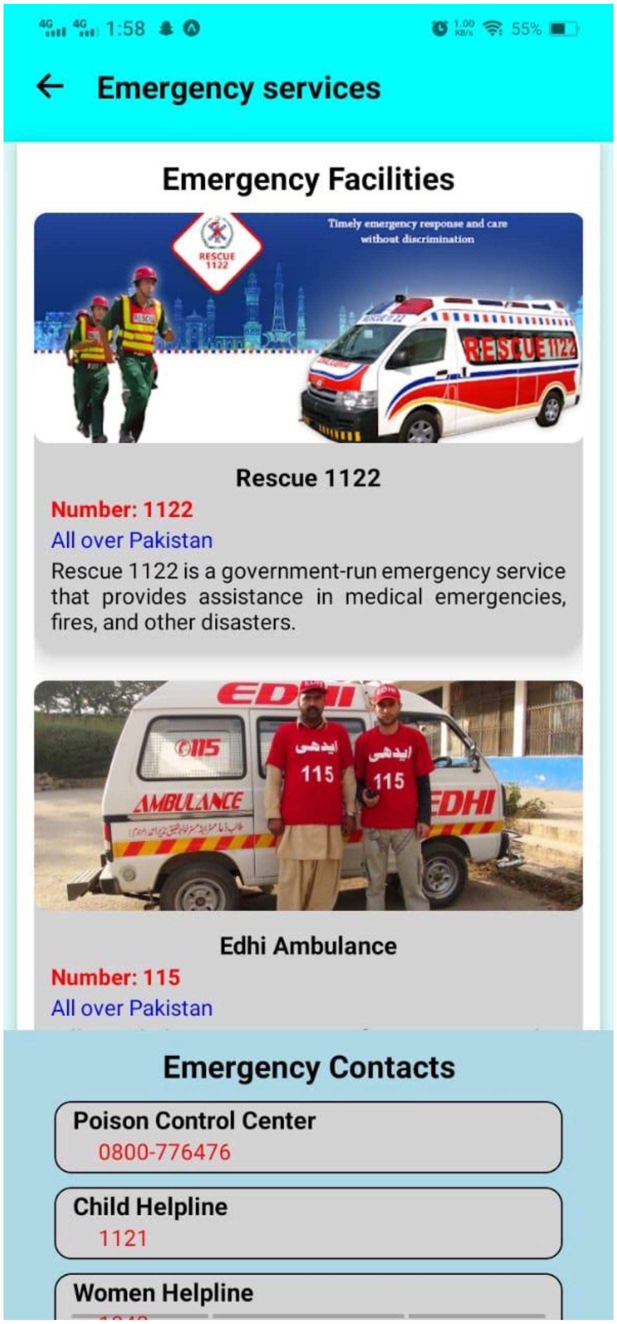
5.4.6 Emergency services Screen

- Emergency condition information.
- Emergency Contacts

First-Aid

Fir Aid Kit and Information,

How we use First Aid



← Emergency services

Emergency Facilities

Rescue 1122

Number: 1122

All over Pakistan

Rescue 1122 is a government-run emergency service that provides assistance in medical emergencies, fires, and other disasters.

Edhi Ambulance

Number: 115

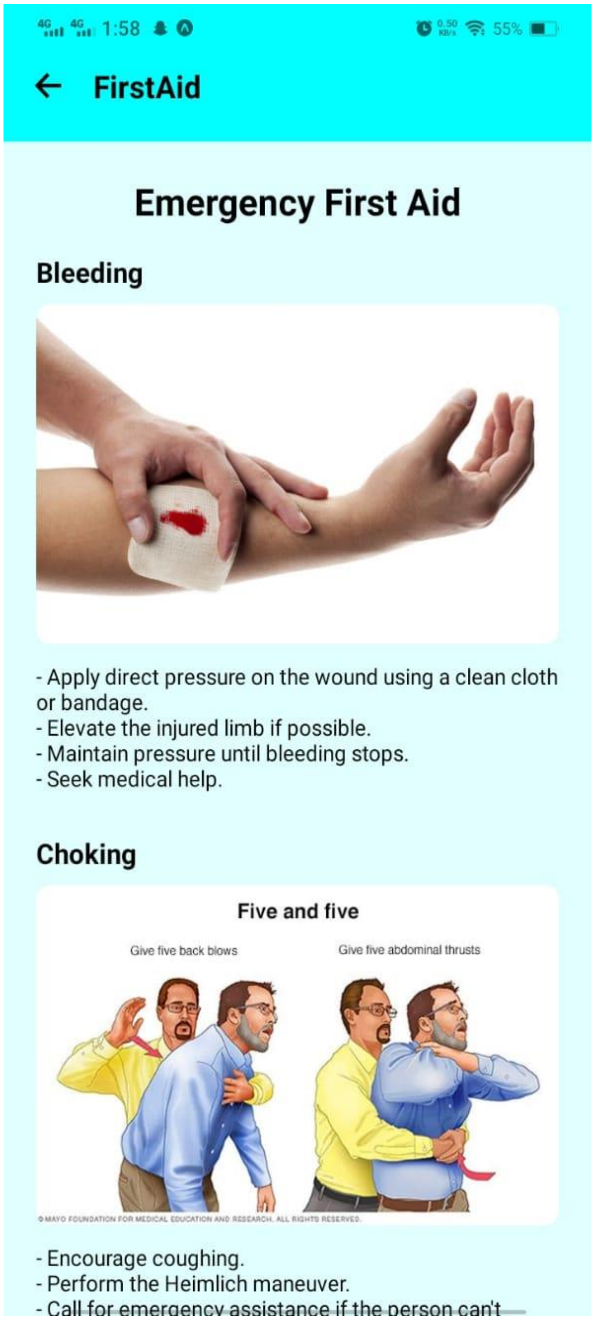
All over Pakistan

Emergency Contacts

Poison Control Center
0800-776476

Child Helpline
1121

Women Helpline


4G 4G 1:58
0.50 KB/s 55%
Five and five
Give five back blows
Give five abdominal thrusts
© MAYO FOUNDATION FOR MEDICAL EDUCATION AND RESEARCH. ALL RIGHTS RESERVED.

← FirstAid

Emergency First Aid

Bleeding

- Apply direct pressure on the wound using a clean cloth or bandage.
- Elevate the injured limb if possible.
- Maintain pressure until bleeding stops.
- Seek medical help.

Choking

- Encourage coughing.
- Perform the Heimlich maneuver.
- Call for emergency assistance if the person can't

5.4.7 Health Screen

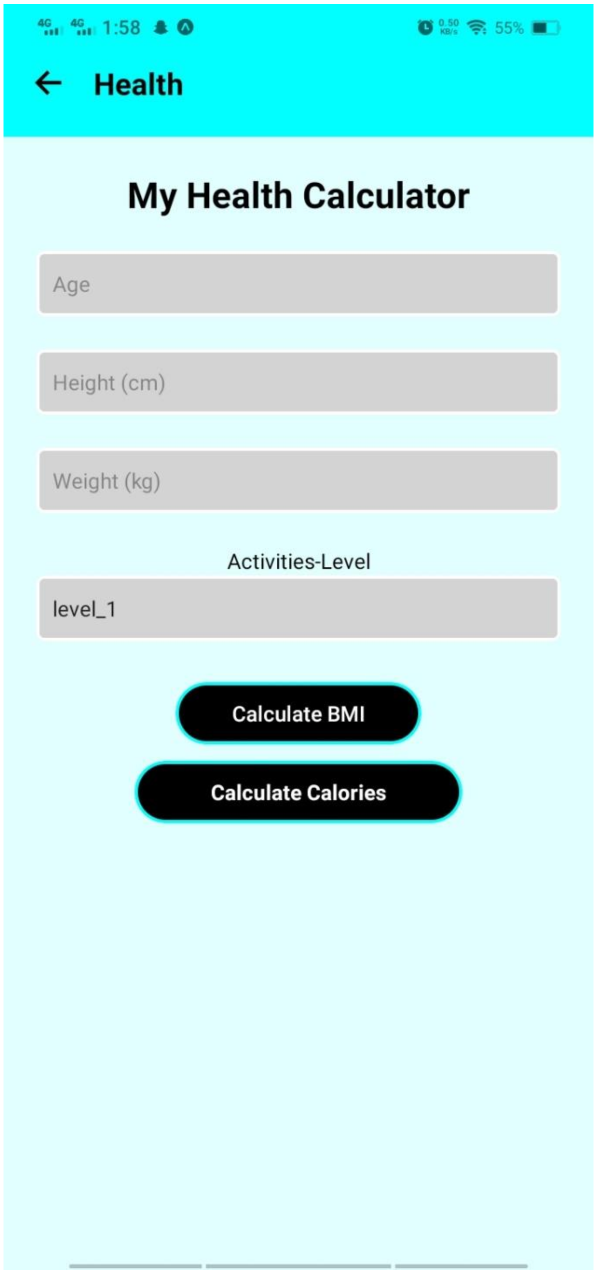
- User can calculate BMI.
- Calculate Calories
- Press button for results

Exercise

Some common exercise available,

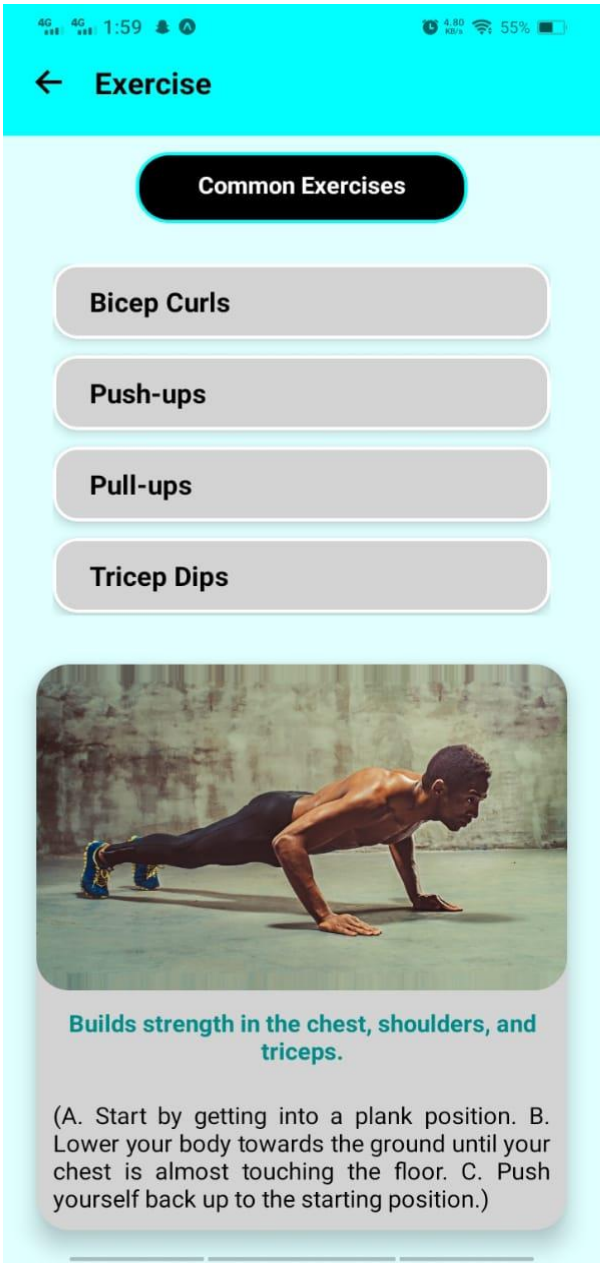
Exercise with how to use

click each and see information.



Activities-Level

level_1



Builds strength in the chest, shoulders, and triceps.

(A. Start by getting into a plank position. B. Lower your body towards the ground until your chest is almost touching the floor. C. Push yourself back up to the starting position.)

5.4.8 Professionals Screen

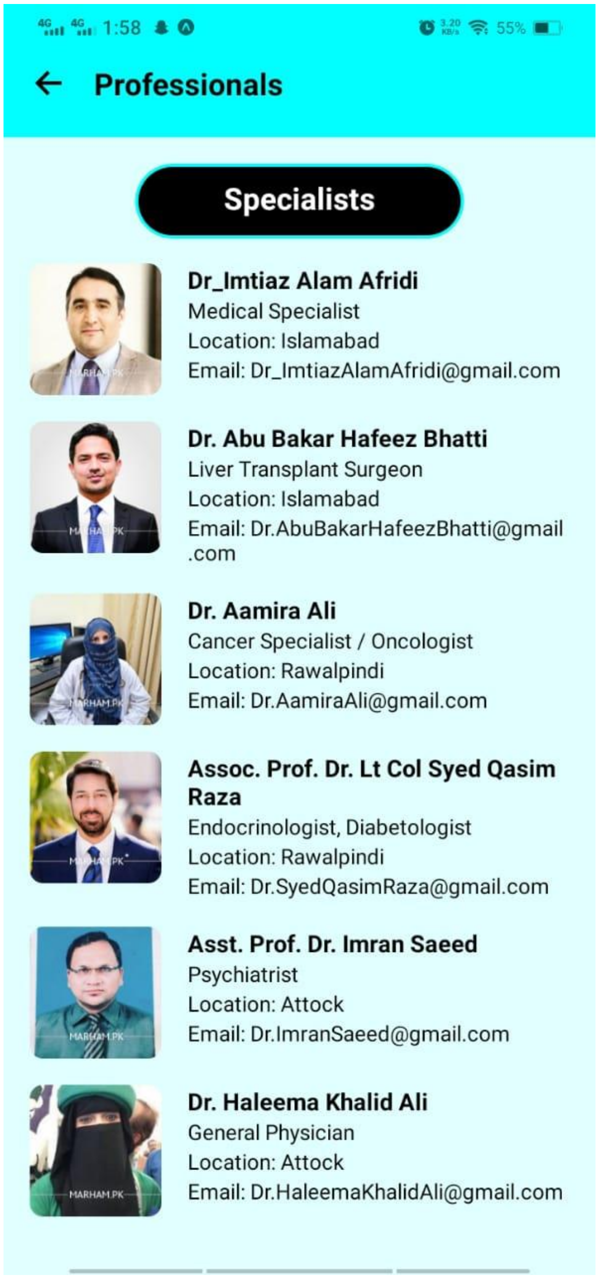
- Specialists List.
- Different Field
- Form different Location

Diet Screen

Diet categories,

With all information

calories, protein, fats, carbs.



Dr_Imtiaz Alam Afridi
Medical Specialist
Location: Islamabad
Email: Dr_ImtiazAlamAfridi@gmail.com

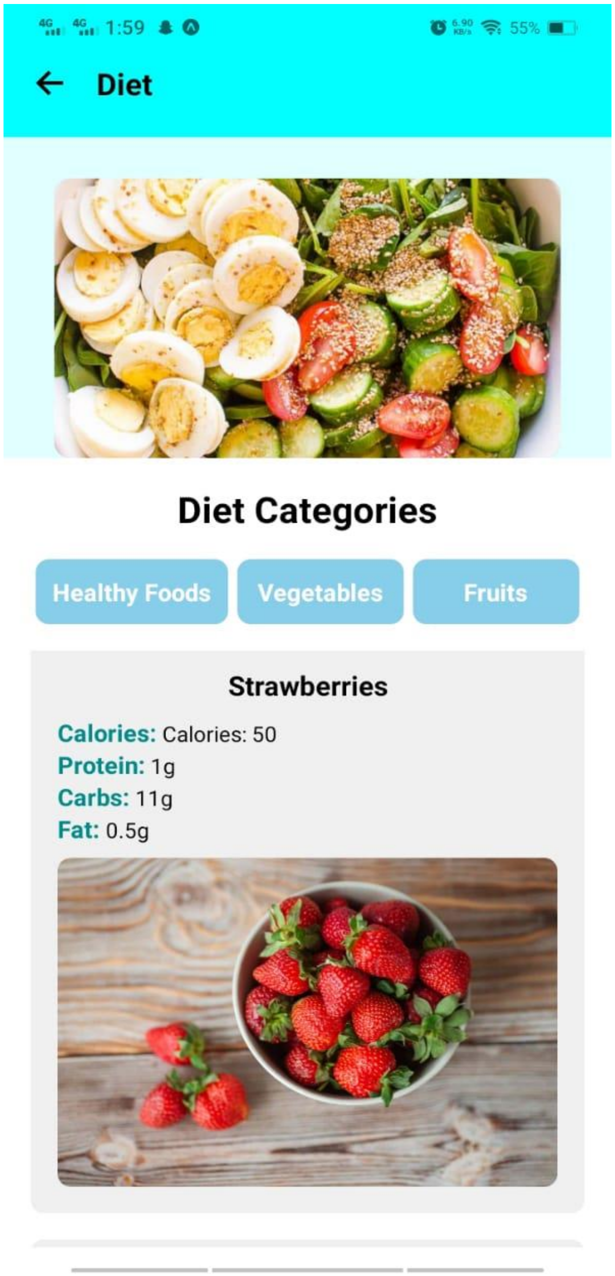
Dr. Abu Bakar Hafeez Bhatti
Liver Transplant Surgeon
Location: Islamabad
Email: Dr.AbuBakarHafeezBhatti@gmail.com

Dr. Aamira Ali
Cancer Specialist / Oncologist
Location: Rawalpindi
Email: Dr.AamiraAli@gmail.com

Assoc. Prof. Dr. Lt Col Syed Qasim Raza
Endocrinologist, Diabetologist
Location: Rawalpindi
Email: Dr.SyedQasimRaza@gmail.com

Asst. Prof. Dr. Imran Saeed
Psychiatrist
Location: Attock
Email: Dr.ImranSaeed@gmail.com

Dr. Haleema Khalid Ali
General Physician
Location: Attock
Email: Dr.HaleemaKhalidAli@gmail.com



Diet Categories

Strawberries

Calories: Calories: 50
Protein: 1g
Carbs: 11g
Fat: 0.5g

6 Testing and Evaluation

Thorough testing has been a vital component of the entire development process for our React Native-based E-Health mobile application for disease prediction, which is integrated with machine learning and API fetching.

6.1. Testing Strategy:

We used a rigorous testing strategy during the development stages, concentrating on seeing and fixing any possible problems along the way. To guarantee a solid and dependable application, we test both functional and non-functional components.

6.2. Bug and Error Resolution:

Utilizing React Native's testing capabilities, we systematically identified and addressed bugs and errors as they emerged. The incorporation of exception handling mechanisms, such as 'try and catch,' has played a crucial role in preventing application crashes, thereby enhancing overall stability.

6.3. Unit Testing

The “E-Health App” has multiple modules and testing is applied on all individual modules at the time of development because every module is interlinked with each other. Testing applied on the module to check its functionalities or not. Debugger is used to check the code functionalities. We also used exception where application crashes due to some issues.

6.4. Function Testing

The complete App is passed through a process to check its working or not. We not only work on functionalities of the application but also work on user friendly and attractiveness of interface. After integrating the system, testing was done on the functions which the application will perform.

6.5. Testing Objectives

The main objectives of passing the App from long process of testing is to fix the bug and improve the performance of app. “Intellectual system for E-Health is a platform that is created to handle all kind of Health Problems and give Information about Health”. To make the app efficient and bug free we apply many testing methods and applying many tests.

6.6. Test cases

Test case contain clearly defined test step for testing a feature an application.

Test case focus on “What to test and How to test”.

6.6.1 Register Test case

Table 7 Registration Test-A

Test Case ID: Register-A	**Test Designed by:** shahid iqbal
Test Priority (Low/Medium/High): High	**Test Designed date:** 16-09-2023
Module Name: User Registration	**Test Executed by:** shahid iqbal
Test Title: Verify Registration process	**Test Execution date:** 16-09-2023
Description: Test the successful User Register process	

Pre-conditions: User must have valid email address

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (P/F)
1	Click on Register Now Button to jump on Register Screen		Display all field with proper design on the registration page	Show all the fields of registration page	Pass
2	Provide valid user information	Username: Ali Email: ali123@gmail.com Password: Ali123 C-password: Ali123	User Successfully filled all the fields	The user filled all the field correctly.	Pass
3	Click on Register Button	Username: Ali Email: ali123@gmail.com Password: Ali123 C-password: Ali123	The system displays the Login Page. User information should be saved in database.	The system displays the Login Page. User information should be saved in database.	Pass

Post-conditions:

The New user successfully register and user information successfully store in database.

6.6.1.1 Registration Test-B

Table 8 Registration Test-B

Test Case ID: Register-B	**Test Designed by:** shahid iqbal
Test Priority (Low/Medium/High): High	**Test Designed date:** 16-09-2023
Module Name: User Registration	**Test Executed by:** shahid iqbal
Test Title: Verify Registration process **Description:** Test the successful User Register process	**Test Execution date:** 16-09-2023

Pre-conditions: User must have valid email address

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (P/F)
1	Click on Register Now Button to jump on Register Screen		Display all field with proper design on the registration page	Show all the fields of registration page	Pass
2	Provide valid user information	Username: Ali Email: Password: Ali123 C-password: Ali123	Error message shown because user do not provide an email address	The user filled all the field correctly.	Fail
3	Click on Register Button	Username: Ali Email: Password: Ali123 C-password: Ali123	The system displays an error massage "**Please Enter email address**".	The system displays the dashboard. User information should be saved in database.	Fail

Post-conditions:

The User does not register.

6.6.2 Login Test Cases

Table 9 Login Test-A

Test Case ID: Login-A	**Test Designed by:** shahid iqbal
Test Priority (Low/Medium/High): High	**Test Designed date:** 16-05-2022
Module Name: User Login	**Test Executed by:** shahid iqbal
Test Title: Verify Login process	**Test Execution date:** 16-05-2022
Description: Test the successful User Login process	

Pre-conditions: User must have registered email address

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (P/F)
1	Click on Login Now Button to jump on Login Screen		Display all field with proper design on the Login page	Show all the fields of Login page.	Pass
2	Enter valid Email address and Password	Email: ali123@gmail.com Password: Ali123	User Successfully filled all the fields	The user filled all the field correctly.	Pass
3	Click on Login Button	Email: ali123@gmail.com Password: Ali123	The system checks the given detail in the database and display Dashboard.	The system displays the Dashboard.	Pass

Post-conditions:

User is validated with database and successfully login to account.

6.6.2.1 Login Test-B

Table 10 Login Test-B

Test Case ID: Login-B	**Test Designed by:** shahid iqbal
Test Priority (Low/Medium/High): High	**Test Designed date:** 16-09-2023
Module Name: User Login	**Test Executed by:** shahid iqbal
Test Title: Verify Login process	**Test Execution date:** 16-09-2023
Description: Test the successful User Login process	

Pre-conditions: User must have verified email address

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (P/F)
1	Click on Login Now Button to jump on Login Screen		Display all field with proper design on the Login page	Show all the fields of Login page.	Pass
2	Enter valid Email address and Password	Email: ali659@gmail.com Password: Ali123	Show error massage	The user filled all the field correctly.	Fail
3	Click on Login Button	Email: ali659@gmail.com Password: Ali123	The system displays the error massage "The email does not exit".	The system displays the Dashboard.	Fail

Post-conditions:

User is not validated with database and unable to Login.

6.6.3 Forget Password Test Case:

Table 11 Forget password Test-A

Test Case ID: Forget Password-A	**Test Designed by:** shahid iqbal
Test Priority (Low/Medium/High): Med	**Test Designed date:** 18-09-2023
Module Name: Forget Password	**Test Executed by:** shahid iqbal
Test Title: Verify Forget Password process	**Test Execution date:** 18-09-2023
Description: Test the Reset Password process	

Pre-conditions: User must have an account for forget password.

A user may forget his/her password. They can reset it by giving their email address.

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (P/F)
1	Click on the Forget password button on the login page		Display all field with proper design on the reset password page	Show all the fields of reset password page.	Pass
2	Enter valid Email address	Email: ali123@gmail.com	User Successfully filled all the fields	The user filled the entire field correctly.	Pass
3	Click on send Button	Email: ali123@gmail.com	The system checks the provided detail in the database	The system sends the link to the provided email to change password.	Pass

Post-conditions:

User is validated with database and successfully reset password.

6.6.3.1 Test Case for Forget Password-B

Table 12 Forget password Test-B

Test Case ID: Forget Password-B	**Test Designed by:** Muhammad Usama
Test Priority (Low/Medium/High): Med	**Test Designed date:** 16-05-2022
Module Name: Forget Password	**Test Executed by:** M. Usama
Test Title: Verify Forget Password process	**Test Execution date:** 16-05-2022
Description: Test the Reset Password process	

Pre-conditions: User must have an account for forget password.

A user may forget his/her password. They can reset it by giving their email address.

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (P/F)
1	Click on the Forget password button on		Display all field with proper design on the reset	Show all the fields of reset password	Pass

	the login page		password page	page.	
2	Enter valid Email address	Email: ali569@gmail.com	Show error massage	The user filled the entire field correctly.	Fail
3	Click on send Button	Email: ali569@gmail.com	The system displays the error massage “**The email does not exit**”	The system sends the reset password link to the provided email.	Fail

Post-conditions:

User unable to reset password.

6.7. Functional Testing

6.7.1 Symptom based Disease prediction Test:

Table 13 Symptom based Disease prediction Test

User Input	Expected Prediction	Actual Prediction	Status
Headache, fever, sore throat	Flu	Flu	Positive
Cough, runny nose, sneezing	Common cold	Common cold	Positive
Nausea, vomiting, diarrhea	Food poisoning	Food poisoning	Positive
Fatigue, muscle aches, joint pain	Dengue fever	Dengue fever	Positive
Painful urination, frequent urination	Urinary tract infection	Urinary tract infection	Positive
No symptoms	No disease	No disease	Negative
Rash, itching, sneezing	Allergic reaction	Allergic reaction	Positive
Shortness of breath, chest pain	Heart attack	Heart attack	Positive
Dizziness, fainting, confusion	Stroke	Stroke	Positive
Seizures, loss of consciousness	Epilepsy	Epilepsy	Positive

6.7.2 Diabetes Disease prediction Test:

Table 14 Diabetes Disease prediction Test

User Input	Expected Prediction	Actual Prediction	Status
Age: 35, Sex: Male, BMI: 30, Blood Pressure: 130/80, Glucose: 120, Insulin: 200, HOMA: 3.5	Positive	Positive	Positive
Age: 45, Sex: Female, BMI: 25, Blood Pressure: 120/70, Glucose: 100, Insulin: 150, HOMA: 2.5	Negative	Negative	Negative
Age: 55, Sex: Male, BMI: 40, Blood Pressure: 140/90, Glucose: 150, Insulin: 300, HOMA: 5.0	Positive	Positive	Positive
Age: 25, Sex: Female, BMI: 20, Blood Pressure: 110/60, Glucose: 90, Insulin: 100, HOMA: 1.5	Negative	Negative	Negative
Age: 65, Sex: Male, BMI: 35, Blood Pressure: 150/100, Glucose: 200, Insulin: 400, HOMA: 7.0	Positive	Positive	Positive
Age: 30, Sex: Female, BMI: 22, Blood Pressure: 125/75, Glucose: 110, Insulin: 175, HOMA: 3.0	Positive	Negative	False Negative
Age: 40, Sex: Male, BMI: 27, Blood Pressure: 115/65, Glucose: 95, Insulin: 125, HOMA: 2.0	Negative	Positive	False Positive
Age: 50, Sex: Female, BMI: 32, Blood Pressure: 135/85, Glucose: 140, Insulin: 250, HOMA: 4.0	Positive	Positive	True Positive
Age: 20, Sex: Male, BMI: 18, Blood Pressure: 105/55, Glucose: 85, Insulin: 75, HOMA: 1.0	Negative	Negative	True Negative

6.7.3 Heart Disease prediction Test:

Table 15 Heart Disease prediction Test

User Input	Expected Prediction	Actual Prediction	Status
Age: 65	Positive	Positive	At risk of heart disease
Age: 40	Negative	Negative	Not at risk of heart disease
Sex: Male	Positive	Positive	At risk of heart disease
Sex: Female	Negative	Negative	Not at risk of heart disease
Chest pain type: Angina	Positive	Positive	At risk of heart disease
Chest pain type: Non-anginal pain	Negative	Negative	Not at risk of heart disease
Resting blood pressure (in mm Hg): 140	Positive	Positive	At risk of heart disease
Resting blood pressure (in mm Hg): 120	Negative	Negative	Not at risk of heart disease
Cholesterol (mg/dl): 240	Positive	Positive	At risk of heart disease
Cholesterol (mg/dl): 180	Negative	Negative	Not at risk of heart disease
Fasting blood sugar (mg/dl): > 120	Positive	Positive	At risk of heart disease
Fasting blood sugar (mg/dl): < 120	Negative	Negative	Not at risk of heart disease
Electrocardiographic results: Abnormal	Positive	Positive	At risk of heart disease
Electrocardiographic results: Normal	Negative	Negative	Not at risk of heart disease

6.8. Business Rules Testing

Testing the E-Health app's business rules involves thoroughly checking if the system follows predefined rules in its operations. This includes ensuring that disease prediction models, personalized health information, and emergency response guidelines align with established standards. The testing also covers privacy and security rules, validating the accuracy of predictive algorithms, and assessing the relevance of health education content. The goal is to confirm that the app not only meets regulatory requirements but also provides a reliable and trustworthy user experience for proactive health management.

Table 16 Disease Prediction Model Rules

Rule	**Description**
Rule 1	The disease prediction models must be based on current medical research and evidence.
Rule 2	The models must be trained on a large and diverse dataset of patient data.
Rule 3	The models must be regularly evaluated to ensure their accuracy and reliability.

Table 17 Personalized Health Information Rules

Rule	**Description**
Rule 1	The app must collect and analyze user data in a secure and confidential manner.
Rule 2	The app must only use user data to provide personalized health information.
Rule 3	The app must provide users with the ability to control how their data is used.
Rule 4	The app must clearly communicate how user data is being used to users.

Table 18 Predictive Algorithm Accuracy Rules

Rule	**Description**
Rule 1	The predictive algorithms must be evaluated on a regular basis to ensure their accuracy.
Rule 2	The app must clearly communicate the accuracy of the predictive algorithms to users.
Rule 3	The predictive algorithms must be updated regularly to reflect new data and research.

7 Conclusion and Future Work

E-Health app system has the potential to be a valuable tool for improving the health and well-being of its users. With continued development and refinement, the app can become an even more powerful resource for personal health management.

7.1. Conclusion

The E-Health app system has the potential to significantly improve the lives of its users by providing them with convenient and personalized access to a variety of healthcare services. The symptom-based disease prediction model, diabetes disease prediction model, and heart disease prediction model can help users identify potential health problems early on and get the care they need. The app also provides users with information on diet, health, exercise, first aid, emergencies, and medicine tabs. This information can help users make informed decisions about their health and well-being.

7.2. Future Work

The app's future development will focus on improving the accuracy of prediction models, adding new features, and expanding the app's coverage. There are a number of areas where the E-Health app system can be improved in the future. One area of focus is on improving the accuracy of the prediction models. This could be done by collecting more data and using more sophisticated machine learning algorithms. Another area of focus is on developing new features for the app. For example, the app could be integrated with wearable devices to track users' health data. The app could also be used to connect users with healthcare providers for virtual consultations. E-Health App System plans to expand its features to include a personalized health plan, a medication reminder system, and a virtual doctor consultation service. These new features will further enhance the app's ability to provide users with comprehensive healthcare support. Additionally, the app will continue to utilize machine learning and AI to improve the accuracy of its prediction models and develop new features that address the evolving needs of its users.

8. References

Datasets, Article’s, React Native, Tool’s, Api:

https://www.kaggle.com/datasets/uom190346a/disease-symptoms-and-patient-profile-dataset/data
https://www.kaggle.com/code/thedankdel/disease-symptom-prediction-ml-99/notebook
https://www.kaggle.com/code/qaisalghanim/medical
https://www.kaggle.com/datasets/iammustafatz/diabetes-prediction-dataset
https://www.kaggle.com/datasets/johnsmith88/heart-disease-dataset
https://annalsofrscb.ro/index.php/journal/article/view/6151
https://www.sciencedirect.com/science/article/pii/S1877050920300557
https://iopscience.iop.org/article/10.1088/1757-899X/1022/1/012072/pdf
https://www.irjmets.com/uploadedfiles/paper/issue_5_may_2022/24065/final/fin_irjmets1653367944.pdf
https://ieeexplore.ieee.org/document/9914945
https://ieeexplore.ieee.org/document/9388603
https://iopscience.iop.org/article/10.1088/1757-899X/1022/1/012072/meta
https://www.hindawi.com/journals/cin/2021/8387680/
https://www.ncbi.nlm.nih.gov/pmc/articles/PMC10107388/
https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9690067/
https://reactnative.dev/docs/getting-started
https://reactnative.dev/docs/javascript-environment
https://reactnative.dev/docs/native-modules-android
https://reactnative.dev/docs/native-components-android
https://reactnative.dev/docs/components-and-apis
https://reactnative.dev/docs/accessibilityinfo
https://reactnative.dev/architecture/overview
https://reactnative.dev/community/overview
https://reactnative.dev/docs/colors
https://legacy.reactjs.org/docs/lists-and-keys.html
https://rapidapi.com/products/api-hub/
https://katalon.com/api-testing?utm_term=free%20api&utm_campaign=PK_Search_Non_Brand_API&utm_source=adwords&utm_medium=search&hsa_acc=4390546474&hsa_cam=19792555331&hsa_grp=147755054598&hsa_ad=650982288842&hsa_src=g&hsa_tgt=kwd-298648206390&hsa_kw=free%20api&hsa_mt=b&hsa_net=adwords&hsa_ver=3&gad_source=1&gclid=CjwKCAiAjfyqBhAsEiwA-UdzJE4rGCldTqABHOYvFuzsbOoZvmdl3jIT-FkhbEM8oNljlaTkuS1OihoCXrMQAvD_BwE
https://devapi.ai/?gclid=CjwKCAiAjfyqBhAsEiwA-UdzJD1ynnxqwfRWzh4ZzB5m79ZynlSxaspnEOjpiif3v_168XYiW0owPBoCa-8QAvD_BwE