MAKERERE UNIVERSITY

COLLEGE OF COMPUTING AND INFORMATION SCIENCES

DEPARTMENT OF NETWORKS

BACHELOR OF SCIENCE IN SOFTWARE ENGINEERING

Software Design Document

For

A project on

HEALTH API AND MOBILE APPLICATION SYSTEM

Supervisor: Dr. Rashidah Kasauli

Prepared By:

Group BSE23-9

NAME	**REGISTRATION NUMBER**
JOSHUA GIFT KARAMUZI	19/U/11186/EVE
EMOKOL HILLARY	19/U/8797/EVE
NGABIRANO ISAAC	19/U/12546/PS
RUTAHIGWA EMMANUEL NOEL	19/U/12154/PS

Contents

Table of figures

1 INTRODUCTION

1.1 Purpose

The purpose of this SDD is to describe the system design of the HEALTH-API and MOBILE APPLICATION SYSTEM proposed by BSE23-9. We address the key business rules that the project tends to address in the Ugandan Health Sector, detailing the functionality, components and modules. Modifications to this document are subject to version control to track its history. The intended audience for this SDD would include developers, software architects, project managers, and other stakeholders involved in the development process of the HEALTH-API and MOBILE APPLICATION SYSTEM. It may also include healthcare professionals who will be using the system to manage their day-to-day operations.

1.2 Scope

The HEALTH-API and MOBILE APPLICATION SYSTEM provides a digital platform in form of an application that consumes the services provided by the API like managing patient medical reports, AI powered notifications based on the prescription, give recommendations to of mild symptoms to patients through the doctor smart module and Data Explorer which will enable the health care providers predict the out break of a diseases in the society.

Note:

Only four divisions are to be used as the divisions were patients will be coming from as the test case for the system i.e. Makindye, Rubaga, Nakawa and Kawempe.

Only three diseases will be considered as tested from patients i.e. malaria, typhoid and HIV/AIDS.

1.2.1. The benefits of the HEALTH-API and MOBILE APPLICATION;

- It will Improve patient care and outcomes through more efficient and accurate management of patient records and treatment plans.
- It will increase the operational efficiency through automation of processes, reducing errors and redundancy.

- It will provide streamlined inventory management, reducing waste and lowering costs.
- It will enhanced reporting capabilities, providing healthcare providers with valuable insights into their operations and performance.

1.3 Overview

The Software Design Document for the HEALTH-API and MOBILE APPLICATION SYSTEM typically includes the following sections:

- **Introduction.** This section provides an overview of the document, including its purpose, scope, and intended audience.

System Overview. This section gives a general description of the functionality, context and design of your project. Provide any background information if necessary

System Architecture. This describes the high-level structure and organization of the software system being designed.

Data Design. This section describes how data will be stored, processed, and managed within the software system being designed.

- **Component Design:** In this section, we take a closer look at what each component does in a more systematic way.

Human Interface Design. This section shows how the user interacts with the software system being designed. It includes details about the user interface design, the user experience, and the overall usability of the software.

Requirements Matrix. This section will provide a cross-reference that traces components and data structures to the requirements in our SRS document.

- **Appendices:** This section may be included, either directly or by reference, to provide supporting details that could aid in the understanding of the Software Design Document.

1.4 Reference Material

List any documents, if any, which were used as sources of information for the test plan.

1.5 Definitions and Acronyms

Figure 1. This is a table for the abbreviations

Acronyms/ Abbreviations	**Acronyms/ Abbreviations**
SDD	*Software Design Document*
API	*Application Programming Interface*
AI	*Artificial Intelligence*

2. SYSTEM OVERVIEW

2.1. A GENERAL DESCRIPTION OF THE FUNCTIONALITY

The HEALTH-API and MOBILE APPLICATION SYSTEM will provide a digital platform inform of an application that consumes the services provided by the API which include;

- **Data Explorer**. The system will provide analytics that will be used to predict an outbreak of a particular disease in order to help the health institution prepare to combat the outbreak.
- **Patient medical reports.** The system will provide medical reports of patients to the health personals and enables them to edit or view them through its access of the hospital database using the API.
- **AI powered notifications**. The system will provide notifications to patients that will be triggered on incase the prescription is entered the database to remind them to take their medication based on the prescription entered in the database of the patient’s health center and they are cancelled automatically in case the prescription is done.

- **Doctor Smart.** The system will provide recommendations on how to treat the mild symptoms communicated by the patient and incase the symptoms communicated are severe the patient is recommended to see a doctor.

2.2. CONTEXT AND DESIGN.

HEALTH-API and MOBILE APPLICATION system is a software application that will be designed to help healthcare organizations and patients(users) manage various aspects of their operations through the API services consumed by the mobile application and they include patient medical reports, predictions, recommendations and AI powered reminders. The system will also include a mobile application that will consume these API services.

In terms of design, the HEALTH-API and MOBILE APPLICATION system will be user-friendly and intuitive with a well-designed user interface that allows healthcare professionals to quickly access information they need and perform their tasks efficiently. The system's API will also be flexible and customizable allowing integration with the mobile application and tailor it to its specific needs and workflows. The system will also be able to provide recommendations to patients and it will also provide notifications to patients as reminders. The system will provide analysis that will be used by the health personnels to predict an out break of a disease in the society.

3. SYSTEM ARCHITECTURE

3.1.Architectural Design

The HEALTH-API and MOBILE APPLICATION system can be broken down into two main subsystems i.e. The health-API subsystem and The Mobile Application subsystem.

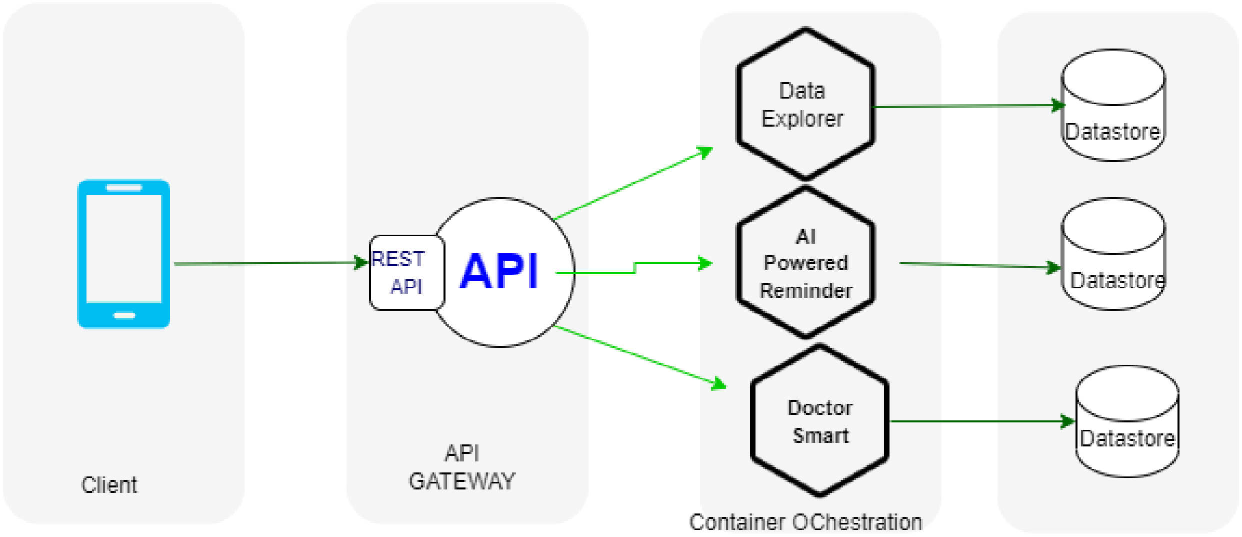


Figure 2. Showing the micro service architecture for the system

3.1.1. The health-API subsystem.

The API subsystem will enable communication and data exchange between the hospital database and the mobile application. The following modules are found in the API-subsystem as services offered by it;

- **Doctor Smart**. This will consist of a chatbot that will provide recommendations to mild diseases of to the patients with the help of natural language processing based on their communicated symptoms and incase the symptoms really require you to see a doctor the chatbot would recommend you to go see a doctor in a hospital or it recommends you to go to the consultation submodule in the telemedicine module so as to schedule an online consultation from the doctors or health personnel.
- Patient Medical reports. This will enable doctors or a health personnel working on a patient be able to see the patients past medical record in order to know the kind of patient they are dealing with and this will all be made possible by the health personnel making an API call through the frontend by querying the database to return a patient medical report and view it.
- **AI powered notifications**. These will be sent to the patient based on the medical record stored in the database and they are activated as soon as a medication is given to a

patient and the details are entered in the database. These notifications will have self-awareness and if the patient's time for a specific medication is done, they will cancel themselves from being received by that patient.

3.1.2. The Mobile Application subsystem.

The Mobile Application sub system shall comprise three layers, thus following the layered architecture. These layers shall include; Presentation layer, Business Logic layer, and the data layer.

These layers perform the following function;

- Business Logic Layer. It will be responsible for implementing the business logic that governs the behavior of the application. It processes data received from the API, applies the necessary business rules, and sends the results back to the presentation layer. It also handles security and authorization checks, ensuring that only authorized users can access sensitive data.
- Presentation Layer. The presentation layer will be responsible for displaying the application's user interface and handling user input. It receives data from the Business Logic Layer and sends user input back to the Business Logic Layer for processing. The presentation layer will be accessed through a mobile application developed for Android phones.
- Data Layer. The data layer will be responsible for storing and retrieving data from the remote database accessed through the API. It will provide an abstraction of the database to the Business Logic Layer, allowing the Business Logic Layer to interact with the data without knowing the details of how the data is stored or retrieved. The data layer will also handle the data mapping between the database and the Business Logic Layer ensuring that the data is in the correct format for processing.

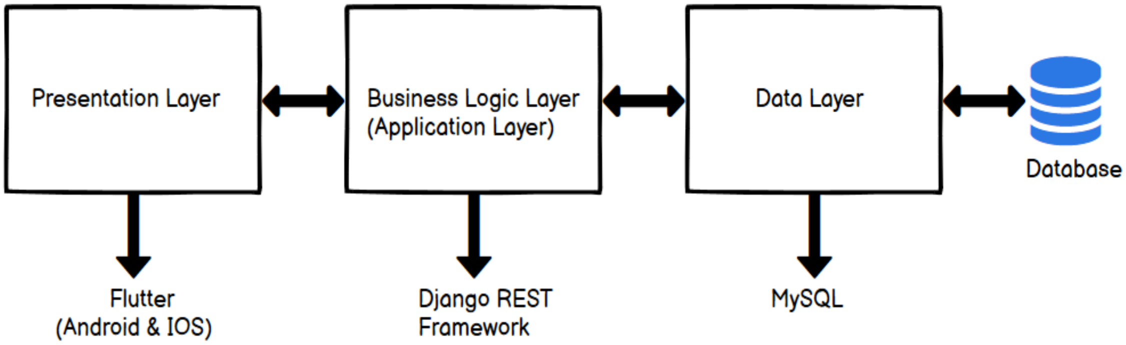


Figure 3. The figure below shows a layered architecture for the mobile application subsystem

3.2. **Decomposition Description**

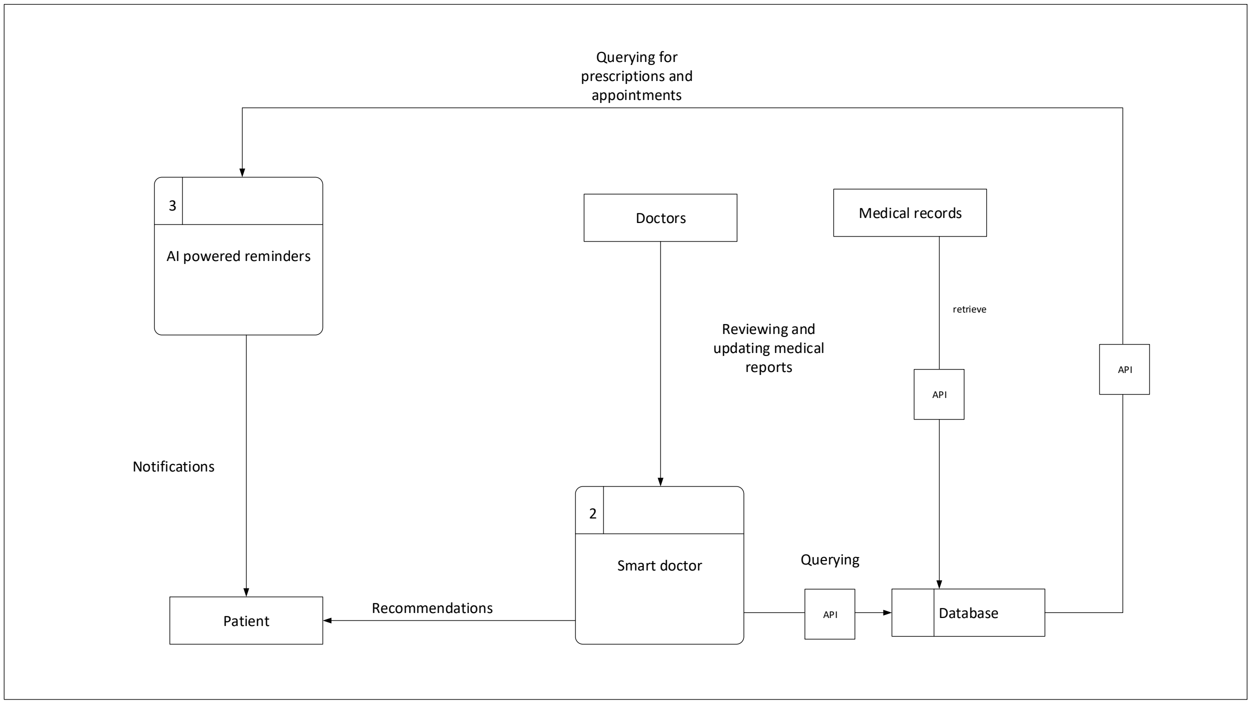


Figure 4. Level 1 Data Flow diagram for a Health API and Mobile Application System.

The level one DFD has two external entities; Patient and health personnel. The Patient and Health personnel entities will be the sources of data that are input into the system, while the Health API entity will be the endpoint for requesting all the processes from and to the database like sending AI powered notifications as per the prescription in the database, accessing medical reports of the patients from the database, mapping doctors for consultation according to the communicated symptoms by the patient and also availing data for recommendations to patients through the chatbot. The Patient entity represents someone indeed of hospital services or health care attention or advice who will interact with the system through the mobile application. The Patient entity will make consultations through video conferencing, get recommendations through communication of the symptoms to the chatbot. The health personnel entity represents doctors and nurses who will interact with the system to achieve specific tasks. The health personnel will provide consultations to the patient, update the patient's medical report and give prescriptions to the patients if there is any. The system itself will contain the various processes and data stores that make up the system.

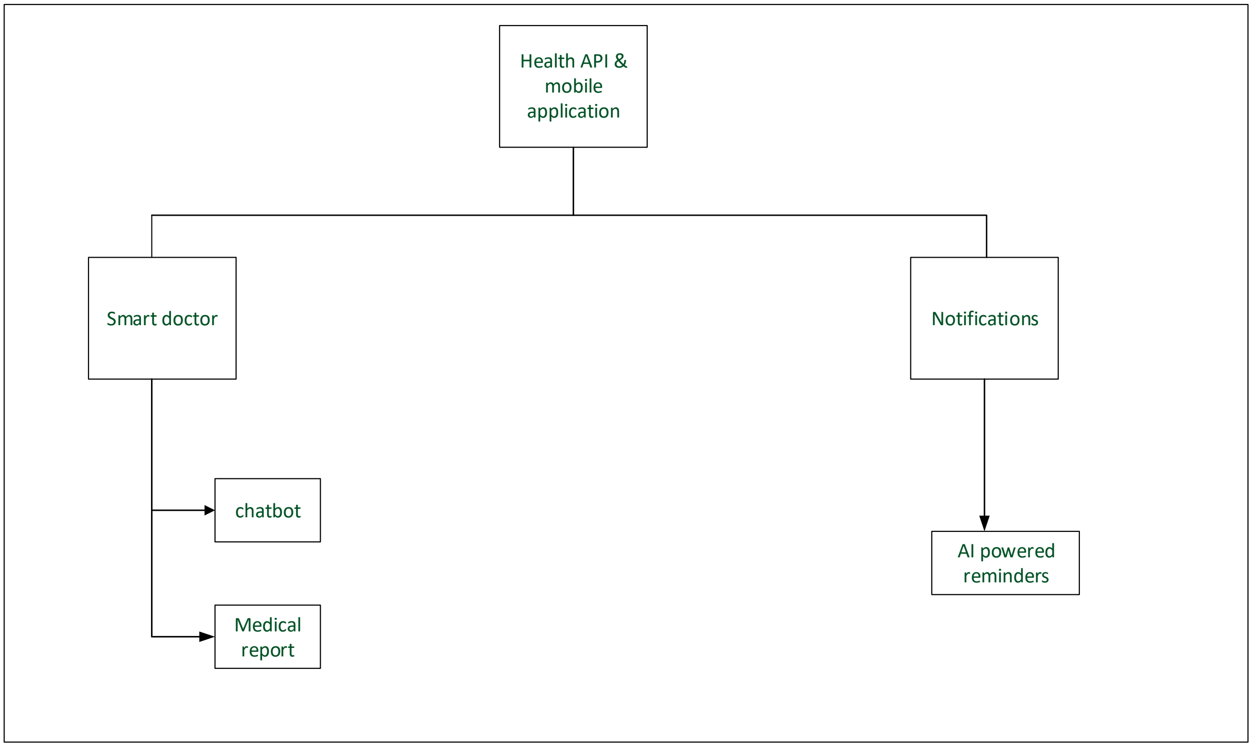


Figure 5. Functional decomposition diagram.

- **User Interface.** This module will provide an interface for end-users to interact with the system. It will include a mobile application. This will enable users to use and interact with the other modules of the system that are mentioned.

- **Doctor Smart**. This will consist of a;
    a) Chatbot. This will provide recommendations with the health of natural language processing to the patients based on their communicated symptoms and incase the symptoms really require you to see a doctor the chatbot would recommend you to go see a doctor in a hospital or it recommends you to go to the consultation submodule in the telemedicine module so as to schedule an online consultation from the doctors or health personnel.
    b) Patient Medical reports. This will enable doctors or a health personnel working on a patient be able to see the patients past medical record to know the kind of patient they are dealing with, and this will all be made possible when the health personnel hit the API end point which queries the database for that patients report.
- **AI powered reminders**. These will be sent to the patient basing on the medical prescription stored in the database and they are activated as soon as medication prescription is given to a patient and the details are entered in the database. These notifications will have self-awareness and if the patient's time for a specific medication is done, they will cancel themselves from being received by that patient.

- **Health API.** This module will provide an interface for other parts of the system to interact with the backend processes. It will include a RESTful API. The health API module enables the different parts of the system to communicate with each other and exchange information to and from the different databases of hospitals.

3.3.Design Rationale

The HEALTH-API and MOBILE APPLICATION system will use a micro service architecture which is a software design pattern where a complex application is broken down into a collection of small, independent, and loosely coupled services. Each service is designed to perform a specific business function and communicate with other services using a well-defined API. The key idea behind micro service architecture is to break down a large monolithic system into smaller, independent components that are easier to manage, scale, and deploy. Each micro service will be developed and deployed independently, allowing for greater agility and flexibility in software development.

In this architecture, each service will typically have its own database or data store and communicate with other services using lightweight protocols such as HTTP or message queues. This allows for each service to have its own data model and business logic which will help to simplify the development and testing of each service.

In this architecture you can also use Apigee to manage the API's connection to the different micro services together providing a centralized point for managing API traffic and enforcing security and access policies.

With the use of Apigee in a micro services architecture, the API gateway will sit between the client and the micro services, intercepting and routing API calls to the appropriate micro services based on the request URL and parameters. The API gateway will also perform other functions, such as authentication and authorization, rate limiting, and request transformation.

Micro service architecture also encourages the use of containerization and orchestration tools such as Docker and Kubernetes to simplify the deployment and scaling of services. With containerization, each micro service is packaged as a lightweight, standalone unit that can be easily deployed and run on any system that supports the containerization technology. Orchestration tools help to automate the deployment and scaling of micro services across multiple servers, making it easier to manage and scale complex applications.

One of the key benefits of micro service architecture is its scalability. Because each service can be scaled independently, it is possible to scale only the services that need more resources, rather than scaling the entire application. This can help to reduce costs and improve performance by optimizing resource utilization.

However, micro service architecture also presents some challenges. One of the main challenges is the increased complexity of managing a large number of services, each with its own data store and business logic.

Additionally, designing and implementing the APIs that allow services to communicate with each other can be challenging, and requires careful attention to the design of the system as a whole.

4. DATA DESIGN

4.1.Data Description

Since the Mobile application will be used to access the services of the API like recommendations, consultation inform of video conferencing, AI powered notifications and patient medical reports parents etc., the users of the mobile application system are expected to create, manipulate, update and delete data. This section therefore gives a description of the data structures of the system.

Table 1. A table of Patient Data Description

Entity	**Description**	**Attributes**

Patient	This entity stores data about the patient.	• Patient_ID • FullName • Email • Password • Contact • Location

Table 2. A table of Medical Report Data Description

Entity	**Description**	**Attributes**
Medical Report	This entity stores data about Medical Reports.	• Medical_ID • MedicalReport

Table 3. A table of Prescription Data Description

Entity	**Description**	**Attributes**

Prescriptions	This entity stores data about the prescriptions.	- Prescription_ID - PrescriptoinDescription

Table 4. A table of Health personnel Data Description

Entity	**Description**	**Attributes**
Health personnel	This entity stores data about the health personnel.	- Personnel_ID - FullName - Email - Password - Contact - Hospital - Description

4.2. **Data Dictionary**

Table.5 A table showing Patient Data Dictionary.

Entity	**Attributes**	**Data Type**	**Field Size**	**Description**
Patient	Id	integer	auto_increment	Unique identifier for each patient in the system
	fullname	varchar	255	This is the name of the patient.

	email	varchar	255	This is the email of the patient.
	password	varchar	255	This is the password created by the patient.
	contact	integer	10	This is the contact of the patient.

Table 6. A table showing Medical Report Data Dictionary.

Entity	**Attributes**	**Data Type**	**Field Size**	**Description**
Medical Report	Id	integer	auto_increment	Unique identifier for each medical report in the system.
	MedicalReport	binary		This is a file containing

				the medical details of a patient.

Table 7. A table showing Prescription Data Dictionary.

Entity	**Attributes**	**Data Type**	**Field Size**	**Description**
Prescriptions	Id	integer	auto_increment	A unique identifier for each prescription is made for a patient in the system.
	PrescriptoinDescription	varchar	25	This is the prescription given to a patient that will trigger the AI Powered reminders.

Table 8. A table showing Health personnel Data Dictionary.

Entity	**Attributes**	**Data Type**	**Field Size**	**Description**
Health personnel	id	integer	auto_increment	A unique identifier for each prescription is made for a patient in the system.
	fullname	varchar	255	This is the name of the health personnel.
	email	varchar	25	This is the email of the health personnel.
	password	varchar	25	This is the password created by

				the health personnel.
	contact	integer	10	This is the contact of the health personnel.
	hospital	varchar	25	This is the hospital where the health personnel are employed.
	description	varchar	255	This is the description of the health personnel and their expertise.

Entity Relationship Diagram

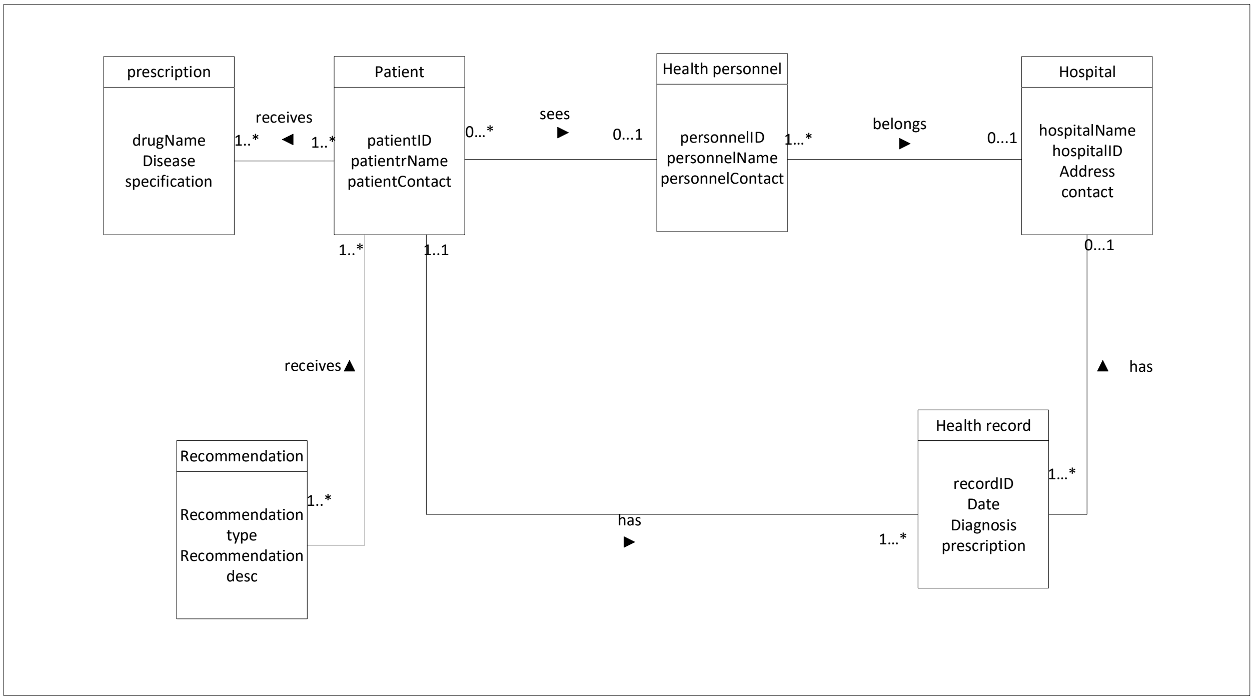


Figure 6 entity relationship diagram

5. COMPONENT DESIGN

This section provides a detailed breakdown of the system's functionality and structure. It also helps to identify and define the individual components our system will have and how they will interact with each other to achieve the system's overall goals and objectives as shown below.

5.1.Chatbot

```
Begin
        Prompt the user for input
        Receive user input
If (the input is a request for a recommendation) then
        Query the health API for relevant information
```

```
    Analyze the results
    Generate recommendations based on the analysis
     Respond with the recommendations
else If (the input is a requisition for a recommendation that requires a health personnel) then
    Query the health API for relevant information
    Analyze the request
    Recommend the user to go to the consultation module


If the input is a farewell:


    Respond with a farewell


    End the chat


Endif
```

5.2.Patient Medical reports

```
Begin
    Health personnel logs into the system
    Health personnel requests a medical report for a specific patient
    System prompts the health personnel for the patient ID
    Health personnel enters the patient’s name
    System sends an API call to the database with the patient’s name.
    Database receives the API call and searches for the patient's medical report
    If (the medical report is found) then
        Database sends the medical report to the system
```

```
            System displays the medical report to the doctor
        else If (the medical report is not found) then
            Database sends an error message to the system
            System displays the error message to the doctor
            A medical report is generated for the patient after the tests
            Health personnel logs out of the system

    End
```

5.3. Data Explorer.

```
    Begin
        Connect to the database containing patient records.
        Retrieve the records of patients tested for malaria, typhoid, and HIV/AIDS in Rubaga, Nakawa, Makindye, and Kawempe regions.
        Preprocess the data by cleaning, normalizing, and transforming it into a machine-readable format.
        Compute the total number of patients tested for each disease in each region.
        Plot a graph showing the total number of patients tested for each disease in each region.

End
```

5.4.AI Powered reminders

```
    Begin

        Query the database for patient prescription information

        For (each patient prescription):
            Check if the prescription is active and valid
            If (the prescription is active and valid) then
                Create a notification message for the patient.
                Send the notification message to the patient inform of a push notification.
            else if (the prescription is done) then
                cancel notification.
    End
```

5.5.Health API

```
Begin
    Define the endpoints for the API:
        /hospital for connecting to hospital databases.
        /dataexplorer for analytics.
        /health recommendations for provision of basic health recommendations.
        /medicalreports for viewing medical report.

    Create authentication mechanisms to ensure secure access to the API endpoints.
    Define input parameters for each endpoint:
        /hospital:
             Hospital ID.
            Query parameters (e.g., patient ID, date range)


        /recommendations:


            User ID.


            Health condition or symptoms.


        /dataexplorer:


            Patient ID
```

disease type.

Define output parameters for each endpoint:

- /hospital:
  - Hospital data (e.g., patient records, appointment information)
- /recommendations:
  - Health recommendations
- /dataexplorer:
  - Graph with details

Define error handling mechanisms for each endpoint

Test the API endpoints to ensure they work as intended

Deploy

6. HUMAN INTERFACE DESIGN

6.1.Overview of User Interface

This section describes the functionality of the system from the user’s perspective. It also explains how the user will be able to use the Health API and Mobile Application system to complete all the expected features and the feedback information that will be displayed for the user.

Welcome Interface. This is the interface that welcomes the user to the application provided he or she has not registered or logged in. The user will have three buttons. i.e., The Register as Patient button, register as a health personnel’s button which will lead them to a registration form so that they can register and the Continue button which will lead the user to another interface to log in.

Registration Interface. This interface has the fields that the user must fill in order to register into the system, here he will provide his number and provide a password which will be used as credentials for logging into the system. Successful completion of this phase leads the user to the login page.

Login Interface. This interface requires the user to provide his number and password recorded during registration to log into the system. When the credentials provided are correct, the user is logged into the system, and they’ll be led to the interface for registering the child and connecting the device. When the user forgets the password, they’ll click a button that will prompt them to recover his password.

Forgotten Password Interface. This interface is used to recover passwords of users who have forgotten their passwords. They’ll be required to enter their numbers and then the password will be sent to them. Upon completion, the user is redirected to the login interface so that they can login with their passwords. If the process is successful, they’ll be logged into the system and prompted to register the child and connect the device.

Doctor Smart Interface. This interface has two interfaces with in itself;

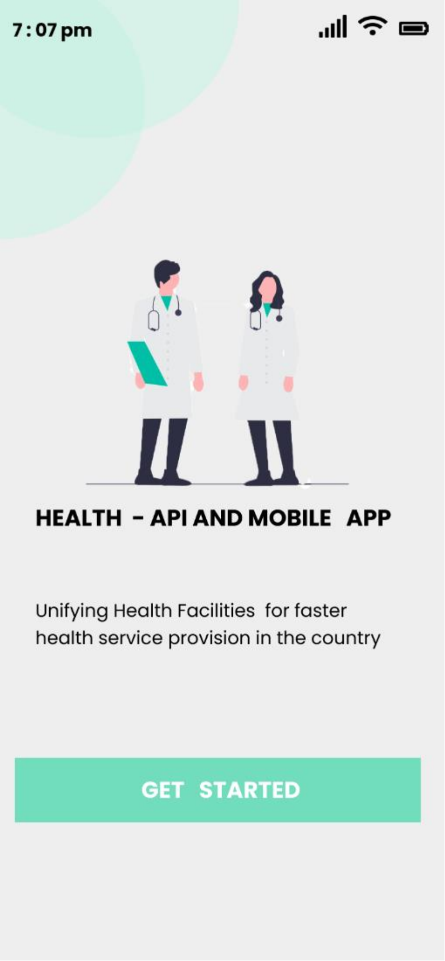
- The interface has a chatbot interface which requires the user to first access the chatbot feature to communicate their symptoms. The chatbot, using natural language processing, will analyze the user's symptoms and provide recommendations based on the severity of the symptoms. If the symptoms require immediate medical attention, the chatbot will recommend the user to visit a hospital, and if the symptoms are less severe, the chatbot will recommend scheduling an online consultation with doctor or health personnel using the telemedicine module.
- The patient medical reports interface will help the doctors or health personnel to access patient past medical records which are stored in the database hitting the API endpoint that queries the database to return this information. This feature will enable the doctors or health personnel working on the user to have a better understanding of the patient's medical history.

Telemedicine Interface: This interface enables the user to schedule an online consultation with doctor or health personnel through a video conference. To use this interface, the user will communicate their signs and symptoms to the chat option in this module. Based on the symptoms communicated, the system will analyze and match the user with doctor or health personnel who is available and has knowledge of the communicated symptoms.

Notification Interface: The user will be able to turn off or remove the reminder using this interface

6.2.Screen Images

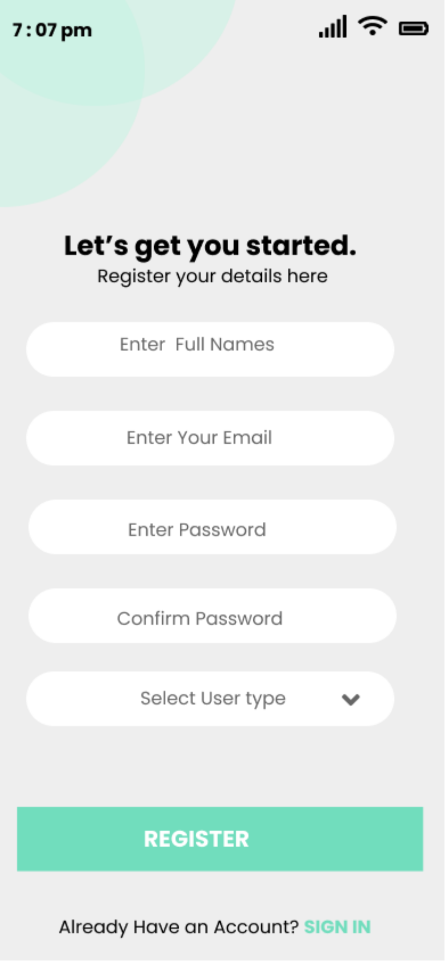
Display screenshots showing the interface from the user’s perspective. These can be hand-drawn or you can use an automated drawing tool. Just make them as accurate as possible. (Graph paper works well.)



HEALTH - API AND MOBILE APP

Unifying Health Facilities for faster health service provision in the country

Figure 7. Welcome Interface



Let’s get you started.

Register your details here

Already Have an Account? SIGN IN

Figure 8, Registration Interface

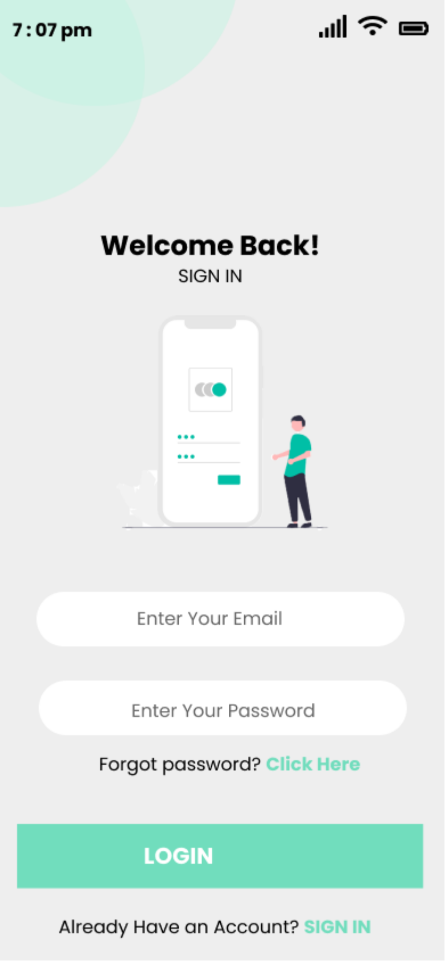


Figure 9.Login Interface.

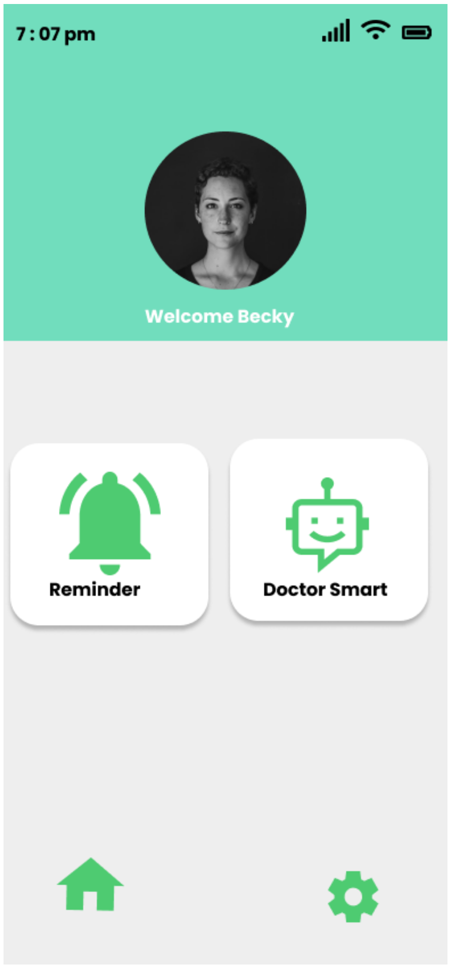
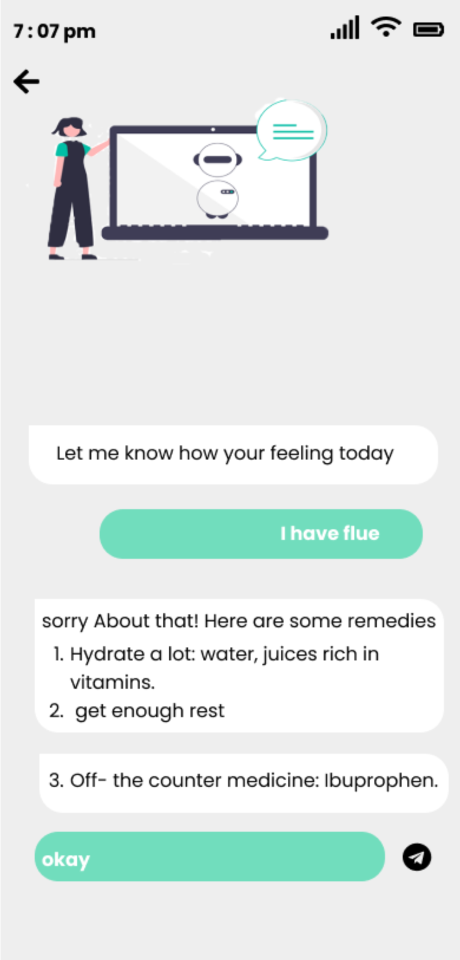


Figure 10. Home screen



Let me know how your feeling today

I have flue

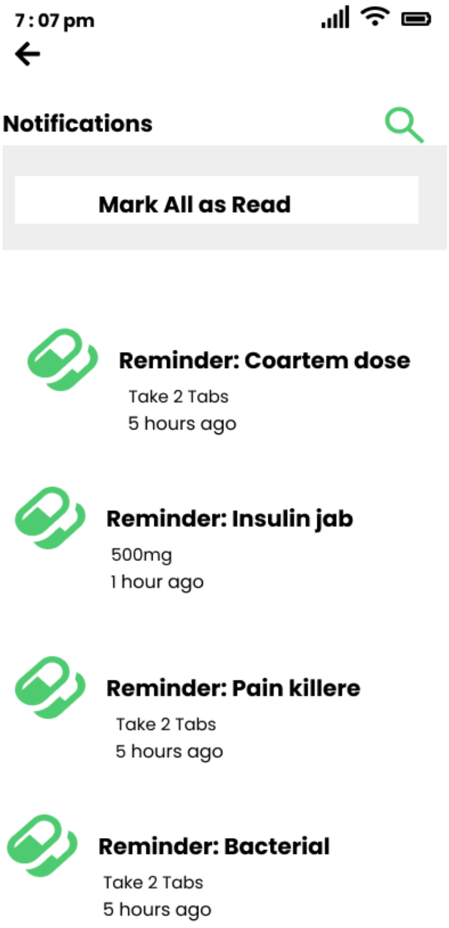
sorry About that! Here are some remedies

1. Hydrate a lot: water, juices rich in vitamins.
2. get enough rest

3. Off- the counter medicine: Ibuprophen.

okay

Figure 11. Chatbot screen



Reminder: Coartem dose
Take 2 Tabs
5 hours ago

Reminder: Insulin jab
500mg
1 hour ago

Reminder: Pain killere
Take 2 Tabs
5 hours ago

Reminder: Bacterial
Take 2 Tabs
5 hours ago

Figure 12.AI Powered notifications screen

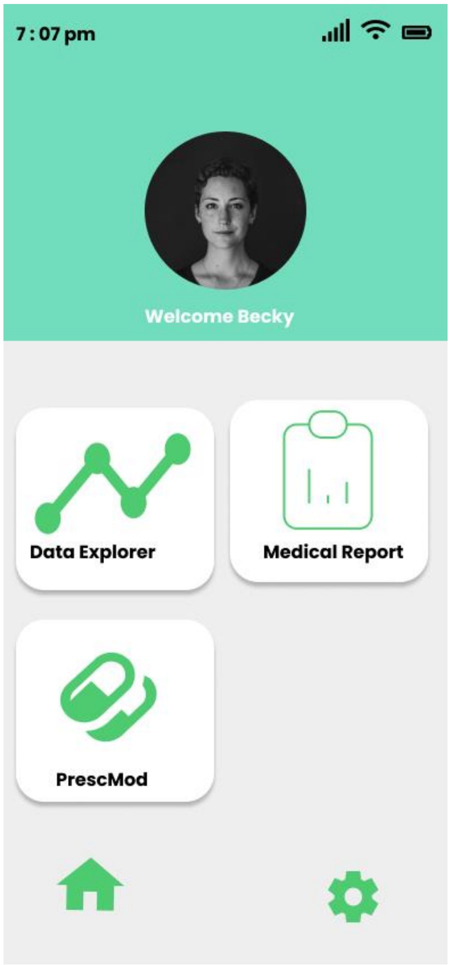


Figure 13.Doctors home screen

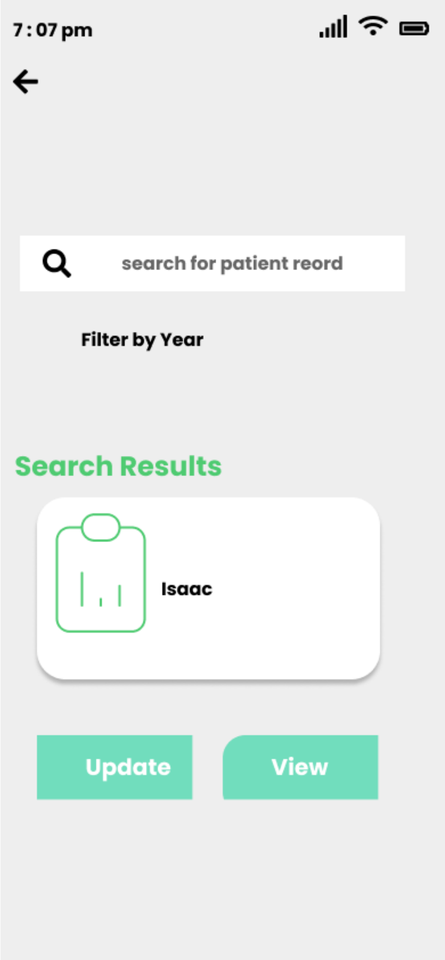


Figure 14. Medical reports screen

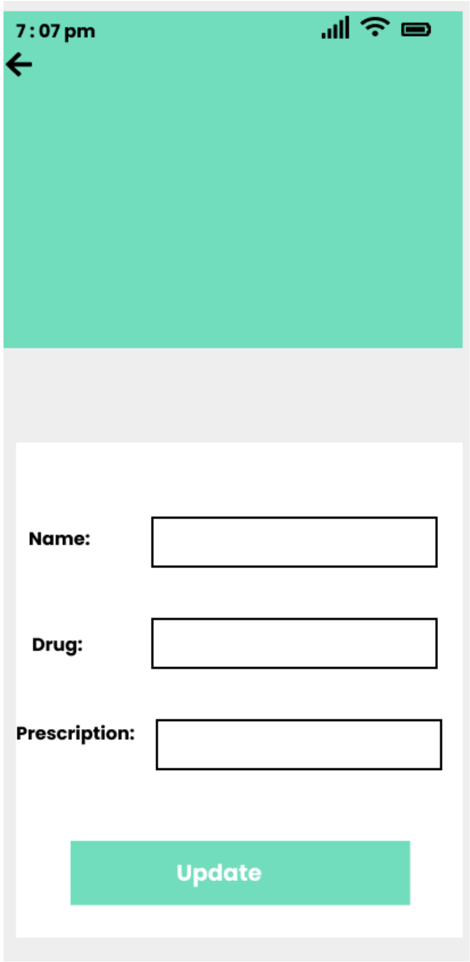
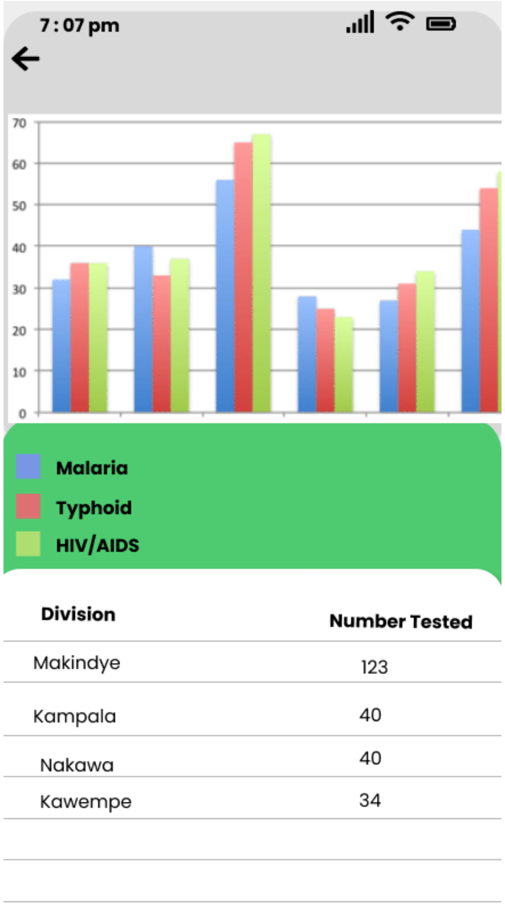


Figure 15.Entering prescription screen



Division	Number Tested
Makindye	123
Kampala	40
Nakawa	40
Kawempe	34

Figure 16.Data explorer screen

6.3.Screen Objects and Actions

Table 9 objects and actions

Identifier	Object	Name	Action
	REGISTER	Register Button	On clicking the register button, the users' details will be stored in the respective database tables and an account created for them
	LOGIN	Login Button	By clicking the login button, users will be taken to either the patient welcome page or the doctor's welcome page. If a user is not registered, they will be referred to the login page
	Update	Update Report button	By clicking the update report button, any changes made by the doctor to a patient's report will be effected and updated in the database tables
	View	View report button	By clicking the view report Button, a more detailed view of the report will be seen
	▼	Filter button	By clicking the Filter Button, the search results from data records fetched by the API will be narrowed to a particular year.
	⌂	Home Button	By clicking the Home Button, a user will be taken back to the home page of that user view I.e., patient home or doctor home.
	🤖	Chatbot	On clicking the Chatbot button, the patient will be taken to the chatbot page.
	🔔	Notification Button	On clicking the notification button, the patient will be taken to the notification page.
	←	Back Button	On clicking the back button, the user will be referred to the previous page

	⚙	Settings	On clicking the settings button, the user will be able to log out of the active account
	➤	**Send Button**	On clicking the send button, a message will be sent, thereafter the user will await a response from the chatbot.
	🔍	**Search Button**	On clicking the search Button, the application with the help of the health-Api will return records (from Health organizations that leverage this API) matching the user inputs. If they exist.
	📈	**Data Explorer button**	On clicking the Data Explorer button, the doctor will be taken to a disease analysis based on tested cases page.
	💊	**PrescMod button**	On clicking the PrescMod button, a doctor will be taken to the prescription page
	📋	**Medical Reports Button**	On clicking the PrescMod button, a doctor will be taken to the medical reports page.

7. REQUIREMENTS MATRIX

REQ1: The system shall provide recommendations to patients through their communicated symptoms.

REQ2: The system shall recommend patients to visit a hospital and see health personnel in case the communicated symptoms require serious medical attention.

REQ3: The system shall allow health personnels to predict an out break of a disease based on the analytics.

REQ4: The system shall provide reminders to patients basing on their prescription inform of notifications

REQ5: The system shall cancel the reminders sent to patients in case their prescription is completed.

REQ6: The system shall remind the patient about the checkups he or she should have at the hospital.

REQ7: The system shall allow health personnel to view and update patient medical reports

REQ8: The system shall provide an API that will provide all the functionalities of the system for integration with other systems.

Table 10 Traceability Matrix for a health API and Mobile Application System

REQUIREMENTS CODE	SYSTEM COMPONENT	DATA STRUCTURE
REQ1	Recommendations module	Text
REQ2	Recommendations module	Text
REQ3	Data explorer module	visual data
REQ4	Notification module	Alert message
REQ5	Notification modification module	Notification schedule
REQ6	Notification module	Reminder
REQ7	Report module	Database

8. APPENDICES

Link to srs: https://drive.google.com/file/d/1L7DgcXeqFVeMyKB1gOmNABky7IEgpmWg/view?usp=sharing

Link to project blog: https://hillaryemokol.github.io/sample_website