Software Requirements Specification For

HEALTH API AND MOBILE APPLICATION

Version 1.0 approved

Prepared by:

BSE23-9

Prepared For:

Health API and mobile application

DECEMBER 2022

Table of Contents

Table of Figures

1. Introduction

1.1 Purpose

The purpose of this document is to detail the software requirements of the HEALTH-API and MOBILE APPLICATION 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

1.2 Document Conventions

Font – IBM Plex Sans

- Product Mindset Pillars
- Approach for development – Outside-in approach
- Time to market – Minimum Viable Product
- Iteration – Synchronize the API lifecycle with a modern agile iterative Software Development Lifecycle.
- Automate API testing and deployment – Using Google Apigee.

1.3 Intended Audience and Reading Suggestions

This document is intended to serve as;

1. **Central Communication Point:** A central communication point between the API and application development team, project coordinator and overall supervisor.
2. **Final description of software features:** This document unifies all requirements into one standard reference, which the intended system should provide for.
3. **Standard guide for product developers/implementers:** This document guides the developers on the steps they should follow in order to precisely build an API that is application developer consumable together with an application that demonstrates the intended capabilities of the API.

This document will help guide the API and Application team to design and develop the API and Application using the quoted requirements as the basis, as well as to plan and manage all project resources. The User Acceptance Testing session will refer to this document in order to generate test data to effectively test the intended API and Application Product.

1.4 Product Scope

The product targets all health institutions in the country. It aims at integrating databases of different health institutions across the country through its API.
The product also targets developers that will be able subscribe to the API once fully developed by means of an API-key.

1.4.1. Description

API

Application Programming Interfaces are a technology that connects systems. More interfaces enable developers to repeatedly leverage data, functions and applications to build new products and services.

In a business context, they are how a business expresses itself using software and they enable that business to rapidly expand into new contexts or adapt to meet changing user needs and preferences.

This API product intends to leverage medical data, different system functions and the Ugandan health eco system to provide deeper customer experiences and mechanisms through which value is increasingly exchanged in modern economies.

Mobile Application

Mobile application is a software application, which is designed to be used on a mobile smart device. It will help individuals track and manage their health and wellness through;

- *Provision of highlights that will provide them with information on how to maintain their wellness. This will also provide the prediction of patient needs passing on their patient history and communicated symptoms.*
- *Smart Doctor. Which will consist of a Chabot that would provide recommendations to the patients based on their communicated symptoms plus a medical report, which will provide medical reports for patients from different hospitals so that the doctor can know the history of the patient.*
- *Telemedicine which will have functionalities like consultation which will involve real-time video conferencing with a doctor, ordering for a drug where a patient will be able to get drugs from any hospital of choice and also ordering pick up where an individual would be able to order pickup for a patient or pregnant lady from any hospital of choice.*
- *AI powered reminders. This will help patients keep up with their medications by reminding them through push notifications or alarms.*

1.4.2. Benefits

APIs make digital society and digital business work by connecting people, business and things. These connections enable new digital products and business models and create new business channels.

1.4.3. Objectives & Goals

Designing and delivering API products for long-term value at scale.
Evolving the product over time to meet the changing customer needs.
Providing an easy-to-use API product seeking to increase the likelihood the API can continue to provide strategic value and extensibility in the future.

1.5 References

This SRS refers to a research article “OF WHAT RELEVANCE ARE APIS TO THE HEALTH SECTOR IN UGANDA” by Joshua G. Karamuzi, Hillary Emokol, Emmanuel Noel Rutahigwa and Isaac Ngabirano published in the month of June 2022.

1. Author, F.: Article title. Journal 2(5), 99–110 (2016).

2. Author, F., Author, S.: Title of a proceedings paper. In: Editor, F., Editor, S. (eds.) CONFERENCE 2016, LNCS, vol. 9999, pp. 1–13. Springer, Heidelberg (2016).
3. Author, F., Author, S., Author, T.: Book title. 2nd edn. Publisher, Location (1999).
4. Author, F.: Contribution title. In: 9th International Proceedings on Proceedings, pp. 1–2. Publisher, Location (2010).
5. LNCS Homepage, http://www.springer.com/lncs, last accessed 2016/11/21.

2. Overall Description

2.1 Product Perspective

2.1.1. Context-
The product targets every patient that acquires a medical service from a hospital within the country. The product is being developed for the health sector with the aim of improving service provision in the country.

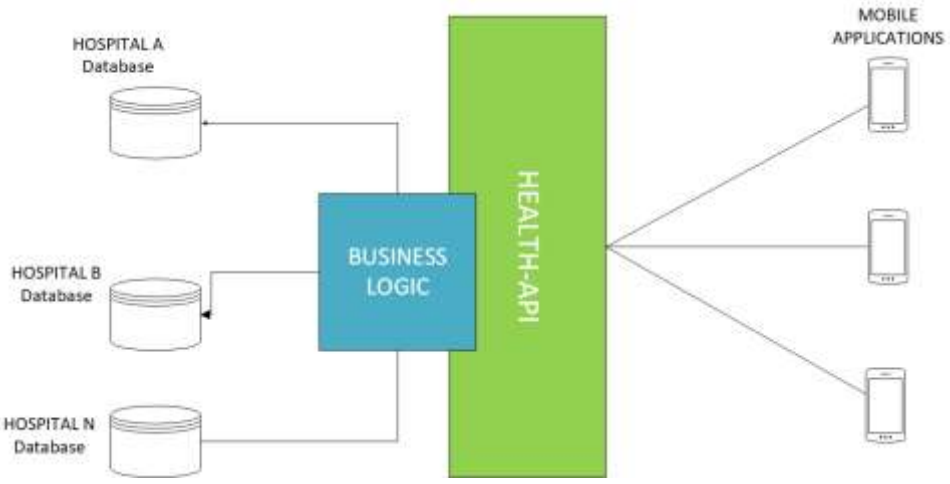
2.1.2. Origin- The idea was conceived during the internship exercise and each of the group members had an opportunity to be attached to an organization relevant to the field of study.
One of the group members had the opportunity to intern at Case Hospital that later turned out to be a full employment opportunity and during this period a system for the hospital was in the pipeline for development. He then had the idea that the health sector could have a similar system to the current mobile money API across the country to supplement health management information systems in hospitals.
Research was then carried out by the entire group during the Research Methodology course. The topic researched on was “health sector and application programming interfaces” the output of which was a concept paper that we have chosen to carry forward to implement, thus giving birth to this project.

The idea is having an API leveraging multiple records of data from various hospital databases and making them available to developers in the health sector.

2.1.3. Interfaces -
The product is set to run on a cloud platform specifically google cloud platform
The Application-programming interface itself through the business logic will merge various records from hospital databases to be leveraged by various front-end applications

2.1.4. Components-
The health-Api and mobile application is based on two major components:
-the Application programming interface with relevant business logic
-the mobile and desktop application
It is important to note that a number of applications can be developed to support the application-programming interface once complete.

Figure 1.



2.2 Product Functions

- Help patients follow up with their medications through reminders and mapping of patient and medical information. (Predict patient needs based on past medical trends.)
- Chat bots/ Health Assistant (providing AI driven medical recommendations based on patient history and communication of symptoms, health tips and medical management)
- Telemedicine

2.3 User Classes and Characteristics

There are mainly three user classes for this product:

- **Developers:**

These are individuals or organizations in the field of Information technology that wish to leverage the application programming interface once complete to develop other health related systems

- **Patients:**

- These are customers to various hospitals, health centers that use the mobile or desktop application to acquire medication.

- **Hospitals / health personnel:**

-hospitals are the most relevant users of this product as it is entirely developed and designed for this setting.

2.4 Operating Environment

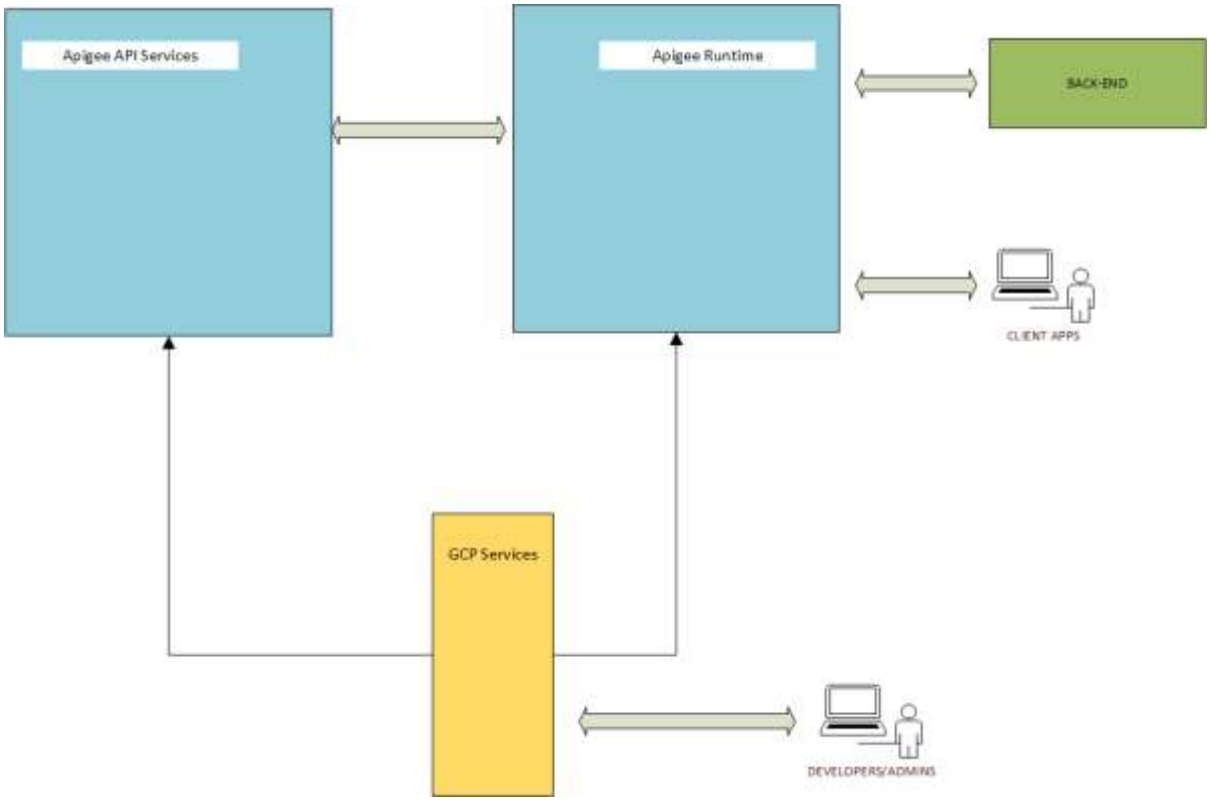
API (application programming interface) operating environment:
The API will be developed on google cloud platform and will be developed using apigee.

Apigee is a tool for developing Restful application programming interfaces (API'S) over the google cloud platform.
The tool can support both development and management of application programming interfaces.
The tool leverages API proxies. We then provide http endpoints that protect our backend services from developers and users of the application.

To the developer; once the health-Api and mobile application is developed, the developers will subscribe to the product and an API key will be provided to enable access to the proxy and backend services.

Components of apigee

Figure 2.



Apigee API services: creating, managing and deploying API proxies.

Apigee runtime: it handles and maintains all API traffic through Kubernetes containers

Back-end services:
GCP Services: provide identity management, cloud monitoring and project management functions.

Application operating environment:
The mobile applications are set to run over android operating system and IOS. This is because development will be done in flutter.
The desktop applications will run over the windows operating system.
Applications will then query

Co-existence:
The flutter application will co-exist with apigee sheets, which is the front-end management tool for apigee.

2.5 Design and Implementation Constraints

ISSUES:
Regulatory policies:
Prevailing policies governing patient information and access to patient data by external entities or organizations are a limiting factor. This implies that the business logic designed by the team will be dictated by organizational policy.

Timing requirements: information will have to be fetched and supplied in a timely manner. This poses a challenge of how much traffic the database servers at various organizations can withstand. The number of results returned at the front-end will have to be filtered. This will improve the speed and response time of the system.

Security: the information we are dealing with is sensitive data. Therefore, security measures will be put in place to deal with prying eyes

2.6 User Documentation

Apigee provides documentation regarding the product- this will be provided on delivery of the product.
Application user manual

2.7 Assumptions and Dependencies

Assumptions:

We assume the google cloud platform's availability is rated at 100%
We assume that a service level agreement, non-disclosure agreement can be signed between the developers and relevant stakeholders regarding access to data.

Dependencies:
The project is dependent on the services and development tools provided by the google cloud platform i.e., apigee and apigee sheets.

3. External Interface Requirements

3.1 User Interfaces

Figure 3. Login Page and Registration Page.

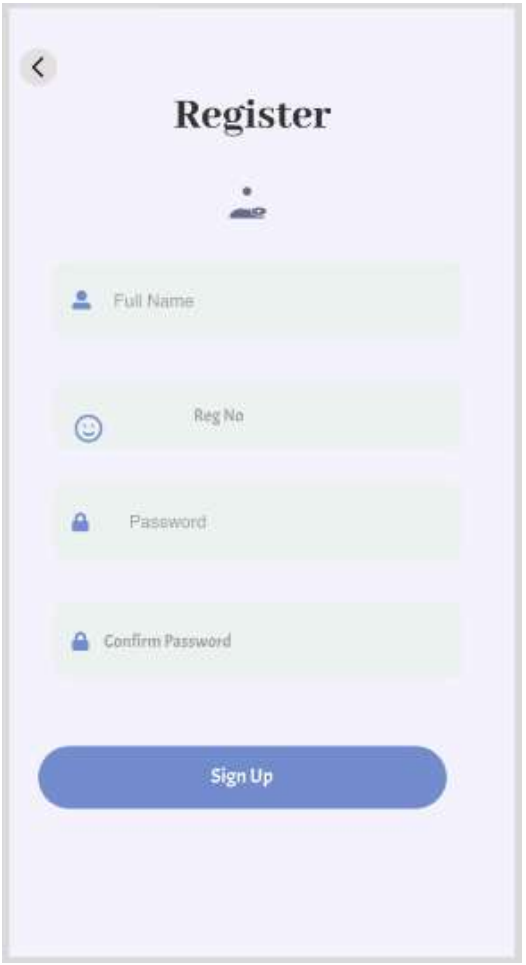


Figure 5. Doctor Smart Screen.

Figure 4. Home Page.

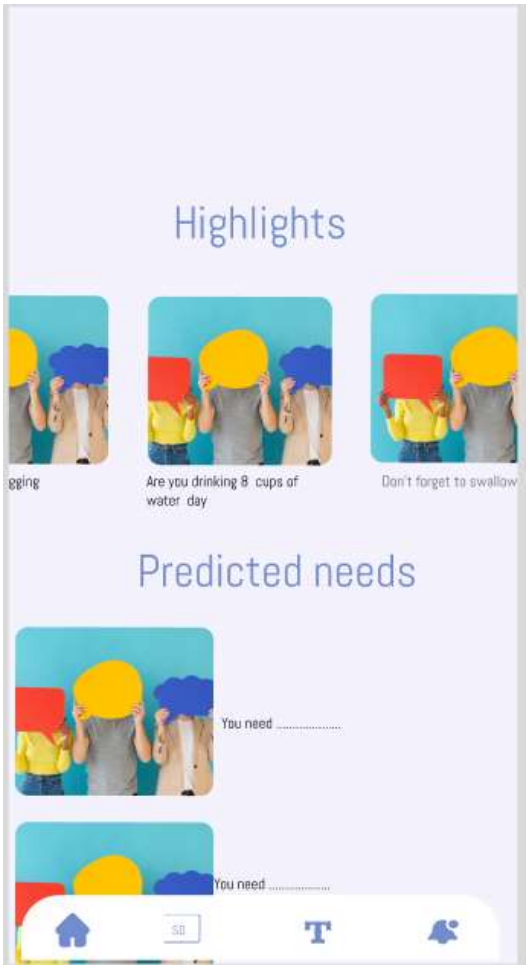


Figure 6. Chabot Screen.

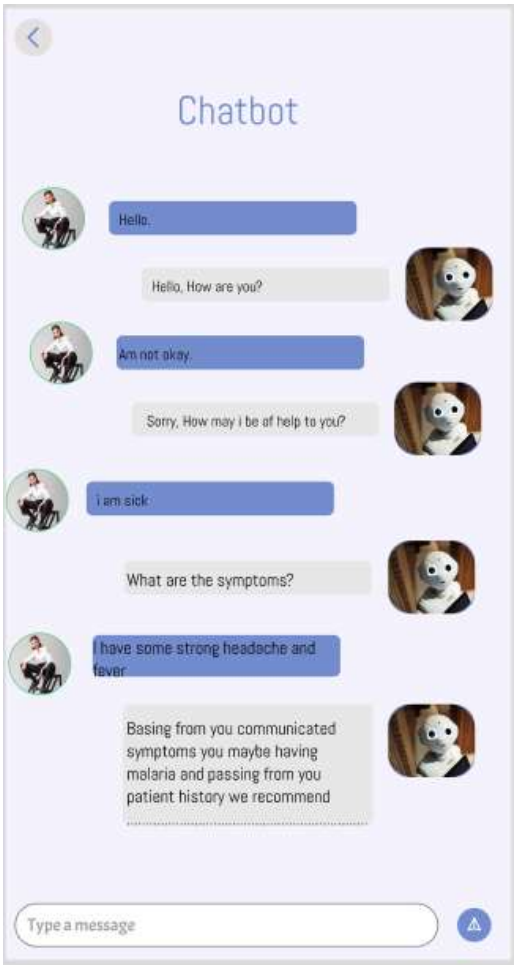


Figure 7. Medical Reports Screen.

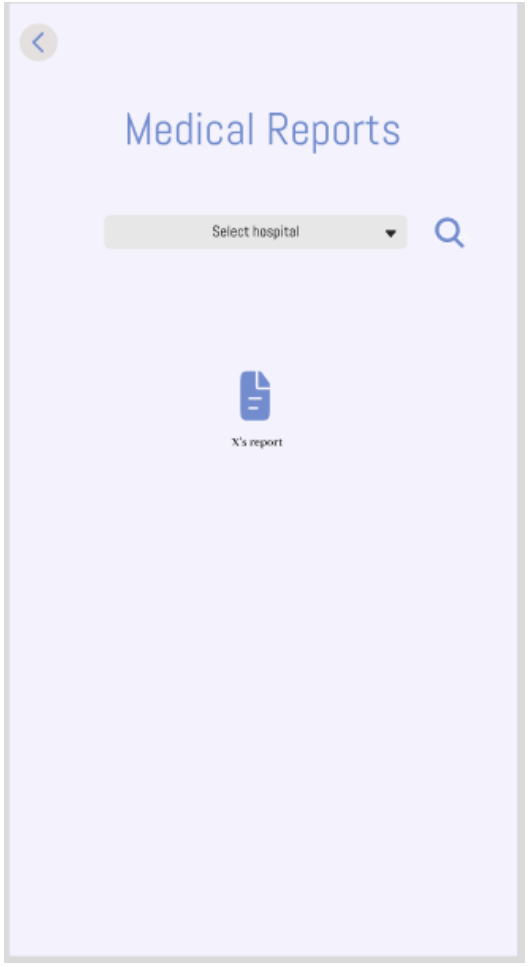


Figure 8. Telemedicine Screen.

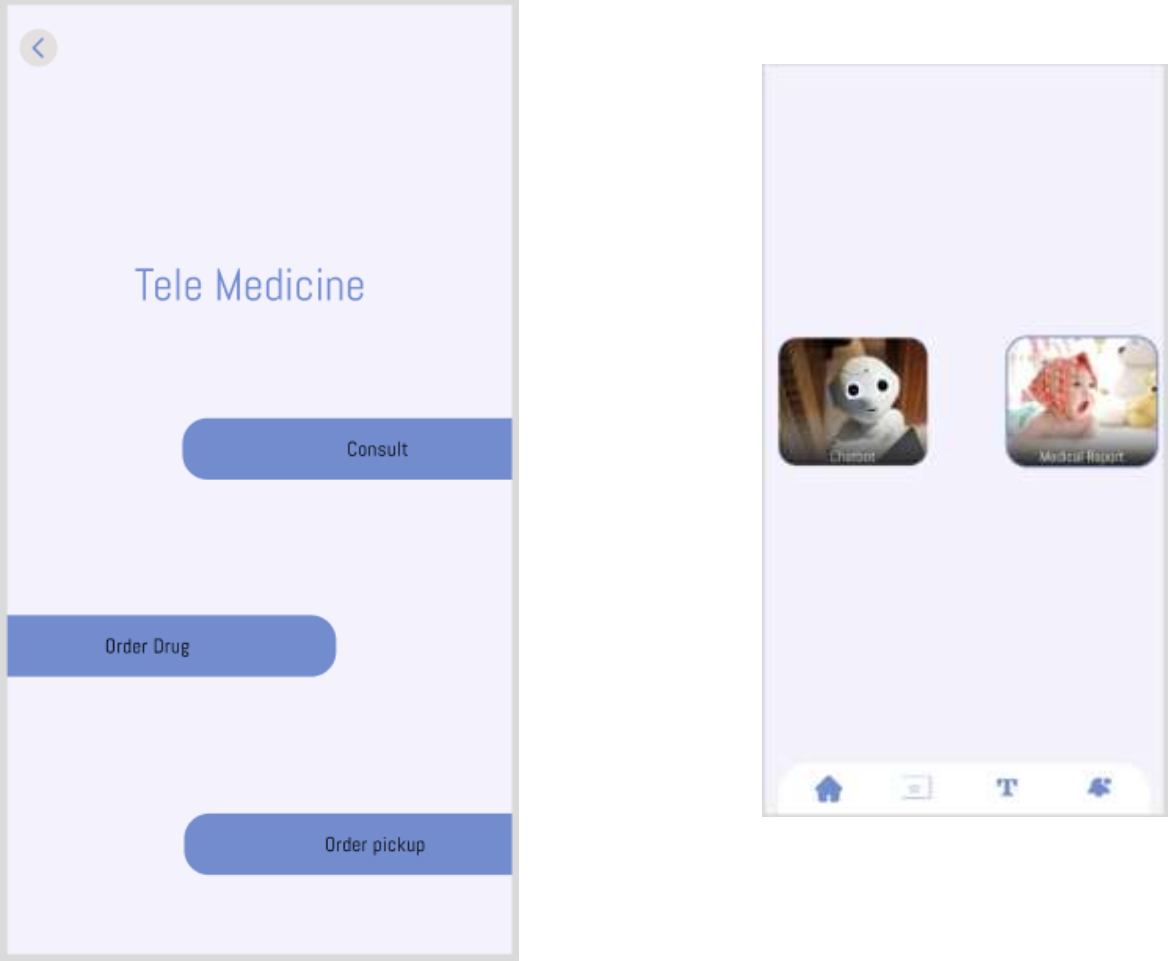
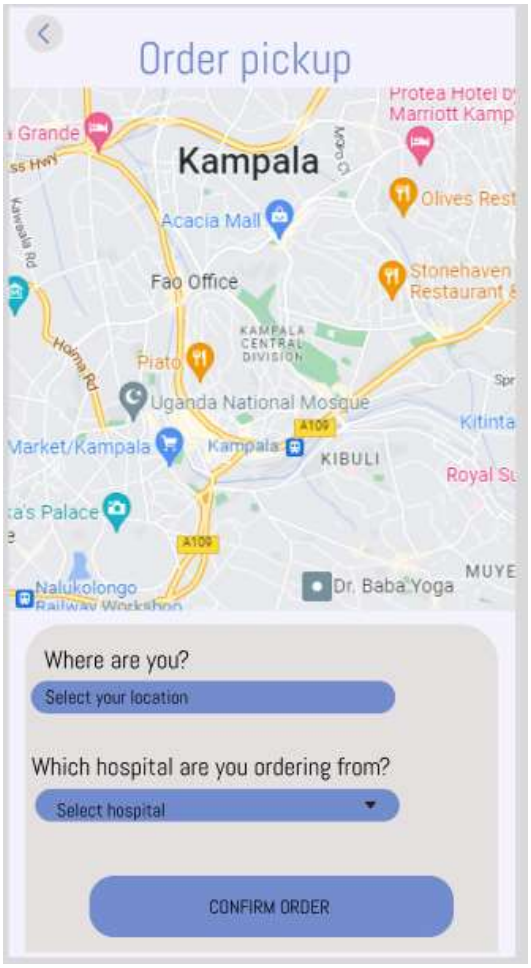


Figure 9. Order Screens.



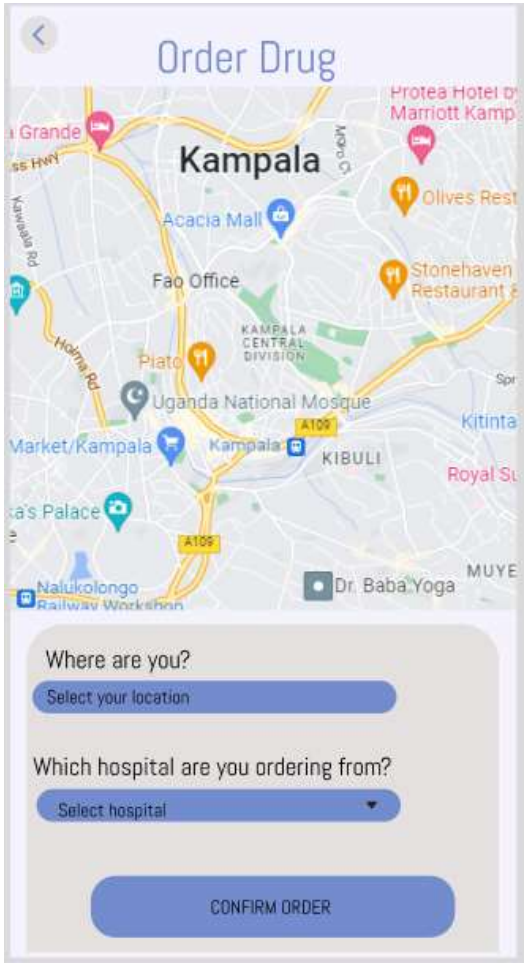


Figure 10. Consultation Screen.

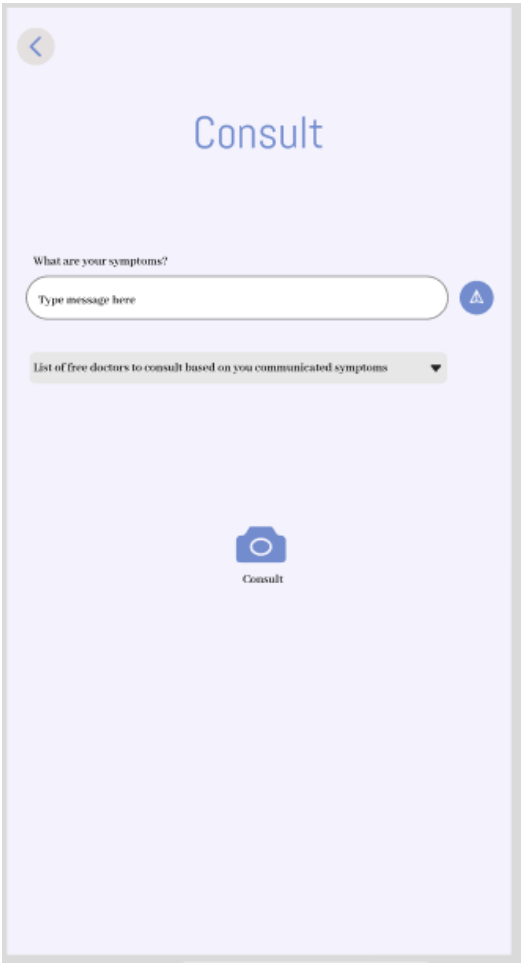
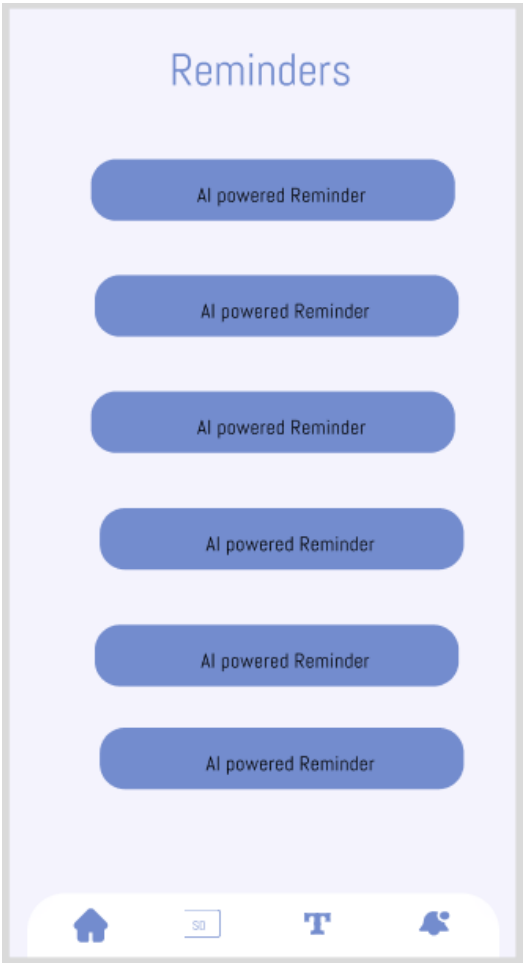


Figure 11. AI powered Reminders.



3.2 Hardware Interfaces

The system is going to be based on an API architecture to communicate between different databases of hospitals and the mobile application is to be hosted to google cloud and will be managed using Apigee sheet.
In order to access functionalities like reminders, mobile doctor and telemedicine, one should also have a smartphone, or a tablet connected to the internet.

Any smartphone or tablet irrespective of the operating system i.e., android or iOS can access the Health API and Mobile Application system as long as it's connected to the internet. The smartphone will have a processor of at least 4GB RAM that is fast enough to handle the tasks the application will perform smoothly and a clock speed of at least 1.5 GHz. The applications will display on smartphones with displays that range from 360*640 to 414*896.

The camera for the smartphone should be at least 4 pixels of camera resolution to support functionalities of telemedicine. Smart phones should have connectivity options such as Wi-Fi to connect to the internet or mobile data connection to enable the use of the application. The tablets will need a processor of at least 4GB RAM that is fast enough to handle the tasks the application will perform smoothly and a clock speed of at least 1.5 GHz. The applications will display on tablets with displays that range from 601*962 to 1280*800.

The camera for the tablets should have at least 4 pixels of camera resolution to support functionalities of telemedicine. The tablets should have connectivity options such as Wi-Fi to connect to the internet and mobile data connection where one of them would be used to connect to the internet to enable use of the application.

The smartphones or tablets shall be running android version 8 and above or iOS version 13 and above. The user will only be required to touch or point to an element on the screen for navigation throughout the application.

The output device is the screen of the smart phone or tablet.

3.3 Software Interfaces

3.3.1. Describing the connections between the Health API and Mobile Application and other specific software components (name and version), including databases, operating systems, tools, libraries, and integrated commercial components.

Software interfaces, also called programming interfaces, are the languages that various software applications use to communicate with each other and with a hardware's internal system.

Table 1. Software Interfaces

Name	Mnemonic	Version number	Source
Google Cloud	Apigee	1.8.3	Google
GSMA	Mobile money API	1.1	GSMA

Operating systems

These are the software applications that manage computing devices' hardware and coordinate their resources, giving a program the right tools to function properly.

The server side will use the Linux operating system while the mobile application can use any operating systems like Android or iOS.

The GSMA

GSMA Mobile Money API is an initiative developed through collaboration between the mobile money industry and the GSMA. It was created to support the mobile money industry and speak the same technical language by providing a modern harmonized API for mobile money transactions and management that is both easy to use and secure.

The payments can also be done via Mobile money by using the GSMA API

3.3.2. Identifying the data items or messages coming into the system and going out and describe the purpose of each.

API header. Contained within sending and receiving of messages to make sure that communication is secure

Data items that are coming into the system include;

- Text. It helps the end user to interact with the chatbot within the system.

It helps the system to choose a doctor who is free in case a patient wants to consult a specific doctor working on her issue.

3.3.2.1. User input.

This will include data entered by the user through the application’s user interface such as;

1. Text input which will help a user(patient) to;
    - It helps the end user to interact with the Chabot within the system.
    - It helps the system to choose a doctor who is free in case a patient wants to consult a specific doctor working on her issue.
2. Selections from the dropdown menu.
    - This will help doctors to select medical reports for the patient they are working on.
    - It will help the patient to select a free doctor for consultation.
    - It will help the user select a specific hospital in case they are making an order for a drug and order for pickup.

3.3.2.2. System data

This will include data generated by the operating system or hardware of the devices such as;

1. GPS location.
- This will help users select the location they are in current so that deliveries can be made to them.

3.3.2.3. Network data

This will include data that will be transmitted over the network connection such as;

1. API requests and responses.
- This will enable the doctor to get different medical reports about their patients from other hospitals.

3.3.2.4. Persistent data.

This will include data that will be in the remote database of the hospital and will be used to provide information to the doctors. For example patient medical reports.

API requests. Will help the doctor to make requests to databases of hospitals when searching for a patient's medical report.

API responses. These are responses to the requests made.

3.3.3. Describing the services needed and the nature of communications

3.3.3.1. Telemedicine.

- This will help patients order for drugs to use in case of any shortages they incur and the nature of communications will be the patient (customer) places an order for a drug from a selected hospital, enter their payment and delivery information and then submit the order through the mobile application which will make an API call to the selected hospital system alerting them that there is an order for drugs confirmed for them.
- It will also help patients to make consultations through real time video conferencing and this will allow healthcare providers and patients to see and hear each other in real time through video connection.
- It will also help patients order for a pickup in case of emergencies like a lady who is about to give birth and the nature of communications will be, the patient (customer) orders for a pickup from a selected hospital, enter their payment and delivery information and then

submit the order through the mobile application which will make API call to the selected hospital system alerting them that there is a pick up order confirmed for them.

3.3.3.2. Doctor smart

This will have two functionalities, which are;

- Chabot

These will give patients recommendations according to the communicated symptoms through a chat interface.

- Medical reports

These enable the doctor to view the patient medical reports from another hospital. This is through API calls to any hospitals database in order to get the medical report.

3.3.3.3. Notifications

- These will be AI powered notifications, which will feed on the information about a patient concerning their prescription in order to provide reminders to the patient though push notifications.

3.3.4. Describing the detailed application programming interface protocols

● HTTP that is a widely used protocol for transferring data over the internet.

Our API with a remote database for any hospital will use HTTP to send and receive requests and responses between the two systems.

In addition, our API with the mobile application will use HTTP to send and receive requests and responses between each other.

● REST that is an architectural style for building APIs that is based on the use of HTTP.

Our API with a remote database for any hospital will use REST to define the structure and behavior of the API, including the types of requests that can be made and the format of the data being exchanged.

Also, Our API with the mobile application will use REST to define the structure and behavior of the API, including the types of requests that can be made and the format of the data being exchanged.

● SQL which is a programming language for managing data stored in relational databases. Our API with a remote database for any hospital will use SQL to query and manipulate the data in the database.

● OAuth which is an open standard for authorization that allows a user to grant access to their data on one system (e.g., a database) to another system (e.g., an API) without providing their credentials. Our API with a remote database for any hospital will use OAuth to authenticate and authorize requests to access the database.

● JSON which is a lightweight data interchange format that is often used to transmit data between a server and a mobile application. Our API with the mobile application will use JSON to encode and decode the data being exchanged between each other.

3.3.5. Identifying the data that will be shared across software components.

● In consultation, the data that will shared may include personal information about the patient or client (e.g., name, age, medical history), information about the specific issue being discussed (e.g., symptoms, diagnostic test results), and recommendations or treatment plans provided by the healthcare provider or professional.

●In the chatbot, the data that will be shared will include information provided by the user like the symptoms and information provided by the chatbot for example recommendations based on the communicated symptoms. The specific data shared will depend on the purpose of the chat and the information designed for response.

● In the ordering processes, the data that will be shared will include information about the products or services being ordered (e.g., name, price, quantity), payment information (e.g., cash payment or mobile money payment), and delivery information (e.g., address). The specific data shared will depend on the requirements of the ordering process and the information needed to complete the order.

3.4 Communications Interfaces

3.4.1. Describing the requirements associated with any communications functions required by the Health API and Mobile Application system.

- The API and mobile application will support the use of email for account registration, it will be compatible with the protocols and standards used for email transmission (e.g., SMTP, IMAP, POP).
- The API and mobile application will use a network server for communication, they will be compatible with the protocols and standards used e.g., TCP and UDP. They will also support features such as encryption, authentication, and load balancing to ensure the security and reliability of the communication.

- The API and mobile application will use electronic forms for reading medical reports, they will be compatible with the data formats and standards used for electronic forms e.g., PDF, XML. They will also support features such as submission and storage.

3.4.2. Defining any pertinent message formatting.

- Text: This messaging format will be supported by the as the user is interacting with the chatbot, which can include a variety of formatting options such as bold, italic, and strikethrough.
- Data formatting: The API and mobile application will format data to structure the data being exchanged from on hospitals database to mobile applications and vice versa. For example, the API may send data in a JSON format, while the mobile application may receive and process data in a different format such as XML or CSV. The message formatting will be compatible with both systems to enable the exchange of data.

3.4.3. Identifying any communication standards that will be used.

- HTTP is a widely used standard for transferring data over the internet. Our API will use HTTP to send and receive requests and responses from different databases.
- HTTPS is an extension of HTTP that uses encryption to secure the data being transmitted between the API and mobile application.
- FTP which is a standard for transferring files over the internet. Our API and mobile application will use FTP to exchange patient medical reports from the database of a hospital to the application where the request has been made.
- TCP which is a standard for transmitting data over the internet. An API and mobile application may use TCP to establish a connection and transmit data between the two systems.

3.4.4. Specifying any communication security or encryption issues, data transfer rates, and synchronization mechanisms.

- The API and mobile application will ensure that security and privacy of the data being exchanged between each other and hospital databases. This may involve the use of encryption protocols such as SSL/TLS to secure communication, or the implementation of authentication and authorization mechanisms to prevent unauthorized access to the data.
- Data transfer rate refers to the speed at which data is transmitted between the API, mobile application and the different hospital databases.

- The API, mobile application and different hospital databases will need to synchronize data or other information among each other. This will involve the use of synchronization protocols or mechanisms to ensure that the data is consistent and up to date across both systems.

4. System Features

4.1 System Features for API

API features
Features provided by the product
Secure connection to different Health Institution Databases.
Secure sharing of information between different Health Institutions

4.1.1 Description and Priority

The Api provides the following functionalities:

- **Auto-documentation**- It provides a documentation for the API to the developers
- **Monetization**- Every developer that wishes to use the Api will subscribe for every number of requests made through it
- **Auto scalability**- With increased traffic access to the API, through Apigee AI – the services provision is scaled based on demand. This helps to reduce costs and also maintain service uptime in instances of increased traffic.
- **Auto security**- Secure connection to different Health Institution Databases.

Secure sharing of information between different Health Institutions.

- **Integrated Testing Environment -** Provides a testing environment for developers to try out the API before a deep dive into development. For example: the nature and kind of data the API is able to provide as a service.

4.1.2 Stimulus/Response Sequences

Use case name:	Health-APP
Main objective:	To grant developers that wish to use the API for health projects access to it.
Trigger	The event starts when a developer wishes to use the API to access various patient records
Primary actors	Developers, the API
Secondary actors	Hospital database administrators, Health-API and mobile application administrators
preconditions	The developer must subscribe to the API to access a product key.
Main flow of events	1. Developer approaches the Api management team to request access to the Api.

	1.1 the developer creates an account on the apigee Api portal by registering all details. 1.2 2. The team verifies credentials and responds with payment plans for the Api. 3. Developer pays subscription to the APi management Team 4. An API key is generated and sent to the developer
output	Developer receives an API key
Post condition	The developer is subscribed to the API and able to access its features.
Alternate flow	non

Table 3.

4.1.3

Figure 12.

4.1.4 Functional Requirements

ApiREQ-1: the system should provide a documentation for the API to developers

ApiREQ-2: the system should provide for secure connection to different Health institution databases.

ApiREQ-3: the system should allow secure information sharing between health institutions.

ApiREQ-4: the system should charge a subscription fee to developers that wish to use the Application programming interface.

4.1.5 Nonfunctional Requirements

ApiREQ-5: the system should scale service provision based on demand.

4.2 APPLICATION FEATURE

4.2.1. Home

- Highlights
- Predicted Patient Needs.

4.2.1.1 Description and Priority

The home feature of the application provides **highlights**- a brief rundown on what an individual should do to keep healthy.

Predicted patient needs- provide what patients need based on their patient history.

Priority- High**.**

4.2.1.2. **Stimulus/Response Sequences**

4.2.1.3 **Functional Requirements**

REQ-1: the system should provide a brief rundown of the items within the application.
REQ-2: the system should suggest to a patient what they need based on patient history

4.2.2. TELEMEDICINE

- Ability to order drugs and have them delivered to your doorstep.
- Realtime online consultation with medical personnel. Facetime.
- Ability to order for pickups. This in cases for example: If a pregnant mother is expectant in the next few hours or day. She could order for the health institution to pick her up. Example 2: In an education institution, the institution doctor could call for a hospital pickup in case they are unable to handle the escalated issue, or they are not skilled enough to handle the issue. Like referrals.

4.2.2.1. Description and priority

Telemedicine is the practice of providing medical care and consultations remotely, using telecommunication and information technologies. It allows healthcare providers to assess, diagnose, and treat patients remotely, without the need for in-person visits.
Telemedicine can take many forms, including video consultations, phone consultations, and online messaging with healthcare providers. It can be used for a variety of purposes, including primary care, specialist consultations, follow-up care, and mental health services.

4.2.2.2.

Use case name	**Telemedicine**
Main objective	To provide remote medical assistance to patients through ICT devices
Primary actors	Healthcare provider, patients
Pre-conditions	Patient logs into the app
Trigger	Patient enters their symptoms into the app
Main flow of events	1. The patient launches the telemedicine app on their smartphone or tablet. 2. The patient logs in to the app using their credentials (e.g., username and password). 3. The patient is presented with a list of available services, and they select the option to speak with a doctor about their symptoms. 4. The patient is asked to enter their symptoms into the app, using a form or a chat interface. 5. The patient submits their symptoms to the app, and the app forwards the information to a doctor. 6. The doctor reviews the patient's symptoms and decides whether they can be treated remotely or if an in-person visit is necessary.

	7. If the doctor determines that the patient's symptoms can be treated remotely, they may ask the patient to provide additional information or conduct a virtual examination (e.g., through a video call). 8. Based on the information provided, the doctor makes a diagnosis and recommends a treatment plan. They may also prescribe medication if necessary. 9. The doctor communicates their recommendations to the patient through the telemedicine app, and the patient can follow up with the doctor as needed through the app. 10. The patient can follow their treatment plan and manage their symptoms from the comfort of their own home, without the need to travel to a healthcare provider's office.
Exceptions	The system shall connect the patient to a healthcare provider specialized in the field of the health problem
Post-conditions	The patient follows their treatment plan and manages their symptoms from the comfort of their own home.

Table 5.

4.2.2.3. **Functional Requirements**

REQ-1: The app should allow patients and healthcare providers to communicate with each other through video and audio, using their smartphones or personal computers.

REQ-2: The app should allow patients and healthcare providers to exchange text messages and chat in real-time.

REQ-3: The app should allow patients to schedule appointments with healthcare providers at a time that is convenient for them.

REQ-4: The app should allow healthcare providers to access and update the patient's electronic medical record as needed.

REQ-5: The app should use secure methods (such as encryption) to transmit data, in order to protect patient privacy.

REQ-6:The app should allow patients to pay for telemedicine services through the app, using mobile money or other payment methods.

REQ-7: The app should be compatible with a wide range of devices (such as smartphones, tablets, and laptops), to be accessible to a wide range of users.

REQ-8: The app should be designed with accessibility in mind, and should include features such as text-to-speech and large font options to make it easier for users with disabilities to use the app.

4.2.3. SMART DOCTOR

- A chat feature with an AI health assistant (chatbot)
- Medical Reports.
  - A combination of medical reports from various health institutions

- A selection criterion from which Health Institution to select data and information.
- Automated combination of medical records from various Health Institutions.
- A selection criterion to acquire data from a desired timeline. For example, 2020 and 2019 only.

4.2.3.1. Description and Priority
The smart doctor feature provides for the following:

A.I Doctor- a chat feature with an A.I health assistant

Medical reports- Automated combination of medical reports from various health institutions
- a selection criterion to acquire data from a desired timeline. For example, 2020 and 2019 only.

4.2.3.2.

Use case name	**Smart Doctor**
Main objective	To provide Artificial intelligence powered medical recommendations and diagnosis for patients.
Primary actors	End user application developed in flutter, patients
Pre-conditions	The patient must be logged into the application
Trigger	The process is triggered when the Patient enters their symptoms into the application
Main flow of events	1.1 The patient launches the application on their smartphone or tablet. 1.2 the patient then navigates to the chatbot feature of the application 1.3 on opening, the chatbot initiates a conversation with a greeting. 1.4 the patient then prompts a reply by inputting any signs and symptoms of an illness they are experiencing. 1.5 the A.I powered health assistant then replies with a diagnosis and possible treatment recommendations to the patient 1.5.1 if home treatment I.e., taking water, eating fruits, rest cannot treat the illness then; 1.5.2 the A.I health assistant can book an appointment for the patient with a doctor Else 1.5.2, An ambulance can be dispatched to the location of the patient
Exceptions	Patients in a critical condition I.e., heart attack, unconscious patients, mothers in labor
Post-conditions	A recommended course of action is achieved, and the patient then proceeds with the recommendation by the health assistant
Alternate flow	1. A patient logs into the application 2. accesses the telemedicine feature 3. requests to consult with an available doctor 4. the doctor then performs triage and diagnosis then recommends the next course of action

Table 7.

4.2.3.3. Functional Requirements

REQ-1: the system should be able to generate medical reports from various health institutions
REQ-2: the system should provide an automated combination of health records from various institutions

Nonfunctional Requirements
REQ-3: The system should provide a selection criterion from which health institutions can select data to view at a a given instance i.e., from a given timeline

4.2.4. AI Powered Notification.

4.2.4.1 Prescription and priority

- A notification feature with an AI powered notification. It provides reminders to patients to take their prescribed drugs

4.2.4.2. Use case specification/Narrative

Use case name	***AI Powered Notifications.***
Main objective	To provide reminders to patients to take their prescribed drugs.
Primary actors	End user application developed in flutter, patients
Pre-conditions	The patient must be logged into the application
Trigger	The process is triggered when the Patient's medication details are entered in the database of the hospital.
Main flow of events	1.1 The patient is given a prescription after being checked. 1.2 The patient data for the prescription is entered into the database of the hospital. 1.3 The API then picks that data from the database and feeds it to the application, and they would come to patients as reminders.
Exceptions	Patients have forgotten to take their medication.
Post-conditions	• The reminder has been delivered to the patient. • The reminder has been acknowledged by the patient. • The reminder has been deleted when the doze is complete. • The reminder has been modified if the patient get another dosage for other medication.
Alternate flow	• Snoozing the reminder by the patient. • Modifying the reminder by the patient.

Table 7.

4.2.4.1. Functional Requirements

REQ-1: the system should send notifications to patients undergoing treatment through oral administration to take their prescribed drugs by a doctor
REQ-2: the system should provide health educational videos to patients watch and enlighten people on keeping a healthy lifestyle

5. Other Nonfunctional Requirements

5.1 Performance Requirements

5.1.1. For the API

- The API should be able to handle a minimum of 100 concurrent users and 1,000 requests per second.
- The API should be very user friendly to developers
- The API should be able to handle data from multiple hospital databases without experiencing any significant performance degradation e.g., lagging.
- The API should provide data updates within 10 seconds of any changes to the underlying hospital databases
- The API should be able to handle a minimum of 10GB of data, with the ability to scale to handle larger amounts of data as needed
- The API should provide real-time access to data, with an average response time of less than 100 milliseconds for queries.

5.1.2. For the Application

- The application should be able to handle a minimum of 1,000 patient records per record.
- The application should provide real-time predictions, with an average response time of less than 10 seconds for queries.
- The application should be able to handle data from multiple sources, including electronic health records, laboratory results, and other medical data.
- The application should be able to handle data from multiple patients without experiencing any significant performance degradation.
- The application should be able to provide accurate predictions based on the data it receives, with an error rate of less than 5%.

5.2 Safety Requirements

Safety is a critical concern for any API or application that is used in the healthcare industry. In this case, the API that leverages multiple records of data from various hospital databases has the potential to impact the health and well-being of patients, and it's important to carefully consider the potential risks and safeguards that are needed to ensure the safe and effective use of the API or application
Here are safety requirements:

5.2.1. For the API

Safety is a critical concern for any API that is used in the healthcare industry. In this case, the API that leverages multiple records of data from various hospital databases has the potential to impact the health and well-being of patients, and it's important to carefully consider the potential risks and safeguards that are needed to ensure the safe and effective use of the API
Here are some potential safety requirements for this API:

- The API should be designed to provide secure access to data from multiple hospital databases, using industry-standard authentication and authorization mechanisms. This could

include measures such as encryption and secure access controls to prevent unauthorized access to patient data.
- The API should be designed to protect patient privacy and comply with relevant regulations and policies, such as HIPAA in the United States. This could include measures such as encryption and secure access controls to prevent unauthorized access to patient data.
- The API should be tested and validated to ensure that it meets safety standards and provides secure access to hospital data. This could include a process for monitoring and responding to user feedback and addressing any issues that arise.
- The API should be subject to regular maintenance and updates to ensure that it continues to provide secure access to hospital data and protect patient privacy. This could include a process for monitoring and responding to user feedback and addressing any issues that arise.
- The API should be designed to prevent errors and provide accurate access to data from multiple hospital databases. This could include measures such as validation and verification of the data, as well as algorithms and models that are trained on high-quality data and are able to provide accurate access to the data.

5.2.2. For the Application

- The application should be designed to prevent errors and provide accurate predictions based on the data it receives. This could include measures such as validation and verification of the data, as well as algorithms and models that are trained in high-quality data and are able to provide accurate predictions.
- The application should be designed to provide clear and concise information to users, including any warnings or alerts that may be necessary. This could include user-friendly interfaces and visualizations that make it easy for users to understand the predictions and take appropriate actions.
- The application should be designed to protect patient privacy and comply with relevant regulations and policies, such as HIPAA in the United States. This could include measures such as encryption and secure access controls to prevent unauthorized access to patient data.
- The application should be tested and validated to ensure that it meets safety standards and provides accurate predictions. This could include clinical trials or other studies to evaluate the performance and safety of the application.

5.3 Security Requirements

Security and privacy are critical concerns for any software that is used in the healthcare industry. In this case, the API that leverages multiple records of data from various hospital databases has the potential to impact the privacy and security of sensitive medical data, and it's important to carefully consider the potential risks and safeguards that are needed to ensure the safe and secure use of the API.

Here are the security requirements for:

5.3.1. The API

- The API should be designed to provide secure access to data from multiple hospital databases, using industry-standard authentication and authorization mechanisms. This could include measures such as encryption and secure access controls to prevent unauthorized access to patient data.
- The API should be designed to protect patient privacy and comply with relevant regulations and policies, as set by the ministry of health. This could include measures such as encryption and secure access controls to prevent unauthorized access to patient data.
- The API should require user authentication to access the data, using methods such as username and password or two-factor authentication or biometric scanning using

thumbprints. This could include a process for verifying the identity of users and granting them access to the data they need.
- The API should be tested and validated to ensure that it meets security and privacy standards and provides secure access to hospital data. This could include a process for monitoring and responding to user feedback and addressing any issues that arise.

5.3.2. The Application

- The application should make sure that only authorized users can access the application and the features it contains. This may involve implementing authentication mechanisms such as login credentials or two-factor authentication.
- The application should help users create strong passwords that cannot easily be guessed by hackers i.e., prompting users to have numbers and other symbols in their passwords
- In terms of external policies and regulations, the application should ensure that there are specific security and privacy regulations that it must comply with

5.4 Software Quality Attributes

Reliability: The API should be highly reliable, with minimal downtime or disruption to service. This could be measured in terms of uptime, or the number of successful API calls over a given period of time.

Interoperability: The API should be able to seamlessly integrate with other systems and technologies, allowing developers to easily access and use the data it provides. This could be verified through testing with a variety of different systems and technologies.

Usability: The API should be easy to use, with clear documentation and intuitive interfaces. This could be assessed through user testing and feedback from developers.

Security: The API should have robust security measures in place to protect the data it provides and prevent unauthorized access. This could be verified through security audits and penetration testing.

Flexibility: The API should be flexible and adaptable; allowing developers to easily retrieve and manipulate the data it provides in a variety of different ways. This could be verified through testing with a variety of different data access and manipulation scenarios.
In terms of relative preferences, it may be more important for the API to prioritize reliability and security over other attributes such as ease of use or flexibility. However, the specific priorities will depend on the needs and requirements of the developers and customers using the API.

Availability: The API should be available for use by developers at all times, with minimal downtime or disruption to service. This could be measured in terms of uptime, or the number of successful API calls over a given period of time.

Correctness: The API should provide accurate and up-to-date data, with minimal errors or inconsistencies. This could be verified through testing and validation of the data provided by the API.

Maintainability: The API should be easy to maintain and update over time, with clear documentation and support for ongoing development and evolution. This could be assessed through feedback from developers and users of the API.
Portability: The API should be portable, allowing developers to easily integrate it with a variety of different systems and technologies. This could be verified through testing with a range of different systems and technologies.

Reusability: The API should be designed in a way that allows developers to easily reuse and repurpose the data it provides in a variety of different contexts. This could be verified through testing with a range of different data access and manipulation scenarios.

Robustness: The API should be robust and resilient, able to handle a high volume of requests and data without breaking or crashing. This could be verified through performance testing and stress testing of the API.

Testability: The API should be designed in a way that allows for easy testing and verification of its quality characteristics, such as reliability, accuracy, and usability. This could be assessed through the availability of testing tools and documentation, as well as feedback from developers.
In terms of relative preferences, it may be more important for the API to prioritize availability, correctness, and security over other attributes such as ease of use or flexibility. However, the specific priorities will depend on the needs and requirements of the developers and customers using the API.

5.5 Business Rules

Access control: The API should have mechanisms in place to control who can access the data it provides, and under what circumstances. This could include authentication and authorization processes to verify the identity of users and determine their access permissions.

Data privacy: The API should have measures in place to protect the privacy of the individuals whose data it provides, in accordance with relevant laws and regulations such as HIPAA in the United States. This could include encrypting the data, anonymizing it, or limiting access to specific data fields or sets.

Data quality: The API should ensure the quality and accuracy of the data it provides, through processes such as validation, verification, and regular updates. This could include mechanisms for feedback and reporting of errors or inconsistencies in the data.

Data usage: The API should have rules and guidelines in place for how the data it provides can be used, including restrictions on sharing, copying, or republishing the data without permission. This could include terms of service or licensing agreements that developers must agree to in order to access the data.

Support and maintenance: The API should have support and maintenance processes in place to address issues, provide updates, and respond to user feedback. This could include a support team, a user forum or community, and regular updates and improvements to the API.

6. Other Requirements

<Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.>

Appendix A: Glossary

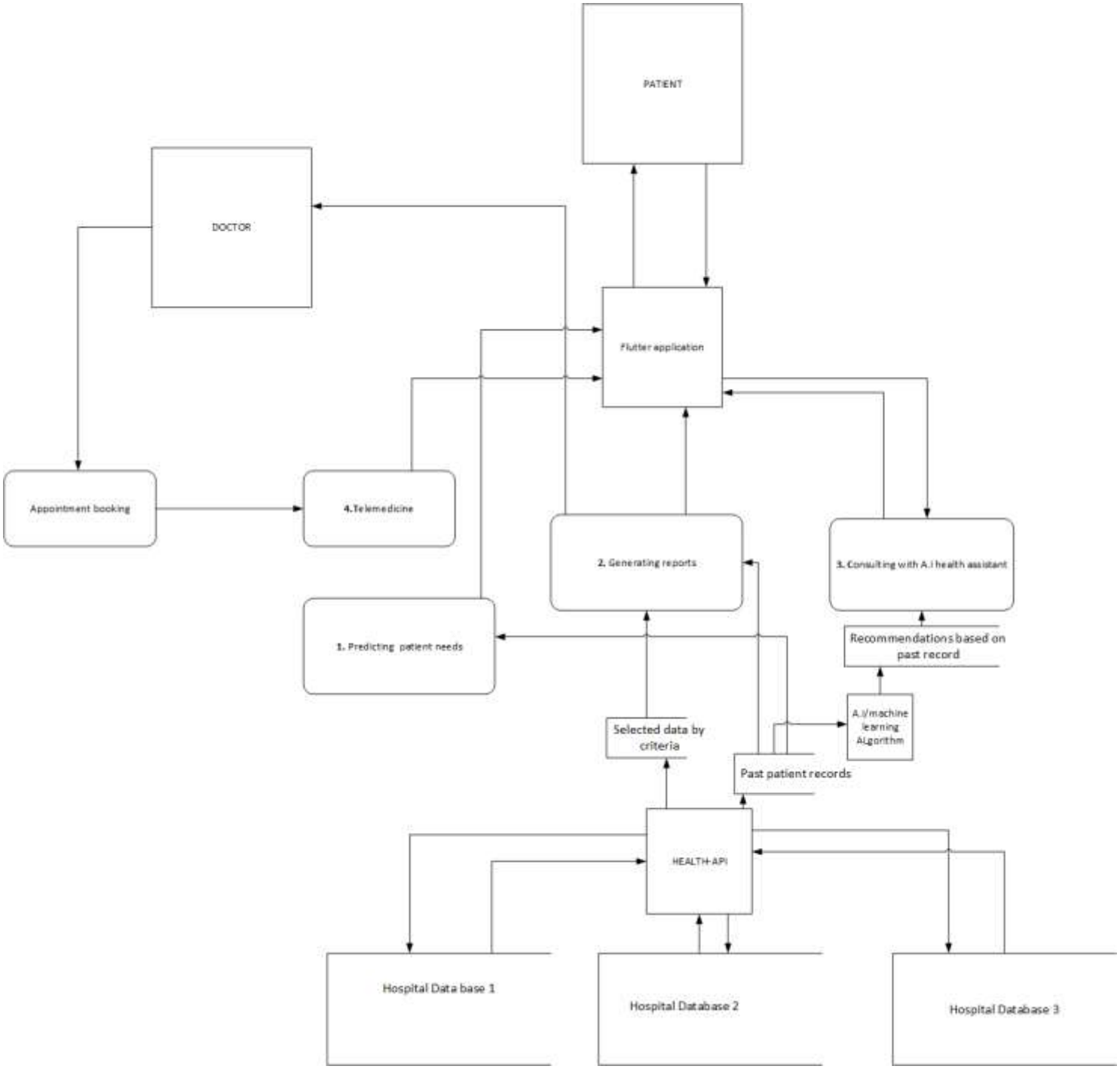
Software Requirements Specification (SRS)	A description of a software system to be developed and how it is expected to perform.
Chatbot	A software that simulates human-like conversations with users via chat. Its key task is to answer user questions with instant messages.
Mnemonic	A learning technique that aids information retention or retrieval (remembering) in the human memory for better understanding.
SMTP (Simple Mail Transfer Protocol)	It is standard for the transmission of email on a computer network.
TLS (Transport Layer Security)	Is a cryptographic protocol that aims at providing secure communication over an IP network.
API (Application Programming Interface)	It is a way for two or more computer programs to communicate with each other.
IBM (International Business Machines)	
AI (Artificial intelligence)	It is the simulation of human intelligence processes by machines, especially computer systems.
GCP (Google Cloud Platform)	It is a suite of cloud computing services that runs on the same infrastructure that Google uses internally for its end-user products.
RAM (Random Access Memory)	It is a form of computer memory that can be read and changed in any order, typically used to store working data and machine code.
GB (gigabyte)	It is a specific unit of data that's equal to about 1 billion bytes of data.
Wi-Fi (Wireless Fidelity)	A system used for connecting computers and other electronic equipment to the internet without using wires.
GSMA (Global System for Mobile Communications Association.)	
GPS (Global Positioning System)	It is a network of satellites and receiving devices used to determine the location of something on Earth.
REST (Representational state transfer)	It is a software architectural style that describes a uniform interface between physically separate

	components, often across the Internet in a client-server architecture.
SQL (Structured Query Language)	It is a domain-specific language used in programming and designed for managing data held in a relational database management system, or for stream processing in a relational data stream management system.
OAuth	It is an open standard for access delegation, commonly used as a way for internet users to grant websites or applications access to their information on other websites but without giving them the passwords.
JSON (JavaScript Object Notation)	It is an open standard file format and data interchange format that uses human-readable text to store and transmit data objects consisting of attribute–value pairs and arrays.
FTP (File Transfer Protocol)	It is a standard communication protocol used for the transfer of computer files from a server to a client on a computer network.
HTTP (Hypertext Transfer Protocol)	It is an application layer protocol in the Internet protocol suite model for distributed, collaborative, hypermedia information systems.
HTTPS (Hypertext Transfer Protocol Secure)	It is an extension of the Hypertext Transfer Protocol. It is used for secure communication over a computer network and is widely used on the Internet.
IMAP (Internet Message Access Protocol)	It is an Internet standard protocol used by email clients to retrieve email messages from a mail server over a TCP/IP connection.
POP (Post Office Protocol)	It is a type of computer networking and Internet standard protocol that extracts and retrieves email from a remote mail server for access by the host machine.
TCP (Transmission Control Protocol)	It is a standard that defines how to establish and maintain a network conversation by which applications can exchange data.
UDP (User datagram protocol)	It operates on top of the Internet Protocol (IP) to transmit datagrams over a network.
PDF (Portable Document Format)	It's a versatile file format created by Adobe that gives people an easy, reliable way to present and exchange documents - regardless of the

	software, hardware, or operating systems being used by anyone who views the document.
XML (Extensible Markup Language)	Extensible Markup Language is a markup language and file format for storing, transmitting, and reconstructing arbitrary data.
CSV (Comma-separated values)	A comma-separated values file is a delimited text file that uses a comma to separate values.
APP (Application)	An application program is a computer program designed to carry out a specific task other than one relating to the operation of the computer itself, typically to be used by end-users.
ICT (Information and Communications Technology)	Refers to the convergence of audiovisuals and telephone networks with computer networks through a single cabling or link system.
HIPAA (Health Insurance Portability and Accountability Act)	

Appendix B: Analysis Models

A DATA FLOW DIAGRAM SHOWING FLOW OF DATA THROUGH THE API AND VARIOUS PROCESSES


PATIENT
DOCTOR
Flutter application
Appointment booking
4.Telemedicine
2. Generating reports
3. Consulting with A.I health assistant
1. Predicting patient needs
Recommendations based on past record
A.I/machine learning ALgorithm
Selected data by criteria
Past patient records
HEALTH API
Hospital Data base 1
Hospital Database 2
Hospital Database 3