MASENO UNIVERSITY

SCHOOL OF COMPUTING AND INFORMATICS

DEPARTMENT OF COMPUTER SCIENCE

CCS 406

COMPUTER SCIENCE PROJECT 11

TITLE:

SMART AFYA CROSS PLATFORM MOBILE APPLICATION

CCS/00255/019

EZEKIEL MAINA WAITHERA

A project submitted to the department of Computer Science in the School of Computing and Informatics in partial fulfilment for the requirements for the award for the degree of Bachelors of Science (BSc) Computer Science Maseno University

Declaration

This project report is my original work, to the best of my knowledge it has not been presented to any other examination body, and the information given here is a true reflection of my research.

Name: WAITHERA EZEKIEL MAINA **Registration No: CCS/00255/019**

Signature: ……. **Date: …….**

This project has been submitted for examination with my approval as University Supervisor.

…….. ……... ……...…….

Name ***Signature*** ***Date***

Maseno, Kenya

Dedication

This project work is dedicated to the almighty God who has been the source of knowledge, wisdom and inspiration and to all those who have been affected by the challenges addressed herein. It is a tribute to the individuals who have lost their lives prematurely, as well as to their families who endure the pain of their absence. I also extend my dedication to the tireless efforts of healthcare professionals, researchers, and organizations committed to combating this critical issue. May this project serve as a catalyst for positive change and contribute towards a future where premature deaths are minimized, and every individual can lead a healthy and fulfilling life

Acknowledgement

First and foremost, I humbly express my deepest gratitude to the Almighty God for the abundant grace bestowed upon me throughout the entirety of this project. It is by His divine guidance and unwavering support that I have found the strength to persevere and overcome the challenges that lay before me.

In this heartfelt dedication, I wish to extend my profound appreciation to my project supervisor, the extraordinary Dr. Henry Okoyo. Words cannot adequately capture the immense impact he has had on this endeavor. With unwavering dedication, he not only provided invaluable guidance but also instilled in me a belief in my abilities and the power of relentless determination. His unwavering faith in my potential has been an unwavering source of motivation, propelling me forward even in the face of uncertainty.

To Dr. Okoyo, I offer my heartfelt thanks for breaking down this project into manageable steps, offering profound insights, and nurturing my growth as a researcher. Your profound wisdom, expertise, and genuine care have elevated this project to new heights and will forever be etched in my heart.

Lastly, I would like to extend my appreciation to my family, friends, and loved ones who have been unwavering pillars of support throughout this journey. Your love, encouragement, and understanding have been my refuge during moments of doubt and fatigue. It is your unwavering belief in me that has given me the courage to persist and bring this project to fruition.

To all those who have played a role, big or small, in shaping this project, I offer my heartfelt gratitude. May the culmination of our collective efforts serve as a testament to the power of determination, guidance, and unwavering support. Together, we have shown that with faith and resilience, we can overcome any obstacle and create a lasting impact.

Abstract

The alarming rate of premature deaths worldwide has emerged as a compelling reason to undertake a project aimed at addressing this critical issue. Premature deaths, defined as fatalities occurring before the expected age of mortality, have devastating consequences on individuals, families, and societies as a whole. With almost 5 billion mobile phone users in the world, health care providers and researchers realized the potential of using mobile technologies for supporting healthy living and health education. Good health has been a major concern since the inception of mankind, for some people attaining good health required exercising, taking balanced diet, relaxing and taking prescribed medicines or pills routinely. The recent advancement in technology provided an enabling technique to solve these types of problems by designing and developing an application that could run on smartphones and which patients would find easy to carry along.

The aim of the study was therefore to design and develop Smart Afya, a smart health-based cross-platform mobile application. It offers four functionalities for the users i.e. article surfing, fitness, balanced diet, and medicine reminders. Firstly, the users can perform exercise anywhere they like by using the Exercise section which contains different types of workouts as well as exercises that could support and aid them to practice a healthy lifestyle. The exercises are age and disease specific exercises. The users of the app are guided towards the diet routines based on their health conditions. Thirdly, Smart Afya aids in medicine adherence. Many patients find it very difficult to keep track of taking their medication at the right time and proportion. This happens especially if it involves taking pills or medication on a daily basis due to several reasons such as heavy workloads, forgetfulness, old age, and alterations in day-to-day behaviour affecting the total well-being of the patient, delaying the curing time, raising the total medical cost of the patient, and could be a matter of life and death. Lastly, the app tries to curb on Misinformation as a factor that poses dilemma in ICT. Users are able to surf medical verified contents as per their interests.

The results of this study indicate that Smart Afya is an innovative approach to improving health management and promoting healthy living and is a promising solution for patients who struggle with keeping track of their health routines and medication adherence. The app's user-friendly interface and personalized approach to health management were well-received by users, as evidenced by positive feedback and high user engagement. By providing patients with a tool to manage their health and well-being, Smart Afya can help to reduce the risk of premature deaths and improve overall health outcomes. The app's ability to provide personalized fitness, diet, and medication reminders in an easily accessible format makes it a valuable resource for patients and healthcare providers alike. This fosters an effective transition on health matters while moving from parental or practitioners' reminders to self-care reminders and personal motivation.

Table of Contents

List of Tables

List of figures

Acronyms and Abbreviations

GUI – Graphical User Interface

WHO -World Health Organisation

ICT- Information Communication Authority

OpenSSL- Open Secure Socket Layer

HCI- Human Computer Interaction

HMAC- Hash-based Message Authentication Code

ERD- Entity Relationship Diagram

OS – Operating System

UI -User Interface

PDA- Personal Digital Assistant

PC-Personal Computer

iOS- iphone Operating System

SSAD- Standards System Analysis and Design

CBIS-Computer Based Information System

KNBTS-Kenya National Blood Transfusion System

Definition of Terms

Cross-platform – A cross-platform computer product or system is a product or system that can work across multiple types of platforms or operating environments.

Mobile application-A mobile application, most commonly referred to as an app, is a type of application software designed to run on a mobile device, such as a smartphone or tablet computer.

Smart health-The key concept of smart health includes eHealth and mHealth services, electronic record management, smart home services and intelligent and connected medical devices.

Cipher- is an algorithm for performing encryption or decryption—a series of well-defined steps that can be followed as a procedure.

Decipher -to find the meaning of (something that is difficult to read or understand)

Encryption -the process of converting messages in ordinary language, or other information into a secret coded form that cannot be interpreted without knowing the secret method for interpretation, called the key.

Authentication-is the process of proving that you are who you say you are. This is achieved by verification of the identity of a person or device

Authorization-*Authorization* is the act of granting an authenticated party permission to do something. It specifies what data you're allowed to access and what you can do with that data.

Surfing-is the navigation from page to page on the World Wide Web using hyperlinks. So basically, it's the thing that we call internet browsing

Discrepancy-the state or quality of being discrepant or in disagreement, as by displaying an unexpected or unacceptable difference; inconsistency.

ERD- a graphical representation that depicts relationships among people, objects, places, concepts or events within an information technology (IT) system.

Context diagram- a diagram that outlines how external entities interact with an internal software system.

Use case- use case is a written description of how users will perform tasks on your website

CHAPTER ONE: INTRODUCTION

1.0 Introduction

This chapter discusses the whole study background by introducing the design, development of the mobile based application, study problem statement, study aim, study objectives, study significance and application limitation.

1.1 Background Information

There is a popular saying that health is wealth. Health is one of the most important things for most individuals simply because not having good health can lead to a very miserable life (Leonard, 2008). In recent times new diseases have emerged which needs to be taken care off by taking medicine or pills routinely (Ohayon, 2002), for many people there are consequences attached to not taking the prescribed medication in proper time and proportion because it can be the difference between life and death. In addition, not taking the prescribed medicine at the proper dosage or time can also result to what is referred to as medicine adherence which can be stated to as the extent at which the medicine is taken at the correct doctor prescribed time and proportion (Hughes, 2004). Medicine adherence is a very serious problem because it may affect the total well-being of the patient, delaying the curing time and also raising the total medical cost of the patient (Grooves et al., 2013). Yet with all the aforementioned consequences people of different ages still forget to take their prescribed medicines or pills in due time and proportion (Banerjee, 2009). However, there are different reasons for been forgetful ranging from busy schedules, old age, cognitive disorders, bad working conditions, Alzheimer disease, loss of memory, dementia, people with emotional problems, stress, anxiety, depression etc.

People at this time are very concerned about their health and nutrition. Most of them are dieting, but are not good at choosing the right nutrition. A healthy diet is a healthy eating habit. A good diet is a balanced diet that meets all the needs of the body, such as foods that contain nutrients such as proteins, carbohydrates, fats, vitamins, mineral salts, water and salts in the quantities required. Diet is an important part of our daily lives. Applying a diet can mean making some good choices about nutrition (eating more fruits, vegetables, and fiber) or bad choices (skipping food, eating too little or not eating enough food). Diet is the amount of food used by a person in a certain amount. Diet often implies the use of specific nutrition for health or the reason for weight management. A specific diet can be selected for weight loss or weight gain. Some foods are specifically recommended or altered for compliance with pertinent dietary requirements.

Diet also has many different types, namely the Paleo diet, atkins diet, vegetarian diet and raw food diet.

The diet should also follow the proper syllabus because different people will have different dietary nutrition requirements based on their weight, age, height, gender and daily activity. For example, having two dieters weighing 60kgs and 140 kgs, the system will set the appropriate rates and quantities according to their weight. Therefore, we recommend this system to help those who want to diet in the right way.

People often overlook the importance of keeping their fitness at average level or above in this modern society. Obesity as well as other diseases are creeping in without any major alert because people are always busying with their works and lives, all they would care is the money that they earn and people that they love. All these diseases must be envisaged as the developments of the diseases are not inevitable and they are largely preventable through lifestyle change (World Health Organization, 2000). Even if they notice the jeopardy that they are facing, they do not have enough spare time or motivation to keep them away from the risk of dying from horrible diseases. Therefore, an app that promotes fitness is the solution for this problem because the personal item that is relatively close to the modern people is smartphone and the mobile applications installed inside it. Additionally, some people also prefer to use the low-cost and wide-availability of the fitness-tracker devices (de Zambotti et al., 2015).

Nevertheless, the recent advancement in technology has provided an enabling technique to solve these types of problems using different methods, one of the methods used is by buying a device designed purposely for a given function for instance to remind patient to take their medicines in the prescribed time and proportion, the aforementioned solution seems to be ineffective and costly (Riehemann et al., 2009). Rather, the use of mobile application seems to be more effective as it does not necessitate a need to procure an additional device and because most people make use of smart phones. Apparently, dozens of hours have been spent by people daily on using the mobile app and it is the clear advantage of this solution.

The study chose to adopt one of the most widely used smartphone Operating systems which are Android OS, and iOS because they are the top notch in the smartphones business. However, the two seem to be very effective in mobile phones according to the topnotch engineers (Nosrati, 2012).

The iOS is an operating system designed and developed by the Apple Inc. mainly for the Apple hardware gadgets such as the iPhone, iPad and iPod Touch, the iOS was developed starting from the earliest stage empower engineers to make convincing versatile applications that take full preferred standpoint of each of the handset. The operating system is mainly created for mobile devices and has been the second most widely adopted after the Android OS. For this reason the proposed mobile application is compatible with smartphones running on the most popular mobile based operating systems which, the iOS and android, basically the application helps people improve on their health by reminding patients or users to take their medicine in due time and actual proportion using an automated alarm ringing system, guiding through fitness and consumption of balanced diet.

Hence, the project aims to design, develop and implement an automated cross platform mobile application for helping people improve on their health e.g. medicine or pill reminder as prescribed by a physician to patients. The designed application would help patient to maximize the full benefit of the medicine and abstain from the risk that result as not taking the medicine or pill within the stipulated time prescribed by the specialist as well as work on their fitness and balanced diet.

1.2 Problem Statement

The high occurrence of premature deaths globally has provided a strong impetus for initiating a project that focuses on tackling this urgent problem. Premature deaths, referring to individuals passing away before reaching the expected age of mortality, carry severe - repercussions for individuals, their families, and society at large.

Living a healthy life cannot be underrated. In recent times new diseases have emerged which needs to be taken care off by taking medicine or pills routinely. In addition, the working conditions of some individuals is bad and hectic and as such had resulted them in forgetting to take their administered medicine or pills in the appropriate timing or proportion and for some people its old age, most elderly people suffer from dementia i.e. forgetfulness. Not taking the prescribed medicine at the proper dosage or sometimes could result to what is referred to as medicine adherence which can be stated to as the extent at which the medicine is taken at the correct doctor prescribed time and proportion. Medicine adherence, exercising and consumption of balanced diet are very serious problems because they may affect the total well-being of the patient, delaying the curing time and also raising the total medical cost of the patient.

A frustrating characteristic of fitness is that the results cannot be seen immediately as they must spend months and years to achieve the ultimate results (Rejeski and Kenney, 1989). So, one of the concerning problems in the society nowadays is that they have a poor tracking ability (Boreham et al., 2004) where it is difficult to keep track of their exercise statistics such as calories, distance travelled, speed, as well as steps taken. It is virtually impossible to keep track of all these statistics conveniently due to several reasons such as lack of measuring tools and self-perseverance to carry out the exercise consistently which will make them to give up eventually. Some of them also find it extremely hard to be motivated and keep exercising constantly due to the fact that exercising is totally voluntary based (Rejeski and Kenney, 1989). Other than that, people also must not forget about diet plays an important part in achieving a healthy lifestyle yet it is also a tough task when it comes to keep track of their diets in terms of what they consume and how much they consume every day.

The above have critical effects when alone and wholesomely contributing to premature deaths (WHO,2020b) and also recommended lifestyle (WHO, 2000).

As such these problems prompt the need of designing and developing a cross platform mobile application that could help in curbing out the aforementioned problems by reminding patients to take their medicines as prescribed by the doctor within the stipulated timing and dosage, tracking fitness progress and recommending of balance diet enabling people to adjust their routines accordingly. The app will play an assistant role in the users' daily life while delivering information such as health articles and so on.

1.3 Proposed solution

As we have seen how critical medical adherence, exercising and taking of balanced diet are for a healthy lifestyle (WHO, 2000), then there is urgent need to address these measures of healthy living in an improved manner. The three are the main addressed in the proposed system. The proposed system is going to be efficient and accurate in retrieval health records thus help in self-health management. The system would also enable privacy at its highest peak as it's a single user system avoiding any room for stigmatization. It takes priority of self-care among its users, an individual being able to do all that is required of him for a healthy living without or with minimal expert interference, in prescription adherence, exercising and consumption of balanced diet.

The system would be able to support medical adherence, fitness tracking, diet recommendations, processing and distribution of health information as well as article surfing in a solution the project aims to implement. The system applies the prototype methodology since the approach can be used in development of user interfaces of different applications independently of the type of desktop web-based or mobile. The user interface is crucial in the development of software applications. Software prototypes support development teams to explore usability, usefulness and acceptability of their project. The solution to be develop will be mobile-based, all kinds of features are available to the users with just several touches so that they can try out different functionalities, thus accessible over wide geographical regions due to the huge acceptance and distribution of mobile devices.

It is portable, which means that one could just install the Mobile application on the phone and they could literally go anywhere and still be able to carry out its functionality, for instance exercising as per the fitness module. As far as the feature is concerned, there are various functions in the fitness module which are categorized by age and disease. The module includes categorized exercises and area of body impact and provide multimedia guide on the different exercises as well as a counter for each. For prescription adherence, the application shall employ a notification system. Users are able to set time, frequency of take, manage reminders, track prescriptions' quantity and so on. This is coupled with balance diet recommendations and article surfing.

The project utilizes a Database Management System for accessibility to many concurrent users and thus enable capturing and manipulation of data in the database. The system connects all the users to a central database where they can concurrently interact with available information. The system abstracts the level of information to be shared to the different users. Non-registered users of the system are able to benefit only on article surfing but to benefit on the app's broader functionalities the user can make a registration request where they can feed their general information.

The system enables the organization and management of records and thus enable people have ease in retrieval as well availability of reports on their medical intakes. Additionally, the concept of gamification like leader board is also added into the app to increase the extrinsic motivation and diminish the boredom of the users so that they will be motivated to practice healthy lifestyle consistently by using the app. For clarification, behaviour that is driven by external rewards called extrinsic motivation can be raised through the in-game achievements

(Cherry, 2016). This has proven to be therapeutic, the users are motivated and excited to practice a healthy lifestyle day in day out.

1.4 Objectives

The general and specific objectives of the project are described below:

1.4.1 General Objective

The main objective of the study is to **design**, **develop** and **implemen**t a cross platform mobile application to be used in promoting a self-care healthy lifestyle in support of medical adherence, fitness and consumption of balanced diet.

1.4.2 Specific Objectives

The following are the specific objectives that the study aims to achieve:

1. To design and develop the system the smart health based cross platform mobile application.
2. To test the developed mobile application.
3. To deploy the system for use via build apks.

1.5 Research questions

The following are the research questions for the study?

1. What are the design tools and how are they used in developing the system?
2. What elements and features of the system are going to be tested?
3. What methods will be used to deploy the system effectively?

1.6 Scope of the project

After mentioning the problems in the previous subsections, the deliverable of the project is a smart health cross platform mobile application called *Smart Afya* in which the users can use the app to keep track of their exercise, diet, prescriptions and so forth. Meanwhile, tons of information will be available in the mobile application so that the users can be provided with useful as well as helpful tips and tricks about health to make their life healthier than before.

The system functions and features include the following:

1. **Registration-** This function allows people to register as a user to interact with the system. The system would require the user to login before viewing and editing any information.
2. **Summary report-** The system can generate reports to summarize prescriptions adherence to enable them to determine gaps in their health status and predict likely impact or shortages of prescriptions and take necessary actions.
3. **Exercise Module -**This module contains several features in which it can be divided into different parts categorized according to the age and special cases in which they impact the most.

   The exercises could be indoor or outdoor depending on the user's choice. The exercises part allows the users to do exercise routines which can be done while keeping track of the duration.
4. **Balance Diet Recommendation -**This application exposes its users to diet and healthy practices that they should maintain for a better healthy lifestyle based on their illness/specialised care.
5. **Reminder Feature -**This feature enables the users to create reminder for prescriptions so that they could be reminded to take them or act accordingly on specific time and date. It allows for anonymous message content in order to enhance privacy.
6. **Help Facility-** The application employs the use of a text contextual based Chatbot as a help facility. It is aimed at helping users have an ample time in their engagement with the application. Users can interact with the bot to get their questions answered.

1.7 Significance of the Study

- The designed mobile application could help people achieve best of the best nutritional values by taking of recommended balanced diet.
- With people living a healthy life blood transfusion process will be run with ease as donors will be in matching healthy criteria for donating. This impacts trust between donors and KNBTS positively.
- The designed application could make a great impact in reduction of premature deaths in current statistics.

- The study demonstrates how to design the smart health cross platform based mobile application to enhance self-care among individual persons.
- The designed application would help patients to maximize the full benefit of the medicine and abstaining them from the risk that result as not taking the medicine or pill within the stipulated time and proportion as prescribed by the specialist.
- The designed cross platform-based medicine reminder feature will be of great help for patients suffering from range of problems such as forgetfulness, busy schedules, old age, cognitive disorders, bad working conditions, Alzheimer disease, loss of memory, dementia, people with emotional problems, stress, anxiety, depression, and also individuals with a very hectic work schedules or lifestyle.
- The application would help patients staying fit by having a personal guide to exercising and remembering them to take their medicine in an appropriate time and proper proportion.
- A healthy citizen impacts the economy of the nation positively as other considerations in its administration's agenda can be looked into with depth.

Health spans across all dimensions of life social, economic, political.

1.8 Justification

Health is a very important aspect of human life. It is the second most basic level in Maslow's Hierarchy of Needs. Especially now, in the recent years with the global pandemics around us, being physically healthy can make a lot of difference. Even though living healthy is a popular trend, there is still a large part of the population that is experiencing health problems, which can lead to death. In 2015, over 40 million global deaths were due to such diseases (WHO, 2020b). Some of them could be avoided or even reversed through everyday choices that serve us well rather than harm us. World Health Organization (WHO) states that 80% of premature death could be prevented (WHO, 2020b).

The WHO has projected that with the fast-growing population, by the year 2030 10% of the population would need blood transfusion services as a way enhancing wellness of an individual WHO recommends exercising and taking of balanced diet (WHO, 2000). A major blow is on the differentiation in users' skill and fitness level as well as each and every one of them has different needs (Higgins, 2016). Besides, to the poor tracking ability of people in diet and

exercise statistics where it may lead to some undesired consequences, there is an urgent need in assisting the people to improve some of the inappropriate dietary habits (Boreham et al., 2004) and exercise frequency gradually by providing statistics or reminding them through the reminder feature. Moreover, the inclusion of reminder feature in the project also intends to raise the awareness of people about medications, exercising more and taking of balanced diets in order to minimize the risk of disease (United States Department of Health and Human Services, 1996).

This mobile application provides a fulcrum point in achieving this goal.

1.9 Overview of the Study

The whole study writes up consists of five chapters, the following list gives a brief overview of each chapter.

- **Chapter one:** This chapter discusses the whole study background by introducing the design, development of the smart health cross platform mobile application, study problem statement, proposed solution, study aim, scope of the study, study objectives, study significance and Justification.
- **Chapter two:** In this chapter a theoretical review is done and obtaining of a conceptual view of the system in its first section. This study reviewed some literatures regarding the design, development and the implementation of smart health mobile application from different academic sources. Four existing smart health applications are studied (Medisafe, WebMd, Myplate and MyFitness pal). The chapter expresses the challenges in smart health mobile applications as well as the importance of the proposed system in its integration feature and the system as a whole.
- **Chapter three:** This chapter describes the prototype methodology to be adopted during the system implementation. A feasibility study is also conducted deeming the proposed system feasible in the four aspects. The proposed system is highly supported as shown by the pie chart later. The information for determining the requirements (requirement elicitation) i.e. functional and nonfunctional is gathered by use of questionnaires and reviewing of existing CBIS documentation. The chapter also brings out the system specifications including technical specifications, system description, proposed testing method etc. Hardware requirements for the development, domain of execution and the system architectures are realized in this chapter. An explanation is also provided on how the system analysis and design is achieved by use of use case diagrams, User case descriptions and use of sequence

diagrams, context diagrams, ERD diagram and the database schema of the proposed system in depth

- **Chapter four:** The phase of code generation and testing was crucial for ensuring its functionality and reliability. During code generation, I diligently wrote the actual code for the system using the proposed technologies and adhering to the established design approaches. This process enabled the translation of the design into a fully functional application. Subsequently, rigorous testing was conducted to identify any errors or discrepancies and ensure that the system was functioning correctly. In this phase, six test cases were utilized, and all of them yielded positive results, indicating that the application was performing as intended. The successful execution of these test cases provided confidence in the application's capabilities and demonstrated its ability to meet the defined requirements. The code generation and testing phase played a pivotal role in delivering a high-quality and reliable mobile application for health management.

- **Chapter five:** The highlights the successful development of a comprehensive tool for health management. The application's diverse modules cater to specific user needs, including medical reminders, article surfing, disease-based diet recommendations, and exercise categorization. The conclusion emphasizes the user-friendly and accessible nature of the application, promoting informed decision-making and empowering users to take control of their health. The recommendations focus on continuous improvement, expanding the disease database, integration with wearable devices, gamification, localization, collaboration with healthcare providers, privacy and security measures, and user education and support. Implementing these recommendations will further enhance the application's functionality and user experience, ensuring its ongoing relevance and effectiveness in supporting individuals' well-being.

.

CHAPTER TWO: LITRATURE REVIEW

2.0 Introduction

The chapter aims to review relevant articles related to the project description and would also aid in designing the system to meet its new goals. It comprises three subsections. The first subsection deals with theoretical review and conceptual frameworks of smart health mobile applications that exist. The second section deals with other works that are related to the named project. The last section deals with the overall summary that is an outline of the entire chapter.

2.1 Theoretical review

The value of health is essential; thus, we need minimal latencies in the retrieval and transaction of information. With the help of smart health mobile, it becomes easier to promote self-care. These systems also update data regularly to ensure that there are no errors.

In the smart health applications, there is a central database where all the information of the different users in different hierarchies is stored, as well as health related information. Below is a general schematic diagram of the general workings of a smart health application.

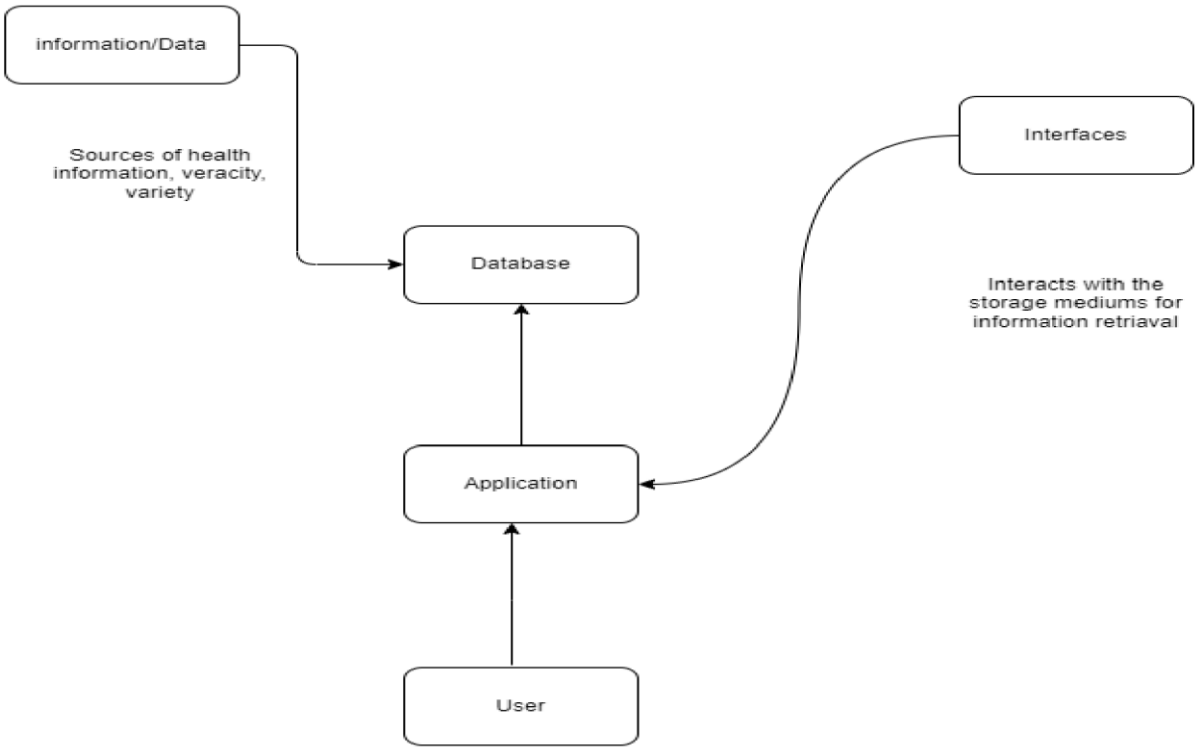


Figure 1: General Concept of smart health mobile applications

The proposed solution makes use of mobile-based technology. The user logs into the system, where they enter their credentials, and the details are validated before being cross-matched with the database. If the login credentials are declined, the user will have to re-enter their credentials, but password retrieval mechanisms via email will be available if it persists. However, they would head to their respective index pages if the credentials were accepted.

The diet page, medical reminder, fitness, help facility are strictly for users who are already registered in the system unlike the surfing page. The users can request disease related diet combinations, and the system would recommend diets that would be applicable; furthermore, the system generates health practices to strengthen the immunity. The system also allows its users to surf, set medical reminders and participate in fitness programmes.

Existing smart health applications lack the integration of the 3 i.e diet, fitness and medical reminder based on a given disease. Since the system is heavily based on data transactions, the smart health applications regularly update their data and share information on their quantities. The system sends mail reports to the users on their medical information.

An individual can monitor all these activities within the application and also be able to make a high-level decision based on the activities downstream.

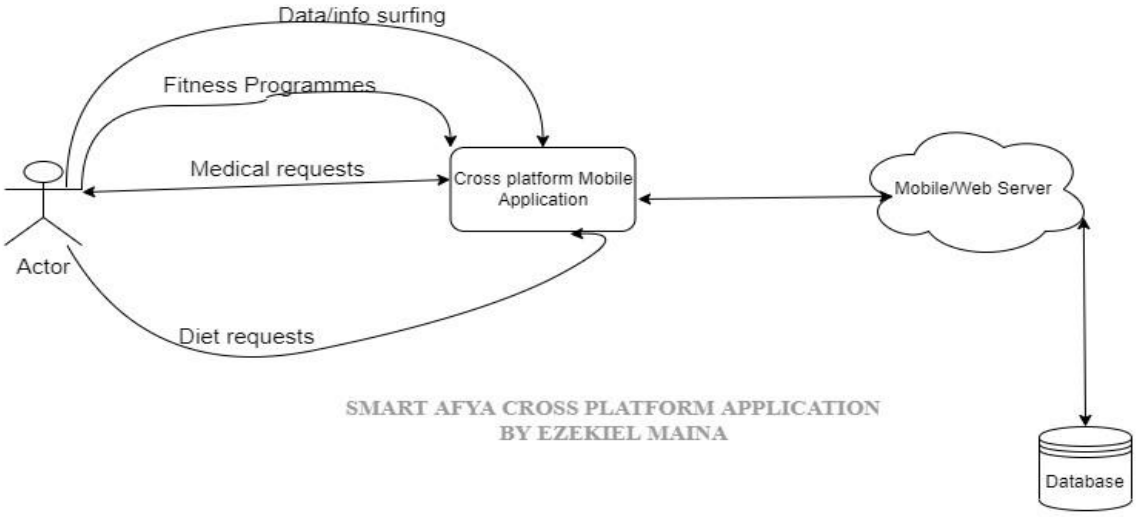


Figure 2: Conceptual review of the system

2.2 Existing Systems and Related work

In recent years, the use of mobile health (mHealth) applications such as WebMD, Medisafe, Myfitness Pal, Myplate among others has become increasingly prevalent, as more and more individuals turn to their mobile devices for health information and support. The applications discussed include both cross platform mobile applications and platform dependent mobile applications. This section presents the above-mentioned mobile applications through various literature reviews of different case studies, their benefits, how they manage records and processes, and how effective they are from various journals, e-books, and the internet.

2.2.1 WebMD Mobile Application

WebMD is a popular mobile application that provides users with access to a wide range of health information, including articles, videos, and medical reference materials for exercising. The program was designed based on the alert ringing framework to remind patient about the intake of their medications. No need for patients to keep a written dosage timing in a paper, they just need to enter these prescribed timings on the apps. The medication reminder could be set for various medical prescriptions with different timings. The app remembers the patients of taking their medicines via text messages.

One of the strengths of the WebMD mobile application is its ability to provide users with a wide range of health information. The app offers a "large and diverse" range of medical reference materials, including articles, videos, and medical images. WebMD has got persons praising the tool for its ability to quickly and easily connect them with the information and resources they need (Kontopantelis et al, 2015).

Another strength of the WebMD mobile application is its ability to provide users with personalization. A study by Kontopantelis et al. (2015) found that the app's functionalities allows users to log and track their progress over time, which can be useful for monitoring the effectiveness of a treatment.

However, the WebMD mobile application also has some weaknesses. One weakness is that the app's reference materials have been criticized for providing users with inaccurate or unreliable information. The app’s recommendations were "less accurate" than those provided by a group of board-certified physicians (Fox et al. ,2015). Additionally, the app has been criticized for its tendency to recommend unnecessary medical tests and procedures.

Another weakness of the WebMD mobile application is that it is not designed to replace professional medical advice. A study by Kontopantelis et al. (2015) found that many users of the app "did not understand the limitations of the symptom checker and other features." This lack of understanding can lead to users relying too heavily on the app and neglecting to seek professional medical advice.

Surprisingly the app's adopted user interface was not very friendly to work with, which could affect the application user acceptance and also its usage effectiveness by most patients (Fox et al., 2015)

2.2.2 Medisafe Mobile application

Medisafe is a popular mobile application that is designed to help users manage their medication schedules, including setting reminders for when to take medication and refill reminders for prescription medications. The app also includes a symptom tracker which allows users to log and track their symptoms over time

One of the strengths of the Medisafe mobile application is its ability to help users manage their medication schedules and improve medication adherence. A study by Millstein et al. (2017) found that the app's medication reminders were effective in increasing medication adherence among users with chronic conditions, such as diabetes and hypertension. Additionally, the app's refill reminder feature helps users to stay on top of their medication needs and avoid running out of medication.

Another strength of the Medisafe mobile application is its ability to provide users with a symptom tracker. According to a study by Millstein et al. (2017), the symptom tracker is a useful tool for users with chronic conditions, as it allows them to log and track their symptoms over time. This feature can be helpful for tracking the progression of a condition or monitoring the effectiveness of a treatment.

However, the Medisafe mobile application also has some weaknesses. One weakness is that the app does not provide users with a wide range of health information. According to a study by Millstein et al. (2017), the app's main focus is on medication management and symptom tracking, rather than providing users with a wide range of health information.

Medisafe's user interface was of great concern as its users had challenges in its feature manipulation due to the GUI implementation (Saveski et al, 2018) and lacks a cloud based back up for its data (Santo et al., 2017).

2.2.3 Myfitness pal Mobile App

MyFitnessPal is a popular mobile application that provides users with a comprehensive tool for tracking their fitness and nutrition. The app includes features such as a food diary, calorie tracker, and a library of healthy recipes. Users can also log their exercises, set fitness goals, and access a library of workout videos

One of the strengths of the MyFitnessPal mobile application is its ability to provide users with a wide range of tools to help them track their fitness and nutrition. A study by Rebar et al. (2015) found that the app's food diary and calorie tracker were highly rated by users. Additionally, the app's library of healthy recipes and workout videos provides users with a wide range of resources to help them achieve their fitness and nutrition goals.

Another strength of the MyFitnessPal mobile application is its ability to provide users with personalized recommendations. A study by Rebar et al. (2015) found that the app's food diary and calorie tracker are designed to provide personalized recommendations based on the user's dietary needs and fitness goals. Additionally, the app's library of workout videos is designed to provide users with exercises that are tailored to their fitness level and goals.

However, the MyFitnessPal mobile application also has some weaknesses. One weakness is that the app's food diary and calorie tracker have been criticized for providing users with inaccurate or unreliable information. According to a study by Rebar et al. (2015), the app's food diary and calorie tracker were found to be less accurate than self-reported food diaries and other dietary assessment methods. The study suggests that this may be due to the fact that the app relies on users to manually input their food intake, which can lead to errors in data collection.

Another weakness of the MyFitnessPal mobile application is its lack of integration with other health and fitness apps. A study by Rebar et al. (2015) found that the app does not integrate with other popular health and fitness apps, such as wearable fitness trackers, which limits its ability to provide users with a comprehensive view of their health and fitness data. Coupled by the tool’s inability to quickly and easily track their food intake and calorie consumption made persons not achieve the maximum of health expectations (Krupinski et al., 2016).

2.2.4 Myplate Mobile App

MyPlate by Livestrong is a popular mobile application that provides users with a wide range of tools to help them maintain a balanced diet. The app includes a calorie tracker, nutrient

tracker, and access to a library of healthy recipes. Users can also set diet goals and track their progress towards these goals.

One of the strengths of the MyPlate by Livestrong mobile application is its ability to provide users with a wide range of tools to help them maintain a balanced diet. A study by Huang et al. (2016) found that the app's calorie tracker and nutrient tracker are highly rated by users, with many praising the tool for its ability to quickly and easily track their food intake and nutrient consumption. Additionally, the app's library of healthy recipes provides users with a wide range of resources to help them achieve their diet goals.

Another strength of the MyPlate by Livestrong mobile application is its ability to provide users with personalized recommendations. A study by Huang et al. (2016) found that the app's calorie tracker and nutrient tracker are designed to provide personalized recommendations based on the user's dietary needs and goals. Additionally, the app's library of recipes is designed to provide users with meals that are tailored to their dietary restrictions and preferences.

However, the MyPlate by Livestrong mobile application also has some weaknesses. One weakness is that the app's calorie tracker and nutrient tracker have been criticized for providing users with inaccurate or unreliable information. According to a study by Huang et al. (2016), the app's calorie tracker and nutrient tracker were found to be less accurate than self-reported food diaries and other dietary assessment methods. The study suggests that this may be due to the fact that the app relies on users to manually input their food intake, which can lead to errors in data collection.

Another weakness of the MyPlate by Livestrong mobile application is its lack of integration with other health and fitness apps. A study by Huang et al. (2016) found that the app does not integrate with other popular health and fitness features, which limits its ability to provide users with a comprehensive view of their health and fitness data. The app is slow to servicing user requests degrading its overall performance and acceptability (Fong et al., 2017)

The table below shows a breakdown of the above reviewed application

SN	Application	Features/Deals with	Review(Challenges)
1	WebMD	-Health information -Medical reminder	-Wide range of health information -Personalization -Unfriendly user interface
2	Myplate	-Balanced diet	-No mockups -Unfriendly GUI - Inaccurate information - Lack of integration
3	Myfitness pal	-Fitness and nutrition	- wide range of data . Inaccurate information - Lack of integration
4	Medisafe	-Medical reminder -Symtpom tracker	-No wide range of data -Poor user interface -Long time servicing user requests

Table 1:Existing sytems comparison

2.3 Drawbacks of existing smart health mobile applications

Smart health cross-platform mobile applications, also known as cross-platform mHealth apps, are mobile applications that are designed to work on multiple mobile operating systems, such as iOS and Android. These apps offer a number of benefits, including increased accessibility and the ability to reach a larger user base. However, despite their potential benefits, crossplatform mHealth apps also present a number of challenges.

One of the main challenges of mHealth apps is their lack of regulation. Unlike traditional medical devices, mHealth apps are not subject to the same level of oversight by regulatory bodies such as the Food and Drug Administration (FDA). This can lead to a lack of quality control and the potential for inaccurate or unreliable information to be disseminated to users (Kansagara et al., 2019). Additionally, mHealth apps may not be held to the same standards of privacy and security as traditional medical devices, which can put user data at risk of being compromised (Ng et al., 2018).

Cross-platform mHealth apps is their lack of consistency across different operating systems. This can lead to issues such as different user interfaces, different feature sets, and different

levels of performance on different devices (Kullberg et al., 2018). This can make it difficult for users to navigate the app and for healthcare professionals to recommend specific apps to patients.

Lots of complexity in cross-platform development, as it requires developers to have knowledge and experience in different programming languages and frameworks. This can lead to increased development time and costs (Kullberg et al., 2018). Additionally, cross-platform development can also lead to issues such as lack of access to platform-specific features, which can limit the functionality of the app.

Another challenge is the potential for different versions of the app to have different levels of security and privacy, which can put user data at risk. This can be particularly problematic when working with sensitive medical information (Ng et al., 2018).

Cross platform mHealth apps lack standardization. Many mHealth apps are developed by small companies or individual developers, which can lead to a lack of consistency in terms of functionality and the information provided. This can make it difficult for users to compare different apps and for healthcare professionals to recommend specific apps to patients (Kullberg et al., 2018).

Additionally, mHealth apps may not be accessible to all individuals. Factors such as cost, lack of internet access, and difficulty with using technology can limit the use of mHealth apps among certain populations, such as older adults or low-income individuals (Kansagara et al., 2019).

Furthermore, cross-platform mHealth apps may not be able to provide personalized experience to users. They may not be tailored to the specific needs of certain populations, such as individuals with chronic conditions or disabilities, which can limit their effectiveness (Saveski et al., 2018).

Cross-platform mHealth apps may not be able to provide a seamless integration with other devices or platforms, which can limit their utility and functionality (Kullberg et al., 2018).

Lastly, mHealth apps can lead to over-reliance on self-diagnosis and self-treatment which can be dangerous if not checked by a medical professional. It's important for users to understand that mHealth apps are not a substitute for professional medical advice and should not be used to make important medical decisions (Fong et al., 2017).

2.5 Importance of Smart Afya cross platform mobile application

Cross-platform mobile applications that include features such as medical reminders, fitness tracking, and diet tracking have become increasingly popular in recent years. These apps offer a number of benefits that can help individuals to better manage their health and wellness.

One of the main benefits of cross-platform mobile applications that include medical reminders is that they can help individuals to stay on top of their health care needs. Medical reminders can be set for things like taking medications, scheduling doctor's appointments, and getting lab work done. This can help to ensure that individuals receive the care they need in a timely manner and can help to prevent health complications (Kullberg et al., 2018).

Another benefit of cross-platform mobile application that include fitness tracking is that they can help individuals to stay active and healthy. These apps can track things like steps taken, calories burned, and distance traveled. They can also provide users with personalized workout plans, which can help to ensure that individuals are getting the right amount of exercise for their needs (Kullberg et al., 2018).

Cross-platform mobile applications that include diet tracking are also beneficial as they can help individuals to make healthier food choices. These apps can track things like calories consumed, macronutrient intake, and water intake. They can also provide users with personalized meal plans, which can help to ensure that individuals are getting the right amount of nutrients for their needs (Kullberg et al., 2018).

In addition, having all these features in one application can help to simplify the process of managing one's health and wellness. Instead of having to use multiple different apps for different needs, individuals can have all of the information they need in one place (Kullberg et al., 2018).

Furthermore, cross-platform mobile applications are accessible to a larger user base, as they can be used on multiple mobile operating systems. This can help to ensure that individuals from all backgrounds and with different technology access can benefit from these features (Kullberg et al., 2018).

In conclusion, cross-platform mobile applications that include features such as medical reminders, fitness tracking, and diet tracking are important tools for managing one's health and wellness. They can help individuals to stay on top of their health care needs, stay active and healthy, and make healthier food choices. Furthermore, having all these features in one

application can simplify the process of managing one's health and wellness and increase accessibility to a larger user base.

2.6 Technologies for Cross platform mobile application

There are several technologies that are commonly used to develop cross-platform mobile applications. These include:

1. React Native: Developed by Facebook, React Native is an open-source framework that allows developers to build mobile apps using JavaScript and React. It allows developers to reuse code across iOS and Android platforms.

2. Xamarin: Xamarin is a Microsoft-owned platform that allows developers to build mobile apps using C# and the .NET framework. It allows developers to share code across iOS, Android, and Windows platforms.

3. Flutter: Developed by Google, Flutter is an open-source framework that allows developers to build mobile apps using the Dart programming language. It allows developers to create high-performance apps that are compatible with both iOS and Android platforms.

4. PhoneGap: PhoneGap is an open-source framework that allows developers to build mobile apps using HTML, CSS, and JavaScript. It allows developers to create apps that can run on multiple platforms with a single codebase.

5. Ionic: Ionic is an open-source framework that allows developers to build mobile apps using web technologies such as HTML, CSS, and JavaScript. It allows developers to create apps that can run on multiple platforms with a single codebase, and it is built on top of AngularJS and Apache Cordova.

2.7 Misunderstood diets

The first is the paleo diet is a diet with no rice, processed foods, sugars, dairy and cereals. Second, diet atkins is a diet that only consumes foods containing protein and fat, and avoids foods containing carbohydrates. Third, a vegetarian diet is a diet that only eat fruits and vegetables only at the right rate. Fourth, raw food diet i.e. food and beverages not processed or cooked ie based on plants and organic matter.

2.7 Summary

With regards to the subject matter of this study, there are loop holes associated with the aforementioned related works.

- The systems are mostly designed to be platform dependent i.e. it is either compatible with the Android OS or the iOS. If the system is designed to run on the Android OS this implies that iOS users cannot use the application and vice versa.
- For most designs that were investigated by this study, it was surprisingly found that the adopted user interface was not very friendly to work with, which could affect the application user acceptance and also its usage effectiveness by most patients.
- For some design there is no option to modify the timing and the notifications are mostly compulsory in which some users could find not comfortable to work with.
- For some system design, there is an adoption of built in default alarm tone that boozes and no option for users to modify.
- These systems suffer from lack of regulation and standardization, lack of accessibility, and over-reliance on self-diagnosis.
- Some of the systems presented necessitate a special hardware that is needed to be procured while other applications need a lot of hardware processing power to function well.
- Finally, the systems are also affected by lack of consistency across different operating systems, complexity of cross-platform development, lack of access to platform-specific features, and lack of personalized experience for users.

The existing systems give limited attention to accessibility, user-friendliness, and anonymity of information. This project has developed a system that is accessible to all users who have access to the internet and keeps the anonymity of user information but also maintains the checklists provided by WHO for effective article surfing.

CHAPTER THREE: SYSTEM ANALYSIS AND DESIGN

3.1 Introduction

In this chapter, we present the methodology employed to accomplish the project objectives. The Software Development Life Cycle (SDLC) served as the guiding framework for the design, development, and maintenance of the software. The SDLC encompasses a series of steps, phases, milestones, and the iterative evolution of the software development process. We carefully considered and utilized different models, such as the spiral model, rapid development model, evolutionary model, waterfall model, prototyping model, among others, during the software design and development stages. By adopting the prototype approach, I aimed to ensure the systematic and efficient progression of the project towards its successful completion.

3.2 Methodology

Methodology as defined by the IEEE is a systematic approach to conducting a particular phase such as analysis, design and testing. It consists of guidelines, activities, techniques and tools based on the SDLC.

System development methodology can therefore be defined as the collection of procedures, techniques, tools and documentation that helps the system analysts, designers and developers to implement the system.

In this specific project, the prototype model was implemented as the chosen approach. This model involves the rapid development of a functional yet incomplete version of the information system. The purpose of the prototype is to mitigate risks by creating a quick and basic replica or mock-up of the intended system. It serves to demonstrate technical feasibility, particularly when there are high technical risks involved. Additionally, the prototype model aids in gaining a better understanding of user requirements by eliciting feedback and input from users. By employing this model, the project aims to minimize risks and costs by enhancing comprehension of proposed solutions before allocating further resources. Notably, this model emphasizes the utilization of user feedback to design the most suitable user interface, ensuring that it effectively meets user goals, satisfaction, and promotes system usage. Furthermore, the use of software prototypes facilitates exploration of usability, usefulness, and acceptability by the development team.

3.2.1 Prototyping Model

Below is a diagrammatic representation of the how the Prototyping method can be applied alongside other SDLC stages:

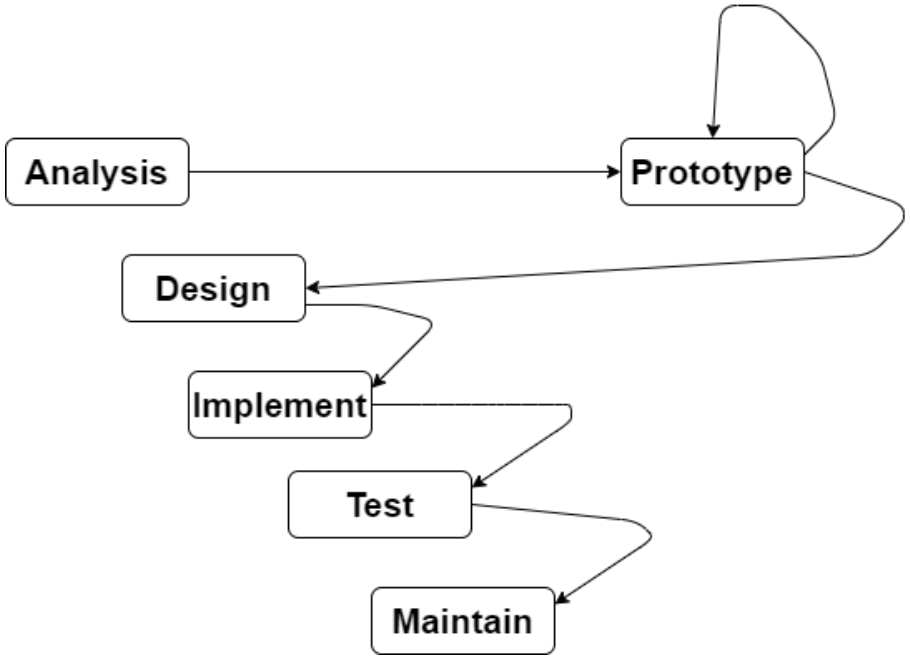


Figure 3:Prototyping model

A prototype is an artifact that approximates a feature of a system or product. The methodology is broken down into five phases of SDLC, however for the prototyping model this would include a sixth phase. The methodology begins with a system analysis mostly referred to as the planning and analysis. Here the business requirements are analyzed by a domain expert who interacts with the stakeholders and various users to gather their requirements to formulate a requirements analysis. The aim of this phase is to ensure quality assurance feasibility of the project and identification of potential risks to be addressed for the success of the project.

After generation of functional requirements in the designing phase commences. Lead developers and technical architects construct the initial high-level design strategy for the software and system throughout the design phase. Delivering the requirements needed to build the Design Document Specification is included in this (DDS). This paper includes information on new transactions to be established, database tables to be added, security procedures, and hardware and system requirements. This is done while prototyping, where a prototype is built, tested and then reworked on as necessary until a desirable feedback from the intended users is

achieved from which the full system can be built from. The prototyping process itself has a series of steps. The steps for the prototyping model are as follows:

1. The new system requirements are defined in as much detail as possible. This usually involves interviewing a number of users representing all the departments or aspects of the existing system.
2. A preliminary, simple design is created for the new system.
3. A first prototype of the new system is constructed from the preliminary design. This is usually a scaled-down system, and represents an approximation of the characteristics of the final product.
4. The users thoroughly evaluate the first prototype and note its strengths and weaknesses, what needs to be added and what should to be removed. The developer collects and analyzes the remarks from the users.
5. The first prototype is modified, based on the comments supplied by the users, and a second prototype of the new system is constructed.
6. The second prototype is evaluated in the same manner as was the first prototype.
7. The preceding steps are iterated as many times as necessary, until the users are satisfied that the prototype represents the final product desired.
8. The final system is constructed, based on the final prototype.
9. The final system is thoroughly evaluated and tested. Routine maintenance is carried out on a continuing basis to prevent large-scale failures and to minimize downtime.

This can be summarized by the diagram below:

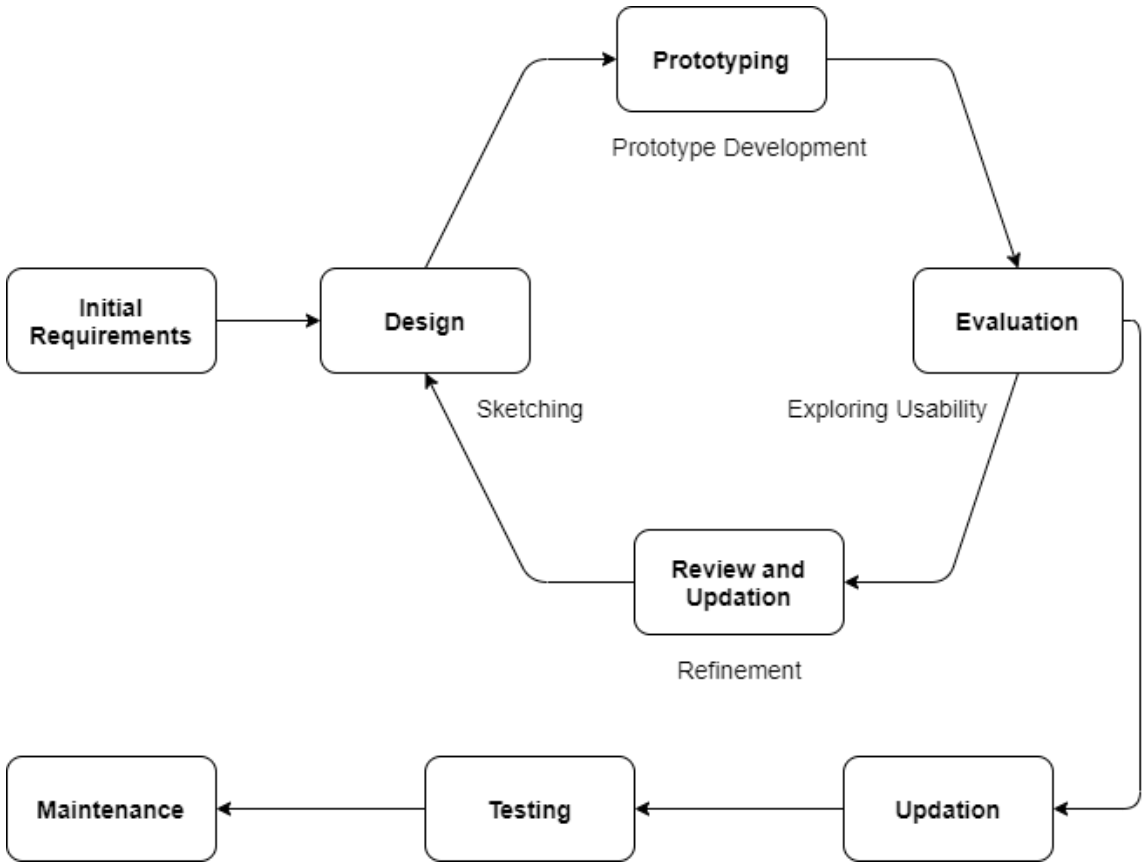


Figure 4:Phases of a prototype model

The motivation to developing this system using this model is mainly because of the following many benefits that positively impact the development of an interactive system:

- Early client involvement in the development of the product boosts consumer happiness.
- Errors and missing functionality are quickly found.
- Prototypes can be applied to more challenging undertakings in the future.
- It places a focus on cooperative teamwork and adaptable design techniques.
- Users are more knowledgeable about how the product operates.
- Consumer demands can be better understood through quicker customer feedback.

The final phases of the SDLC are the implementation and programming of the system where the database administrator creates and imports the appropriate data into the database during this stage. Requirements define the languages used in programming. The interface is built by developers who also perform unit testing in accordance with the coding standards. This stage is crucial for developers. If the business analyst makes any modifications, they must be adaptable and open-minded.

After the system is fully developed then it is tested to ensure that the software is meeting the needs identified and addressed during the planning and analysis phase, testers compare the program to the specifications. Unit tests, integration tests, system tests, acceptance tests, and non-functional tests are all carried out as functional tests.

Finally, once the system has been deployed it is in maintenance mode in a live software environment for post-production. Issues will arise regardless of the software's complexity, user base, or thorough QA testing. That's how software works, with data management, integration, security, and practical application. Regular maintenance and being up to speed on improvements are equally important, as is having access to educated, trustworthy support resources.

3.3 Feasibility Study

Feasibility Study is a preliminary investigation that examines project viability that is, the likelihood the system will be useful to the organization and being developed for the organization. The main objective of the feasibility study is to test the Technical, Operational and Economical feasibility for adding new modules and debugging old running system. All system is feasible if they are unlimited resources and infinite time. There are aspects in the feasibility study portion of the preliminary investigation:

1. Technical feasibility
2. Operational Feasibility
3. Economic Feasibility

3.3.1 Technical Feasibility

The technical issues being raised during this feasibility stage of investigation includes the following:

1. Does the necessary technology exist to develop the proposed solution?
2. Do the proposed equipment’s have the technical capacity to handle required data to be used for the system?
3. Will the system be able to support adequate responses to inquiries, regardless of number of requests or user locations?
4. Will the system be upgradable?
5. What are the technical guarantees of accuracy, reliability, ease of access and data security?

The proposed system is technically feasible as it has a web-based user interface for auditing of workflow through the webs integrated Secure Infrastructure Implementation System thus providing easy access to users.

The database's purpose is to create, establish and manage workflows among the various entities to facilitate all the concerned users in their various capacities and functions in the system. Permission to users would be granted based on the roles specified to their modules. Therefore, this provides a guarantee of accuracy, reliability and security.

The hardware and software requirements for the development of this project are not many and are readily available in-house or are available as free as open source. The development similarly can be done mostly with the current equipment and existing software technology.
Necessary bandwidth exists for providing a fast feedback to the users irrespective of the number of users making the requests.

3.3.2 Operational Feasibility

This aims to determine the benefits of the proposed system if the system is turned into an information system. This compared to whether it meets the organizations operating requirements. Several aspects of operational feasibility are taken as important as part of the project implementation. Some of the important issues raised are:

1. Is there sufficient support for the management from the users?
2. Will there be resistance from the user such that the potential benefits of the proposed system are undermined?
3. Will the system be used and work properly if it is being developed and implemented?

This system is targeted in accordance with the above-mentioned issues. Beforehand, the management issues and user requirements are issues to be handled with modularization and thus taken into consideration. Through this each user of the system gets only data that is relevant to their tasks, by making this diversity each user is able to appreciate the level of organization the system brings as compared to manual or other existing system. To achieve all these, proper requirements analysis has to be performed in order to determine the functional requirements for every user. The well-planned design would also ensure optimal utilization of the computer resources and would help in the improvement of performance status.

3.3.3 Economic Feasibility

A system can be developed technically and then deployed but must still be a good investment for the organization. In this study, the development cost in creating the system is evaluated against the ultimate benefit derived from the proposed system. Financial benefits must be anticipated to exceed or equal to the costs.

The system is economically feasible as it does not require additional hardware or software. This added to the advantage that the software being used are open source thus no additiona l license and product activation keys are needed. Furthermore, the use of frameworks like Node.js and also in programming languages increases code re-usability thus the cost of development is also way cheaper making it less expensive to implement more demanding features from the users. There is a nominal expenditure and economic feasibility for certain.

3.4 Requirements Elicitation

This chapter expounds on how the system was developed to operate and how useful it will be to the user through the design diagrams which are illustrated. It also aims to specify the fact-finding techniques that the project applied for requirements discovery.

3.4.1 Reviewing existing CBIS documentations

Documentation review is also another way I used to collect data by reviewing those existing CBIS'. I used document review for evaluation to gather background information. Reviewing existing documents helps you understand the history, philosophy, and operation of the program you are evaluating and the organization in which it operates.

The data gathering technique also helps when you need information to help you develop other data collection tools for evaluation. This is because reviewing existing documents help to better understand the program and organization you are evaluating hence helping you formulate questions for questionnaires (Melbourne, 2017).

3.4.2 Questionnaires

Questionnaires are a useful fact-finding technique to collect information from a large number of users. Users fill up the questions which are given by the system analyst and then give the answers back to the system analyst. To fulfil the requirements of the project's system objectives, the questionnaires clearly define the questions to suit the objectives (Mamoun, 2015). This technique is useful for this project due to the COVID-19 Pandemic that took place during the period of the project's development.

Questionnaires were presented to a sample of the target population and the results were analyzed to understand the user's requirements in relation to the use of a smart health mobile application.

3.5 Data and System Analysis

Before designing a system there must be a thorough understanding of the old system and determine how computers can best be used to make its operation more effective. System analysis, then, is the process of gathering and interpreting facts, diagnosing problems, and using the information to recommend improvements to the system. A key question is, what is the source of the problem?

Analysis is a detailed study of the various operations performed by a system and their relationships within and outside of the system. A key question is, what must be done to solve the problem?

This project proposes to apply a Structured Systems Analysis and Design (SSAD). This is because theoretically the approach allows one to plan, manage and control a project well. These points are essential to deliver the product on time. Structured Analysis is a set of techniques and graphical tools. They allow the analyst to develop a new kind of system specifications that are easily understandable to the user. SSAD separates the logical and the physical systems design. Therefore, the system does not have to be implemented again with new hardware or software. (Gabry, 2017).

3.5.1 Data Analysis

Data analysis is the process of extracting useful information from the given data series, that will be useful in taking important and sound decisions. Data analysis helps use to understand facts, observe patterns, formulate explanations and try out hypotheses.

Proposal/Response	**Positive**	**Negative**	**Undecided**
Ease of use	23	8	9
Make use of mobile interface	30	3	7
Enable users to respond to appeals	15	12	13
Validate user information	34	2	4
Retain useful components of current system into the proposal	39	1	0
Generation of useful reports	36	0	4

Enable user to view timelines and history	10	22	8
Allow users to deregister from the system at will	31	6	2
Total	218	54	47
N = 319			

Table 2: User responses to the Smart Afya cross platform mobile application proposal

Pie chart is a statistical analysis tool used in presentation of information for analysis in the study.

The above data can be summarized by the following data.

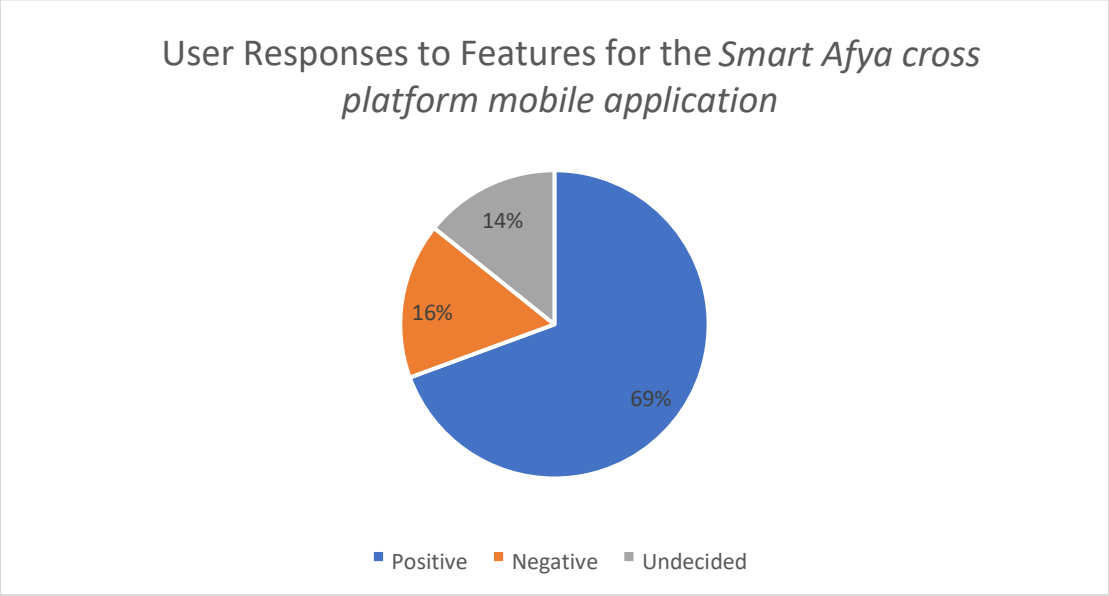


Figure 5: A pie Chart of User responses

From the study, most of the users and the management are in support of the proposed system. Based on the survey the following functional requirements were developed to meet the respondents' needs.

3.5.2 Functional Requirements

Functional requirements serve as the basis for the system to be built. They are requirements given in terms of required operations and/or data. The Smart Afya cross platform mobile application's functional requirements are as listed:

1. The system should allow users to **log in and log out upon registration**.
2. The system should allow its users to **setup and manage reminders**

3. The system should be able to **capture and store, users' details** into the My Structured Query Language (MYSQL) database.
4. The system should be able to **generate pre-defined reports** such as medical reminders over a given time period.
5. The system should be able to allow users **manage notifications.**
6. The system should be able to allow its users to perform **article surfing.**
7. The system should provide **advisories for healthy living based on diseases**.
8. The system should **recommend balanced diet to its users.**
9. The system should **guide through the disease related/age exercises**.
10. The system should allow its **users to file complaints**.
11. The system should **provide help to its users**.

3.5.3 Non-Functional Requirements

(Glinz, 2005) defines non-functional requirements as those which specify how well the system performs its functions i.e. performance requirements and quality requirements.

I. Reliability

The system should be reliable in that when an individual wants to access information from the system, it should be available to the users.

II. Capacity

The storage capacity of the system should be big enough to store all the information needed for a large group for a long period of time that will enable the system to run faster.

III. Role-based security

The system should restrict users to a view based on user profile attributes. This access should be adjustable throughout the service rendering of the application.

IV. Reusability

The System's Database code should be reusable and exported into a library.

3.6 System Specification

3.6.1 Introduction

This section will provide a full description of the system and its users. Then it depicts the functional and non-functional requirements that have been collected using several methods like brainstorming, interview and e-surveys. After determining the most important requirements, requirement analysis was adopted using several tools such as use-case diagram, sequence diagram and activity diagram.

3.6.2 Technical Specification

The following are the technical specifications of the smart health cross platform mobile application:

- Portable
- Help facility
- Compatibility
- Virtual Interaction
- Persistency
- GUI Human Computer Interaction
- Security

a) Portable

The deliverable of the project is a product that can be sent/shared by the users on different sharing platforms. They include Social Media applications, Bluetooth, Files, Xender, Flash share, iShare etc.

b) Help facility

The application includes a content refining help facility that shall be used to aid its users whenever stack while operating the application. This promotes the ease in operating the application. This help facility shall be objective oriented.

c) Compatibility

The application is compatible with different hardware devices (hardware compatibility) and mobile operating systems (software compatibility). This guarantees that with the change in running environment the app shall remain to be consistent regardless.

d) Security

The application shall employ user verification to ensure privacy of the medical records and use's data. The whole system shall be run on encryption between the user requests and the system's response. It offers cryptographic functionality that includes a set of wrappers for OpenSSL's hash, HMAC, cipher decipher, sign and verify functions.

e) Virtual Interaction

The mobile application allows its users to interact with it with virtual means. Its users shall use touchscreen technology for navigation. Users are also able to feed data into the system virtually by use of virtual keyboard. The running platform includes a coordinate reference operable to define a coordinate system with respect to a physically defined reference position.

f) GUI Human Computer Interaction

The system shall employ the use of Graphical User Interface that shall enable its users to interact with it harmoniously. GUI comes with advantages such as multitasking, ease in communication and use, attractiveness among others.

g) Persistent

The systems shall be designed such that it shall continue to exist or occur over a prolonged time amid the changes in technology and its variances.

Additionally, system shall be able to continue firmly or obstinately in an action in spite of technical interference.

3.6.2 System Description

3.6.2.1 User Roles Management and Authentication

The module helps in role management by managing authorization which enables specifying the resources that users in the developed system are allowed access to. The module verifies that the user is who they claim to be by checking the provided username and password against what is stored in the database.

3.6.2.2 Surfing module

The user of the mobile application is presented to him/her health articles with which are expected to be personalized. To achieve the full potential of the functionality users, require

internet connection for surfing. However, they are not subjected to authorization and authentication to use the functionality.

The application incorporates the WIKIPEDIA API for connection to maximize article surfing

3.6.2.3 Medical Reminder Module

The medical reminder module is a core functionality of the system. To access the module the users shall be required to first log in for authorization as the data in this module includes medicine name, notification message, disease name etc. All the inputs in this module are user dependent. A user is able to specify the time period for the intended reminder as well as the frequency of take of the medicine and the audio tone. The module allows users to manage the reminders.

3.6.2.4 Fitness Module

The developed system seeks to provide patients with important information in form of guides, video materials on the different exercises. Once a user enrolls to a programme the system shall be able to track his/her progress by use of a counter.

3.6.2.5 Diet Module

This module shall be recommending to users the different balanced diet in sync with their considerations. The diets shall include palio diet, vegetarian diet etc. The system shall also be able to avail healthy practices to its users.

Other requirements may include the Hardware and software requirements as developed from the feasibility report.

3.6.3 Proposed Hardware requirements

Computer or a laptop with the following specifications:

1.At least 8GB RAM (Giga Byte Random Access Memory)

This will allow for faster loading of the testing platforms.

2. 2.0 GHZ (Giga Hertz) processor speed

During the testing phase of the codes written, this will facilitate faster processing of the requests by the application as well as speed up responses for the earlier given requests.

3. **At least 40GB hard disk capacity**

This is to provide the storage space for the application's workspace where the various codes that will be written will reside.

4. **External Memory (External Hard disk) approximately 20GB**

This is important in the project as it will enable for the process of backup for the purpose of security in case of system crash

5. At least one NIC (network interface card) for connection purpose.

3.6.4 Method used to test the developed system

This project tests the developed system using the functional testing method. Functional testing covers how well the system executes the functions it is supposed to execute, including user commands, data manipulation, searches and business processes, user screens, and integrations. Functional testing covers the prominent surface type of functions and the back-end operations (such as security and how upgrades affect the system).

3.6.5 Domain of Execution

The solution is executed through both mobile only. To achieve a convenient mobile experience Flutter library are going to be used to define responsive sites that can also be browsed on mobile browsers connected to the web server. The system has a back-end mobile application because it allows any user to access the system through any mobile device that has access to the internet.

3.6.6 System Architecture

The system architecture of Smart Afya cross platform mobile application is illustrated below shows the interaction between the system components.

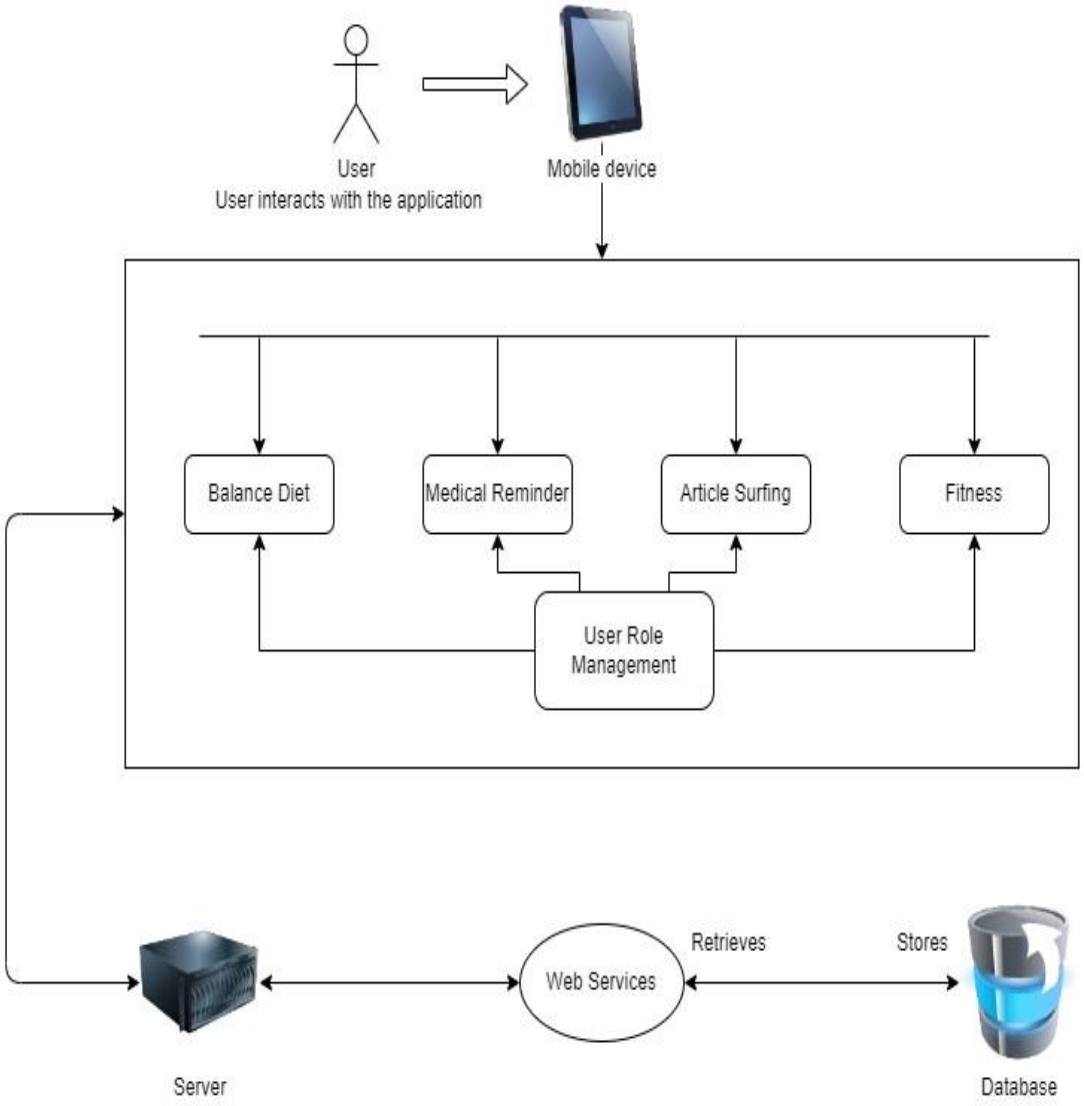


Figure 6: System Architecture

The application's functionality is accessed by the user through a mobile phone, providing a platform for interaction. To facilitate database access, the application communicates with a web server, utilizing web services. Moreover, the application includes a user role management feature that allows for the management of various personalized user roles based on the configured settings. This feature enables the application to cater to the specific needs and permissions of different users, ensuring a personalized and tailored experience.

3.7 Diagrams

3.7.1 Modular representation

The Smart Afya cross platform mobile application achieves its purposes or functionalities through four modules. The Medical reminder module, Diet module, Fitness module and the Article surfing module.

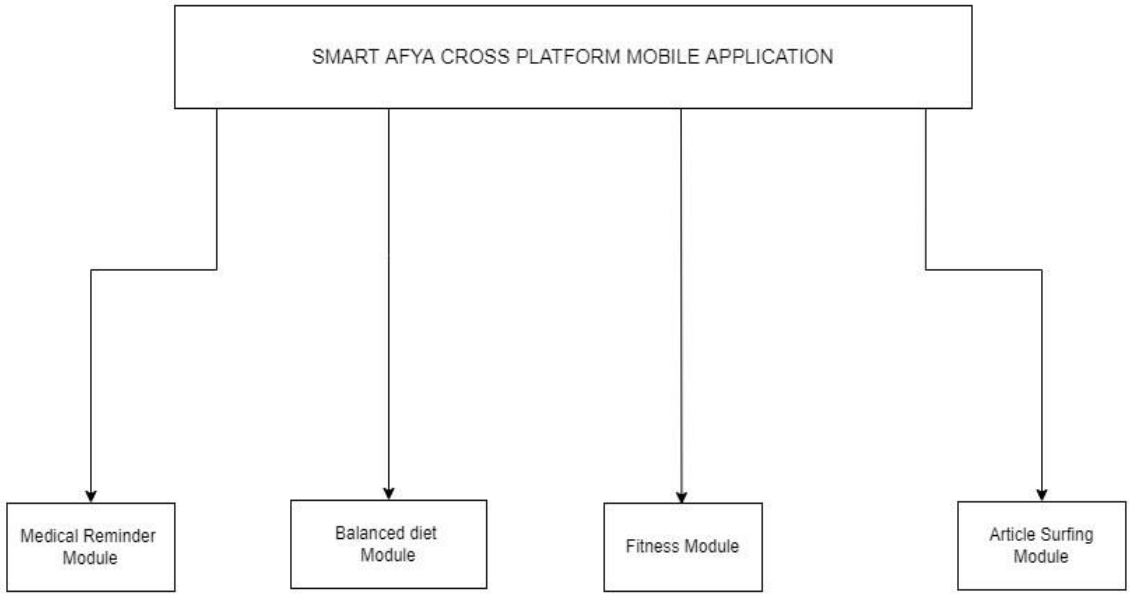


Figure 7: Modular Representation

3.7.2 Use case Diagram

In this section, the analysis of both functional and non-functional requirements is conducted using use case diagrams and detailed use case descriptions. These diagrams and descriptions provide a comprehensive understanding of the system's intended functionality and how different actors interact with it. The project's ultimate deliverable is a single-user system, designed to cater to the needs of a single user, ensuring a personalized and seamless user experience.

The project’s deliverable is a single user system.

3.7.2.1 User interacting with the Medical Reminder Module

As the system is a single user system we only have single actors. In this module the user is expected to log in so as to access the functionality. Once logged in a user can add a new reminder, remove reminder, modify a reminder, activate reminder and check prescriptions. The user once done with accessing the reminder service offered can sign out of the system.

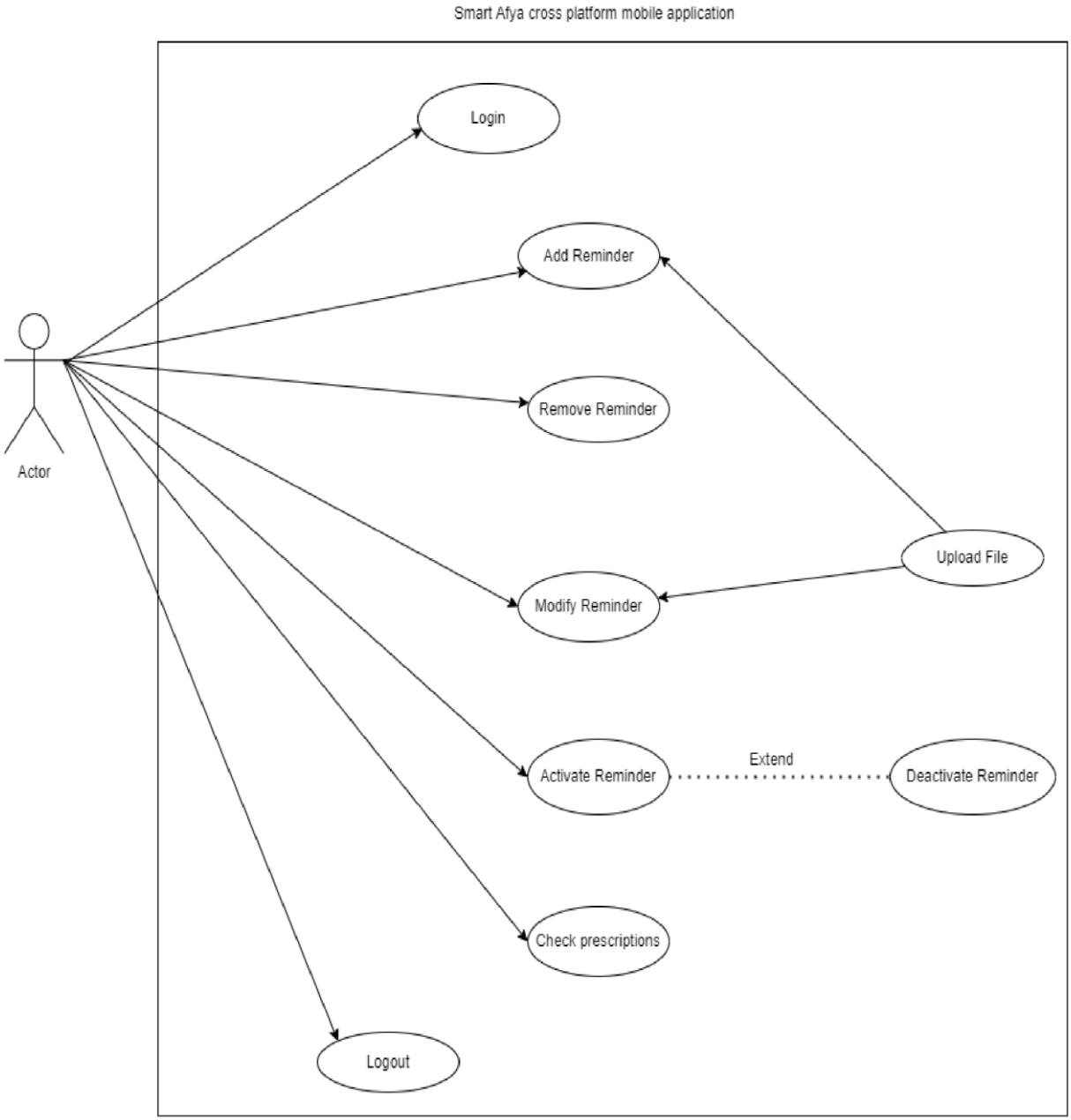


Figure 8: User case on Medical Reminder module

As shown by the above user case diagram when a user activates a reminder he/she can also deactivate it. Furthermore, for a reminder to be added or modified the application interacts with a file that is uploaded for change configuration.

3.7.2.2 User interacting with the Article Surfing Module

For the user to access this module's functionality the user is not expected to sign in to the application. However, when a user signs in the articles could be personalized as well as the module's surfing functionality.

A user upon opening the app is welcomed by this module. The module offers global health related information and provides the user once connected to the internet the surfing capability.

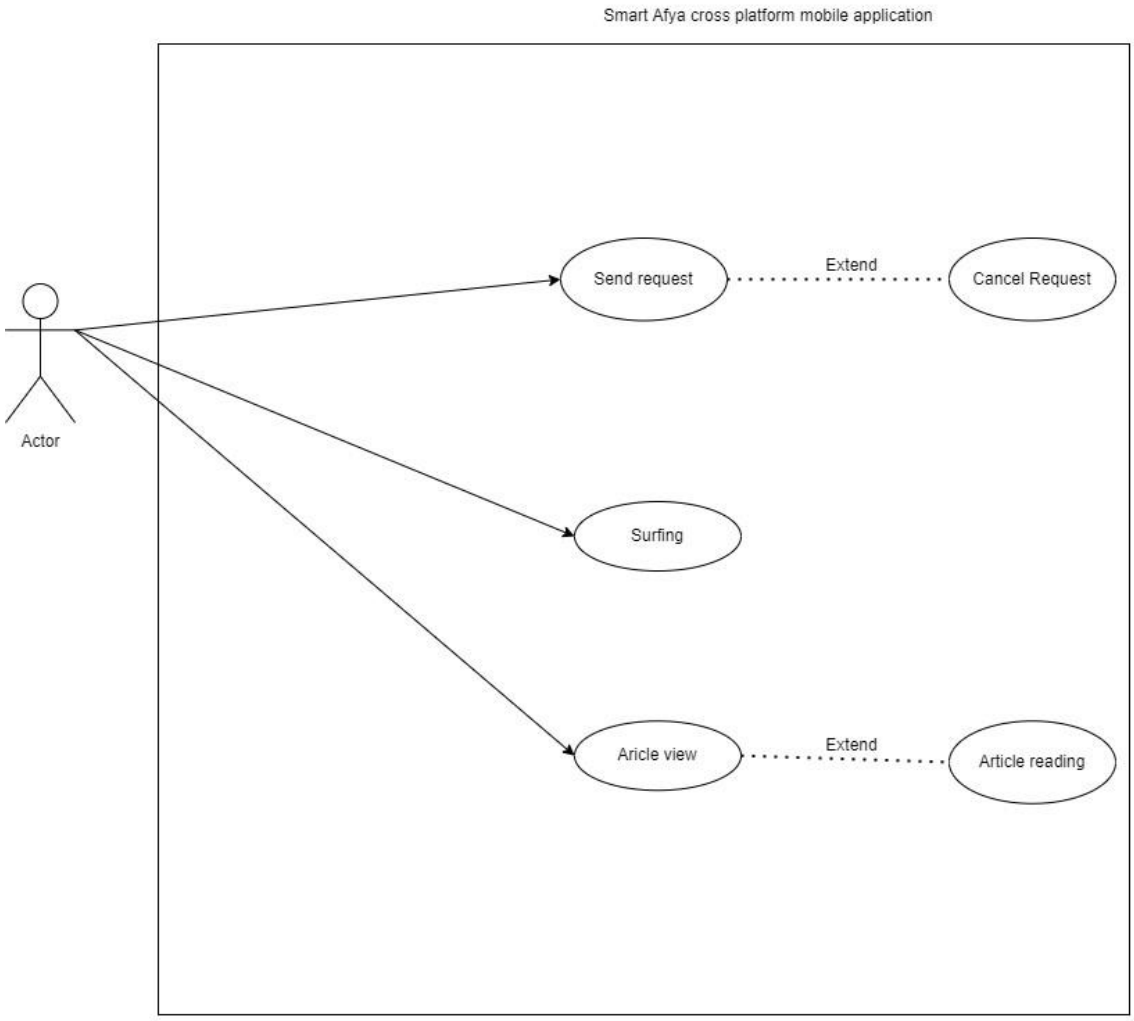


Figure 9: User case on Article Surfing Module

The user initiates a request to the web server and can as well cancel the request. Upon sending the request the web server allows the user to surf and view articles. The user can as well read the article content upon viewing.

3.7.2.3 User interacting with the Fitness Module

A user must log in to the system in order to access the functionalities on the module. Once logged in a user can enroll to a fitness programme as well us unenroll. The enrolled workouts shall be visible to the user as a personalized functionality. The user shall be provided with mock up resources for the different exercises. The user shall take records, which extends modification/update of records.

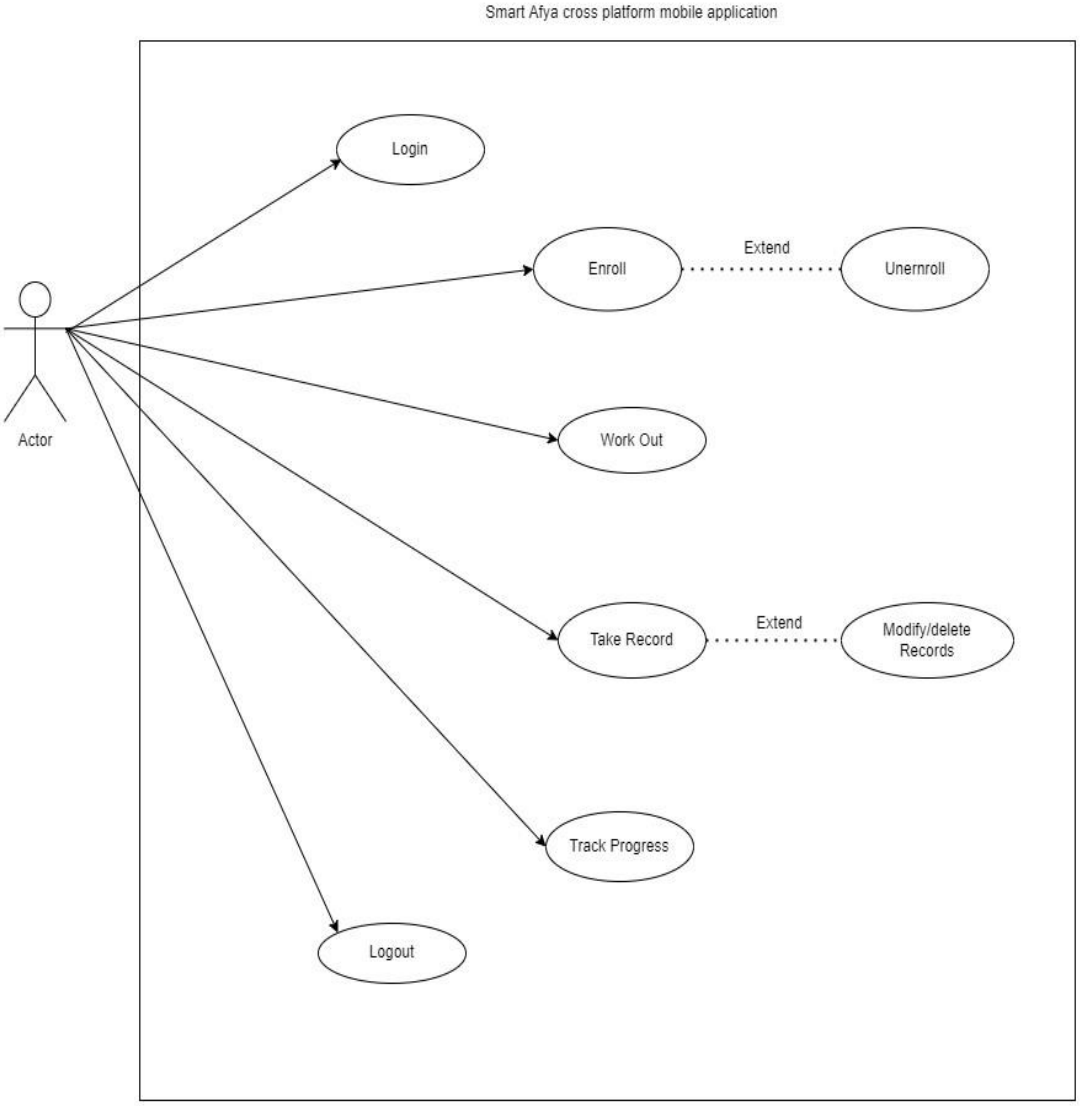


Figure 10: User interacting on Fitness module

The different workouts' progress can be tracked by the user. Once done with the functionality the user can log out of the system.

3.7.2.4 User interacting with the Diet Module

A user must log in to access the module's functionalities. Once logged in the user sends a request to the system for a diet. This functionality can be extended for the user to also cancel the request.

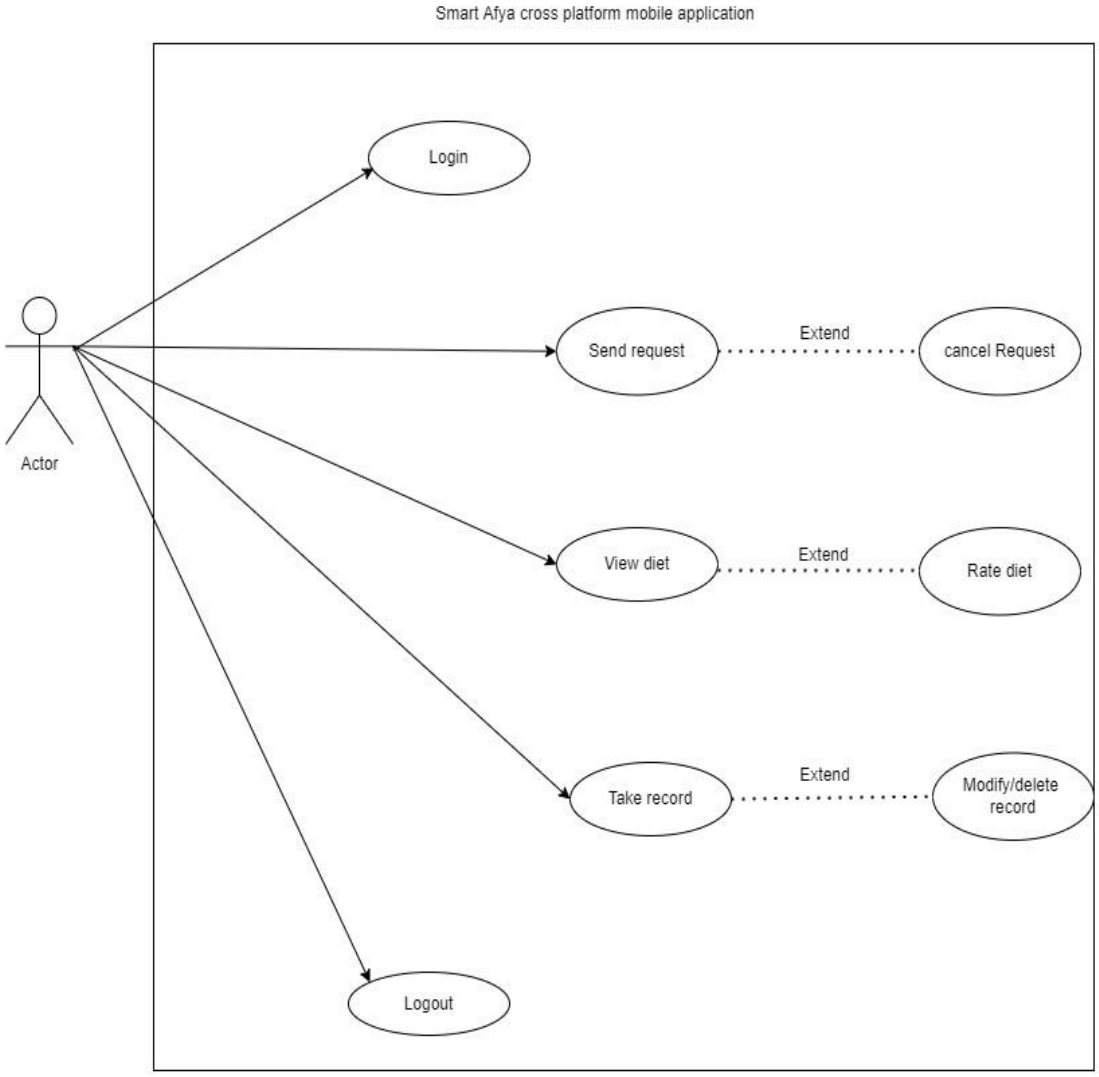


Figure 11: User interaction on Diet Module

Upon servicing the request, the user can not only view the diet but also rate it. The user can take records on the diet. This functionality extends modification or deletion of the record.

Once done with the functionality of the module the user can lo out of the system.

3.7.2.5 User interacting with Report generation

The application generates reports on areas of user's interest except for article surfing.

For this functionality the user logs I to the system. Upon successful authentication and authorization, the user sends request. The functionality extends canceling of the request. Once the request is serviced the user views the report and can also save the report. Once satisfied with the functionality the user can log out of the system.

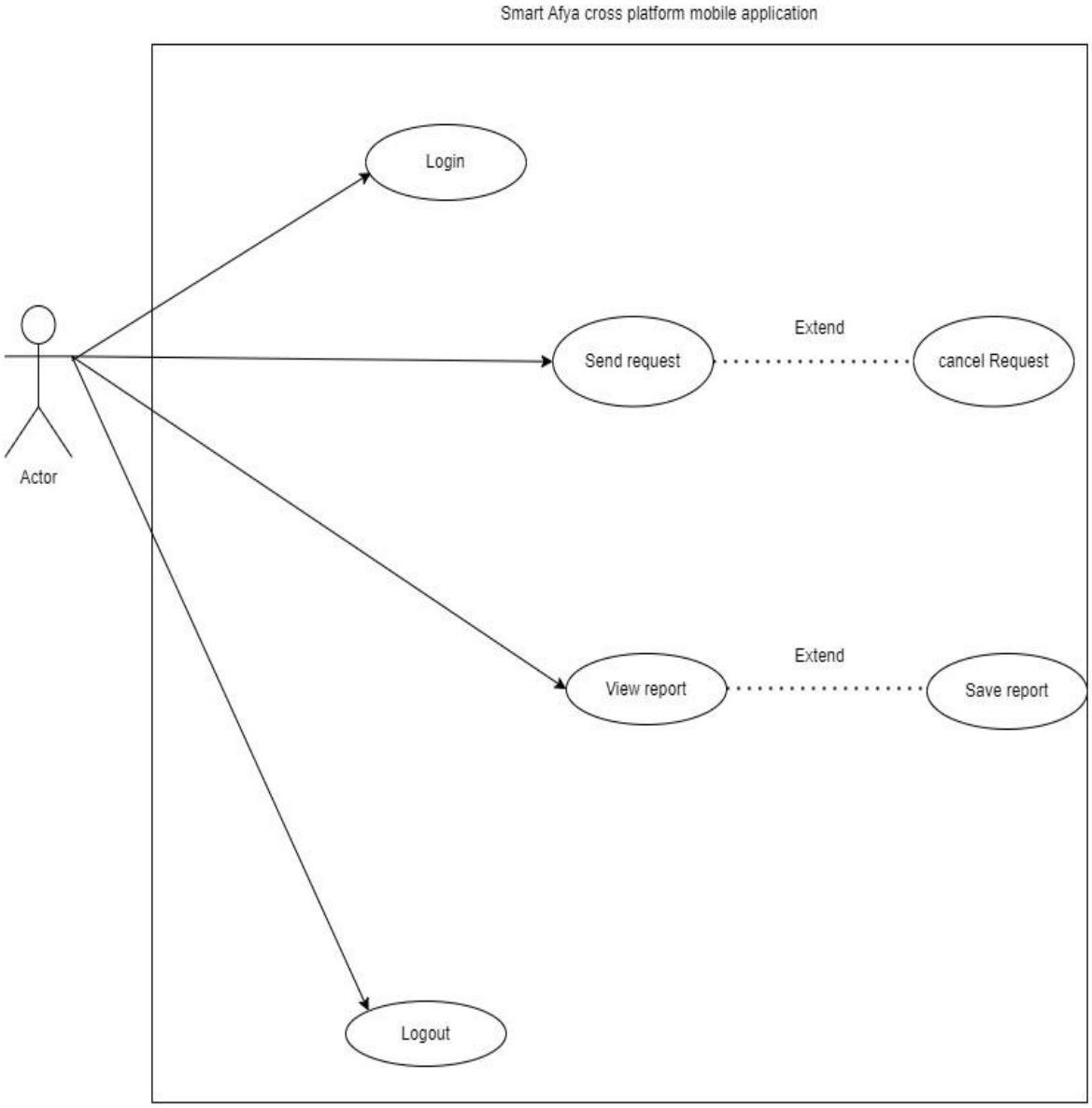


Figure 12: User interaction on Report Generation

3.7.2.5 User interacting with System tool

The system tool is a collection of functionalities that allow sharing of the application, viewing about information, interact with settings preferences and feedback handling.

The about section provides the user with information on other products. Not only will the user be able to share the app but he/she can also rate the app. The rating is an extension of the sharing functionality. The user can also change the preference of the application to a most comforting state of use.

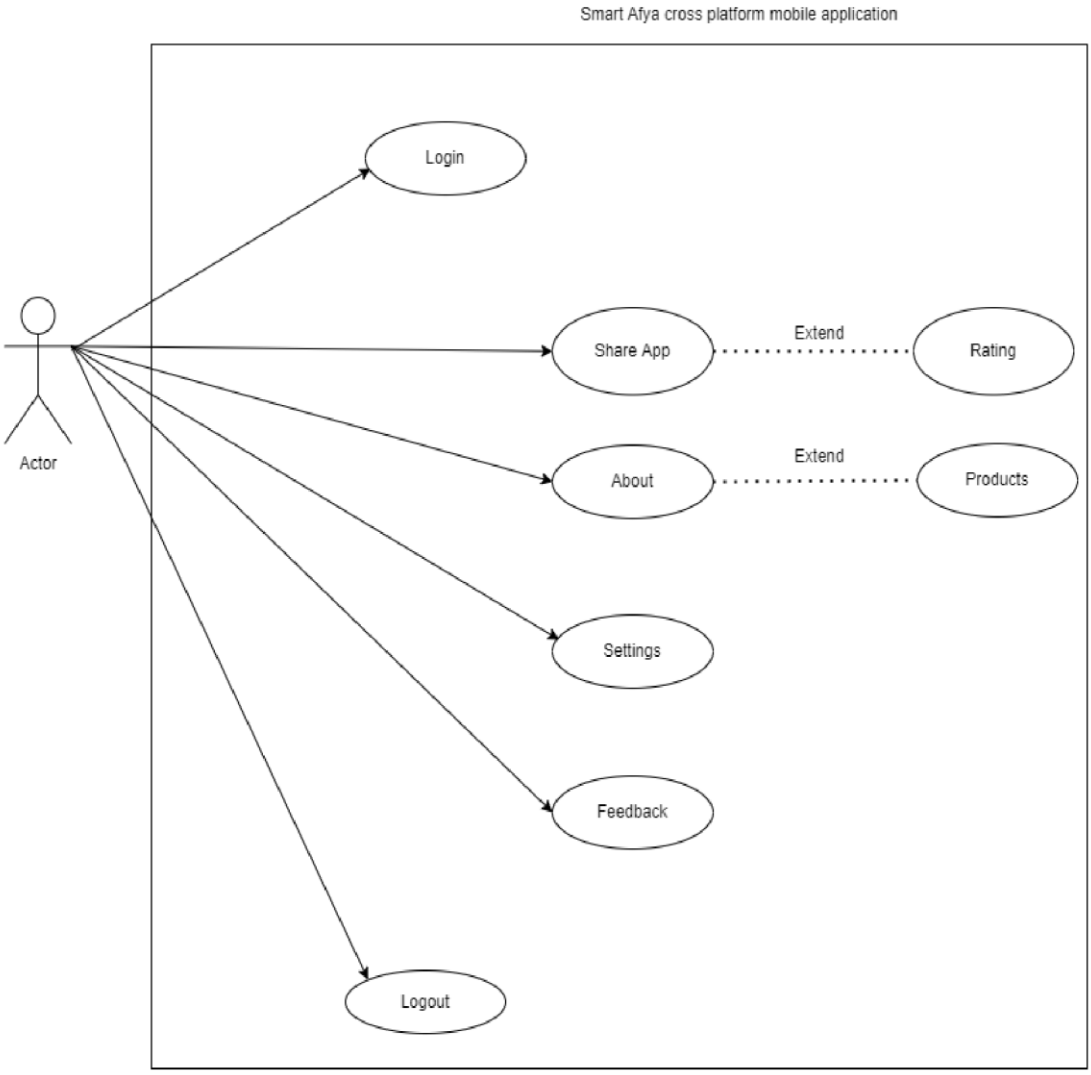


Figure 13: User interaction with Tool functions

:

The user is able to interact with a feedback feature that allows him/her to file complaints, suggestions or recommendations for more consideration in the design.

3.7.2.5 General user interaction with the system

The diagram below summarizes the interaction of the single user system with the system's functionalities to achieve the functional and non-functional requirements of the Smart Afya cross platform mobile application.

It depicts that for medical reminder, fitness and diet functionalities the user must log in to the system. The log in activity includes a signup feature for the new users of system.

However the article surfing module does not require a log in activity.

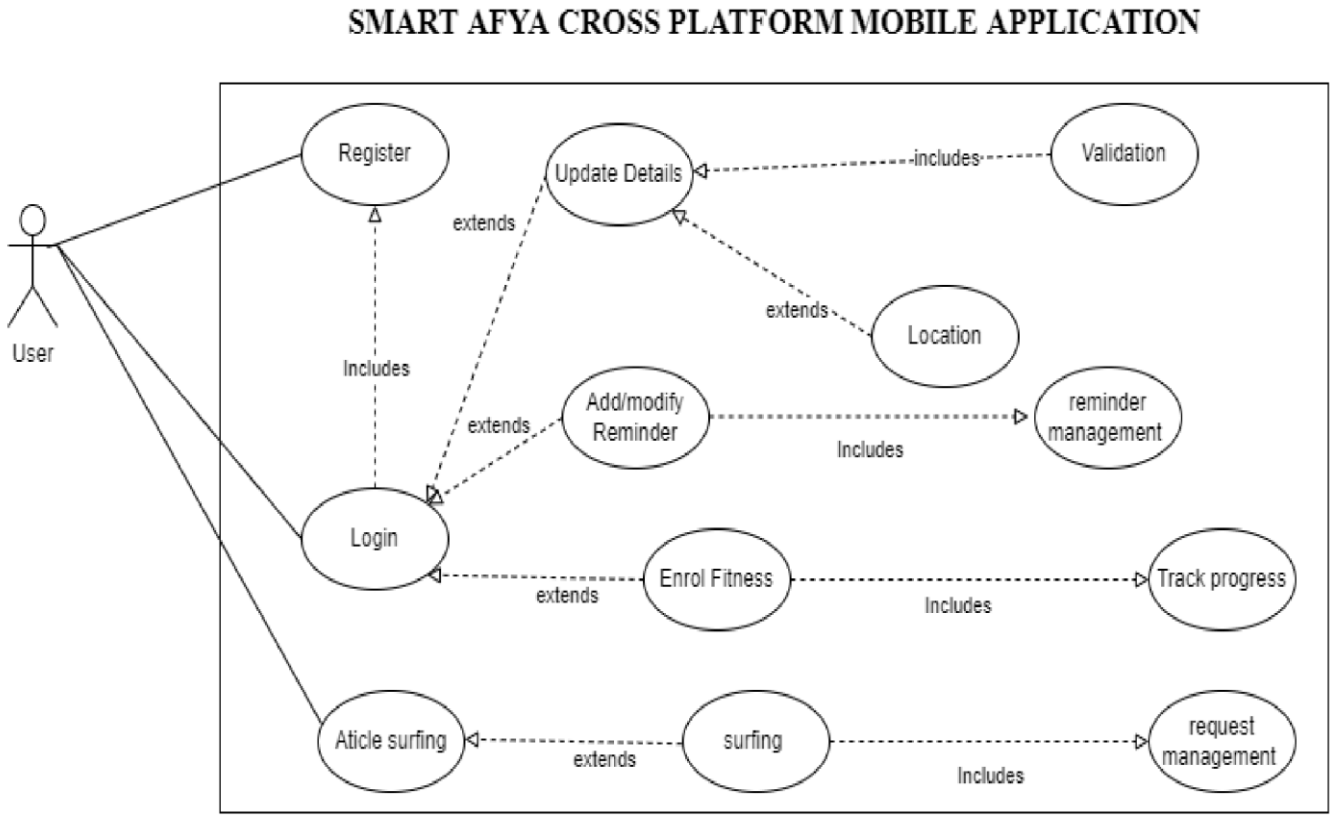


Figure 14: General user interaction with the system

3.7.3 Use Case Description

A use case description provides a detailed narrative of the steps and interactions involved in a specific use case scenario. It outlines the sequence of actions performed by actors (users, systems, or external entities) and the system's response.

3.7.3.1. Registration

Flow of events

1. User wants to be registered
2. Registration form is provided to the user.
3. The user completes the form that contains user's full names, address, email/username, password and other information.
4. The system validates that all fields are filled, email is unique and performs Regex on all inputs for specific pattern matching including password.
5. User submits the form using the submit button.
6. The system registers the details into the database.
7. The use case ends

Use Case Name	Registration
Actor(s)	End User
Purpose	For user Registration
Pre-Conditions	User selected the registration form to fill
Post Conditions	Information is valid. Saved in the database Invalid information. Error message is displayed
Success Scenario	All provided information is valid
User Action	**System response**
User fills the required fields	The system will save the data into the database.

Table 3: Registration use case description

3.7.3.2 Login

Flow of events

1. User wishes to login to the system modules.
2. System provides login form to the user.

3. The user selects module to login, fills the username/email and password fields.
4. The system verifies a match from the database towards the respective selected system.
5. The login is successful, session is started.
6. User interacts with respective module operations.
7. Use case ends.

Use Case Name	Login
Actor(s)	End User
Purpose	For accessing respective module operations.
Pre-Conditions	User selects module, User submits username and password.
Post Conditions	User details are registered in the database, submitted details match in the database, Selected module matches registered user access.
Success Scenario	Successful login and session started, user views Module operations
User Action	**System response**
User selected the respective module	The system shows all module operations.

Table 4: Login Use Case description

3.7.3.3 Reminder setting

Flow of events

1. Login.
2. The login requests are loaded from the database to the approval page.
3. User inputs frequency of take
4. User types in the notification message
5. User activates the reminder/sets reminder times
6. Upon the activation of the reminder the reminder records are updated
7. The changes are saved to the database.
8. The use case ends.

Use Case Name	Reminder Setting
Actor(s)	End user
Purpose	For prescription reminder
Pre-Conditions	User made a request to be approved. Valid details of the user are in the database.
Post Conditions	User medical reminder is set
Success Scenario	Reminder is successfully set and the notification message is sent

Table 5: Reminder setting user description

3.7.3.4 View Requests

Flow of events

1. The user wishes to view the interested appeals in the system.
2. The user selects the requests viewing page.
3. The requests are fetched and loaded from the database to the page.
4. The User views the loaded requests.
5. Use case ends.

Use Case Name	View Requests
Actor(s)	End User
Purpose	For viewing interested requests/ resources.
Pre-Conditions	The user selects the request page. User is logged in.
Post Conditions	The requests are successfully loaded from the database.
Success Scenario	The system will show all the requests to the user.

Table 6: View Requests Use Case description

3.7.3.5 Manage profile

Flow of events

1. The End user wishes to view and update the profile information
2. The officer logs in to the system and proceeds to the profile management page.

3. The user details are loaded from the database to the page.
4. The user performs updates on profile details including edits, deletion and addition of information
5. The updates are saved into the database successfully.
6. Use case ends.

Use Case Name	Manage Profile
Actor(s)	End User
Purpose	For updating the user details.
Pre-Conditions	The user is logged into the system.
Post Conditions	Profile updates are stored in the database.
Success Scenario	Updated user profile successfully
User Action	View, delete, update and add new information
The user has chosen profile management page	The system will update the details in the database.

Table 7: Manage profile Use Case description

3.7.3.6 Surfing

Flow of events

1. User is connected to the internet.
2. The user surfs
3. The request is serviced by mobile web servers and output generated on the screen.
4. Use case ends.

Alternative flow

1. User logs into the system
2. User is connected to the internet.
3. The user surfs
4. The request is serviced by mobile web servers and output generated on the screen.
5. Use case ends.

Use Case Name	Surfing
Actor(s)	End user
Purpose	Make request to web servers for information
Pre-Conditions	User is connected to the internet The user selects the request page.
Post Conditions	The output is displayed. Else the error message will be displayed.
Success Scenario	All new data is validated and saved successfully to the database.
User Action	Select the request page and fill the form.
User fills all the required fields	System saves the data to the database.

Table 8: Surfing user case description

:

3.7.3.7 Diet Recommender

Flow of Events

1. User logs into the system
2. User is connected to the internet.
3. The user goes to the diet module
4. The request is serviced by web servers and output generated on the screen.
5. User views and disease related diet
6. Use case ends.

Use Case Name	Diet rating
Actor(s)	End user
Purpose	View, record and rate a diet
Pre-Conditions	User is connected to the internet The user selects the diet module.
Post Conditions	The output is displayed. Else the error message will be displayed.
Success Scenario	All new data is validated and saved successfully to the database.

User Action	Select the diet module and fill the form
User fills all the required fields	System saves the data to the database.

Table 9: Diet viewing user case description

3.7.3.8 Exercising

Flow of Events

1. User logs into the system
2. User is connected to the internet.
3. The user goes to the fitness module
4. The request is serviced by web servers and output generated on the screen.
5. User views and enrolls for a fitness programme
6. Counter Initiated
7. Use case ends.

Use Case Name	Exercising
Actor(s)	End user
Purpose	View, enroll, record and perform workouts
Pre-Conditions	User is connected to the internet The user selects the fitness module.
Post Conditions	The output is displayed. Else the error message will be displayed.
Success Scenario	All new data is validated and saved successfully to the database. Workout mock ups are visible
User Action	Select the fitness module
User selects the interesting workout	System saves the data to the database.

Table 10 : Exercising user description

3.7.4 Sequence diagram

Describes an Interaction by focusing on the sequence of Messages that are exchanged, along with their corresponding Occurrence Specifications on the Lifelines.

3.7.4.1 Registration and Login sequence diagram

This case scenario encompasses the registration and login processes for all system users, enabling them to access specific modules within the system. The user initiates the registration process by requesting registration through the system, which directs them to the registration page. Here, the user fills in all the required details, which undergo validation using pattern matching regex and comparisons with the database to ensure uniqueness of information such as username and email. Upon successful verification, a new row is created in the database, storing the user's details.

Subsequently, the system loads the details from the database onto the approval page, where the information is validated. If the details are found to be invalid, the user's registration fails, and they are redirected back to the registration page. However, if the details pass validation, the user is successfully registered.

Following registration, the user proceeds to the Login page, where a form is provided for them to enter their login credentials. The system verifies the entered details, such as ensuring the presence of '@' in the email and requiring a minimum password length of 8 characters. The provided details are then matched against the records in the database. If no match is found, the login details are deemed invalid. However, if a match is found, the user successfully logs into the system, and a session is created for them.

Once logged in, the user gains access to the specific module functionalities within the system. They can perform various actions and tasks as per their assigned roles and permissions. When the user decides to end their session, they initiate the logout process, resulting in the destruction of the session and the user being successfully logged out of the system.

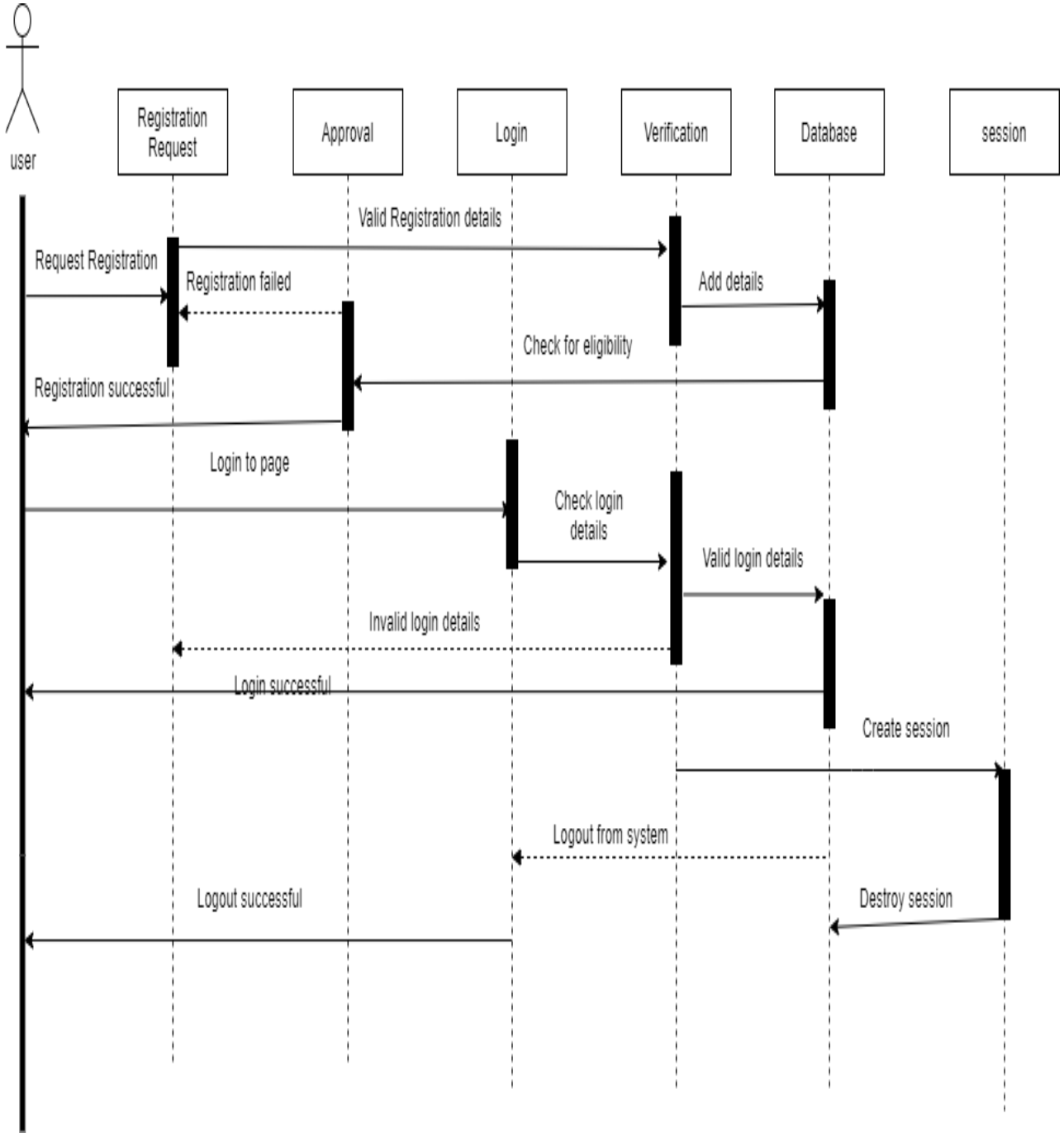


Figure 15: Registration, login, session and logout sequence diagram

The following sequence diagram presents the general sequence of activities by the user as they interact with the cross platform mobile application.

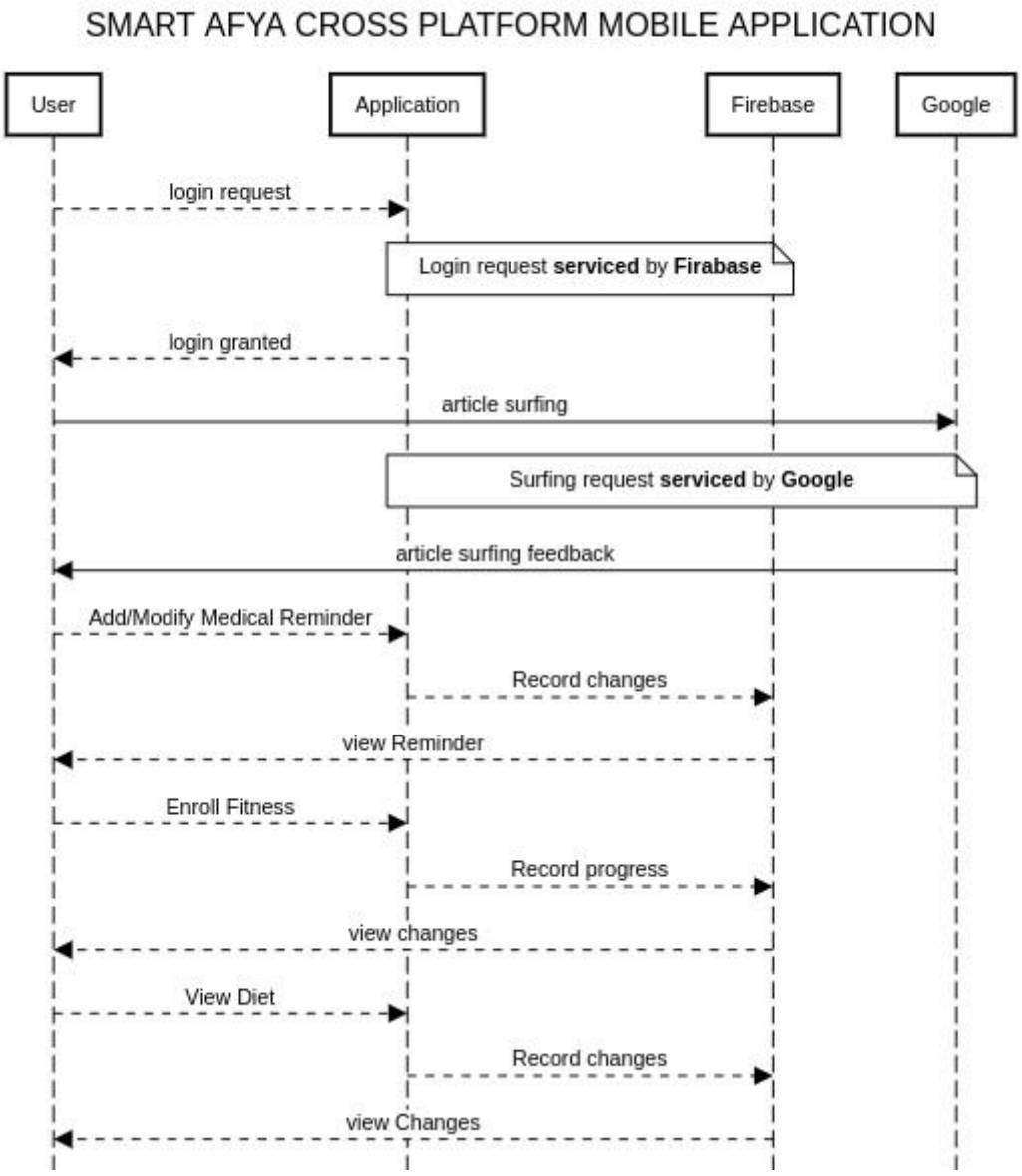


Figure 16: Sequence diagram for the overall system

3.7.5 Context Diagram

The context diagram provides an overview of the system's interactions with its users and depicts the flow of information from left to right, representing the systematic progression within the system. The diagram showcases the relationships between the system and its users, highlighting how they interact with each other.

Typically, the context diagram consists of the system at the center, surrounded by external entities or actors. These external entities represent the users, systems, or organizations that interact with the system. The arrows illustrate the flow of data or information between the system and the external entities, indicating the direction and nature of the interactions.

By reading the context diagram from left to right, you can visualize the sequence and progression of the system's operations. It helps to identify the inputs from users or external entities, understand how the system processes and manipulates that information, and finally, recognize the outputs or responses provided by the system.

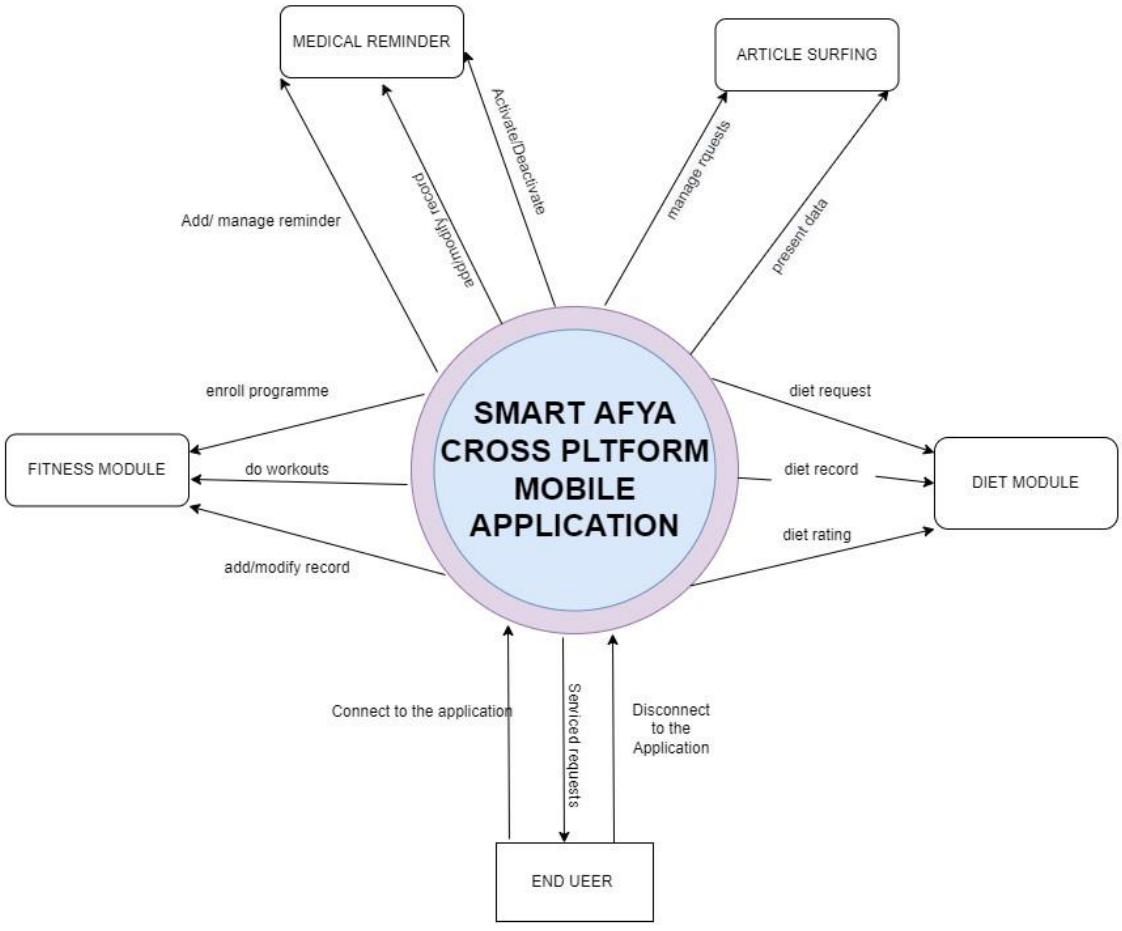


Figure 17 : Context diagram

3.7.6 Activity Diagram

The Activity Diagram describes dynamic aspects of the system. It is basically a flow chart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. So, the control flow is drawn from one operation to another. This serves as the flow charts indicating the necessary steps to be followed by the different users in accessing our system and performing their given functions then saving the changes of any updates during the activity.

All the users of system will first have to login to the system, the system then authenticates the users; details are matched against those in the database for the authentication, upon success the system assigns session to the logged in user else the user is taken back to the login process page. The user then can now perform any or all the functions as allowed for that particular module. Once all changes and modifications are saved, the user can then end the session. The next time the user logs in the system the session is restored and the user resumes from the last changes made.

This description can be illustrated by the flow chart below:

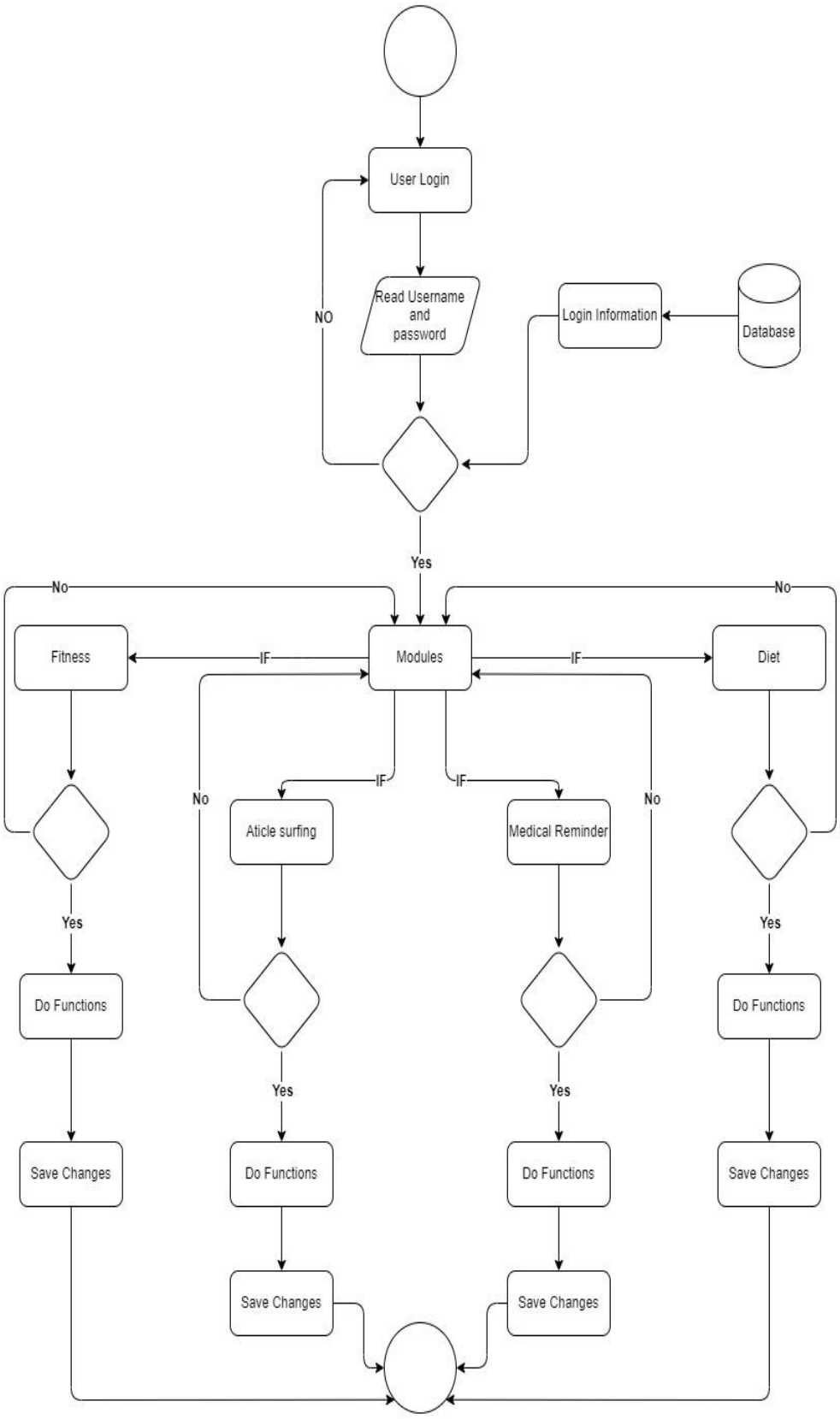


Figure 18 : Activity Diagram for a user in the system

3.8 Design

During the design phase, the relationships between entities was designed and analyzed using the entity relationship diagrams diagram. After that, the database schema was developed to illustrate the mapping of the data.

3.8.1 Entity relationship Diagram

The Entity-Relationship Diagram (ERD) model illustrates the relationships among the different elements within the system. Multiplicity notation is used to represent the various types of relationships, such as one-to-many, one-to-one, and many-to-one. The elements themselves are represented by rectangular boxes, and the relationships between them are depicted using diamond-shaped boxes. Additionally, links are established between the different elements to indicate their associations.

Within the system, there are several key elements represented in the ERD model:

1. **User:** This element stores the details and information pertaining to the users of the application.
2. **Diet:** The diet element contains diet-related information, such as meal plans or nutritional guidelines.
3. **Surf:** This element represents the request messages within the system.
4. **Fitness Mockups**: The fitness mockups element holds data related to the generated mockups for specific exercises or workout routines.
5. **Records:** This element encompasses information about the users or data associated with them.
6. **Reminders:** The reminders element includes details about reminder IDs, reminder times, and other relevant information.
7. **Fitness Programme:** This element captures information about the various exercise programs or fitness routines available within the system.
8. **Registration/Logout/Login:** These elements focus on user access data, including registration details, login credentials, and the process of logging in or out of the system.

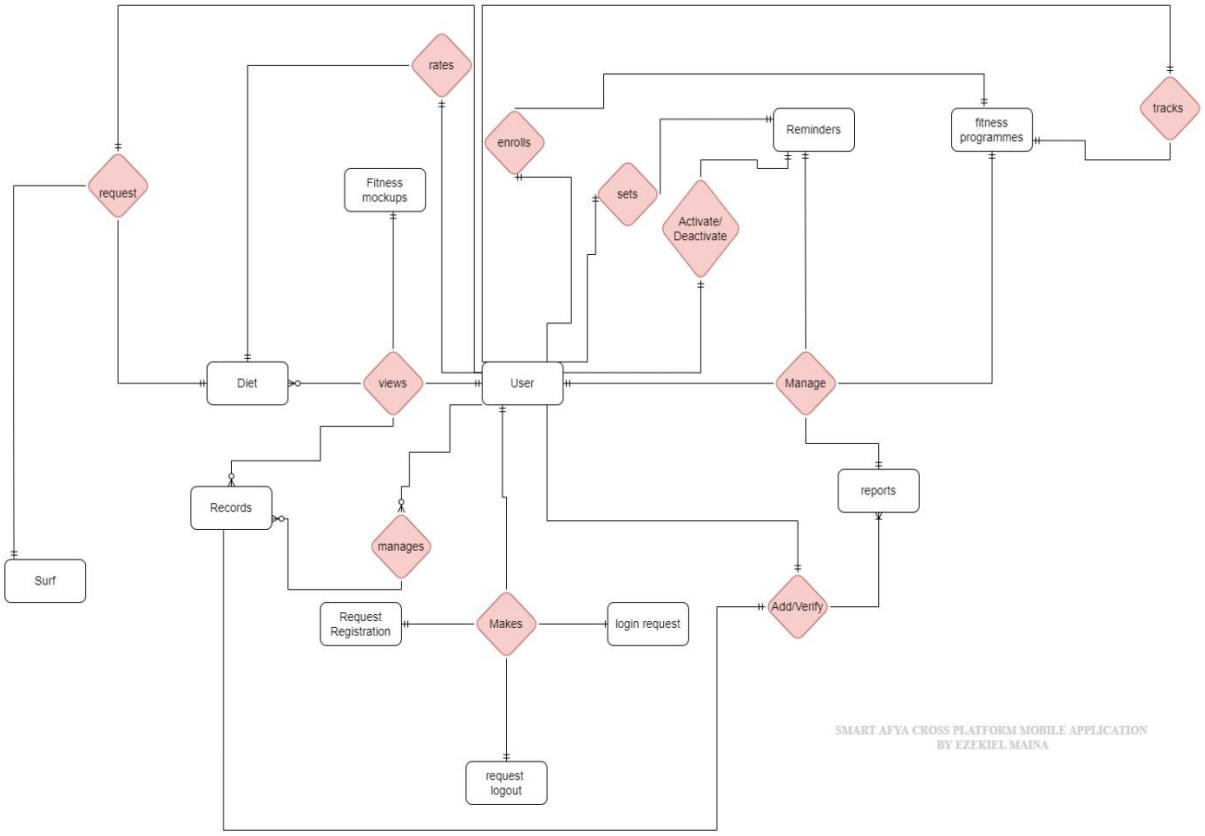


Figure 19: ERD Diagram

As depicted the relationship between users and diet element are views, rates and request. The relationship between the user element and the surf element is request. The user element has a makes relationship with registration element, logout element and log in element. All this data is useful in determining the access traffic of the system. An enroll relationship maps the user element and the programme element. This signs in records on the programmes that users are enrolled on. The reminder element is coupled with the user by activate/deactivate, sets and manage relationship. This generates data on active reminders which can be managed by the user element by the manage relationship. The user can also add/ verify records on them in the system.

3.8.2 Database Schema

A database schema is the skeleton structure that represents the logical view of the entire database. It defines how the data is organized and how the relations among them are associated. It formulates all the constraints that are to be applied on the data. The system has thirteen database tables as shown in Schema diagram. Each table contains an ID and several different attributes related to different functions in the System. The tables' relationships are derived from

the ERD model above and thus they also exhibit multiplicity. Where there is a relationship, the child table contains a foreign key (FK) from the parent table. The various attributes also have their data type description. VARCHAR represents that the datatype of the attribute is text while INT and BIGINT is for integer values and numbers, Other attributes like dates are represented using the DATE/TIME datatype, Attributes like emails and addresses are further defined to be unique for a given table and thus a UNIQUE constraint is added. Various attributes also cannot be null as they are considered essential in any data and thus the NOT NULL constraint is used. The values in brackets such as BIGINT(11) and VARCHAR(50) indicate the length of the characters to be store in the database about that particular attribute.

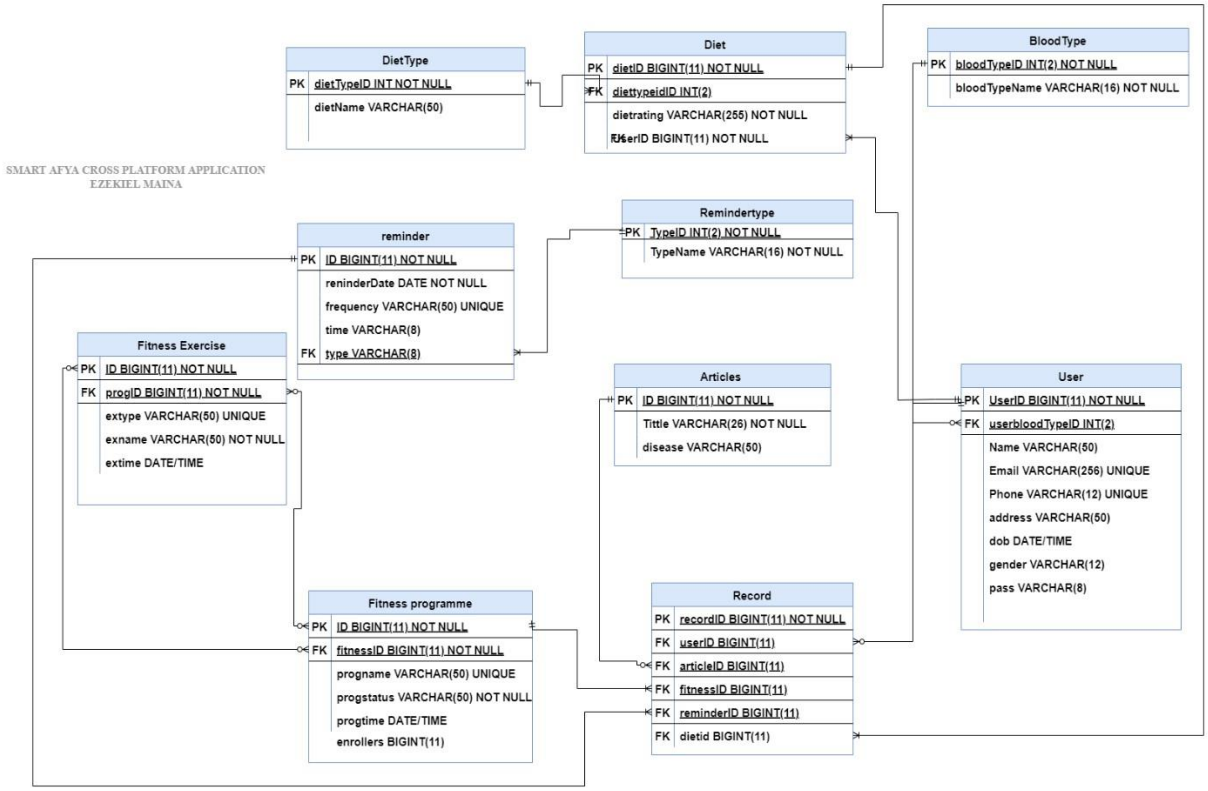


Figure 20: Database Schema

3.8.3 Input and Output Designs

Input design encompasses the creation of user-friendly and efficient forms or screens that facilitate the input of data into the system. The primary objective is to ensure that the forms are intuitive, easy to navigate, and encourage accurate and complete data entry. This involves carefully selecting appropriate input controls, such as text boxes, drop-down menus, and radio buttons, as well as designing the overall layout and flow of the form. The goal is to enhance the user experience by providing a seamless and effective data input process.

On the other hand, output design focuses on generating reports or other outputs based on the input data within the system. The aim is to create outputs that are informative, useful, and easily comprehensible for the intended audience. This entails selecting suitable visualizations, such as tables, charts, and graphs, to present the data effectively. Additionally, the layout and formatting of the output are carefully considered to ensure clarity and readability. The objective of output design is to deliver meaningful and actionable information to users, facilitating decision-making and providing valuable insights based on the system's processed data.

3.8.3.1 Input Design

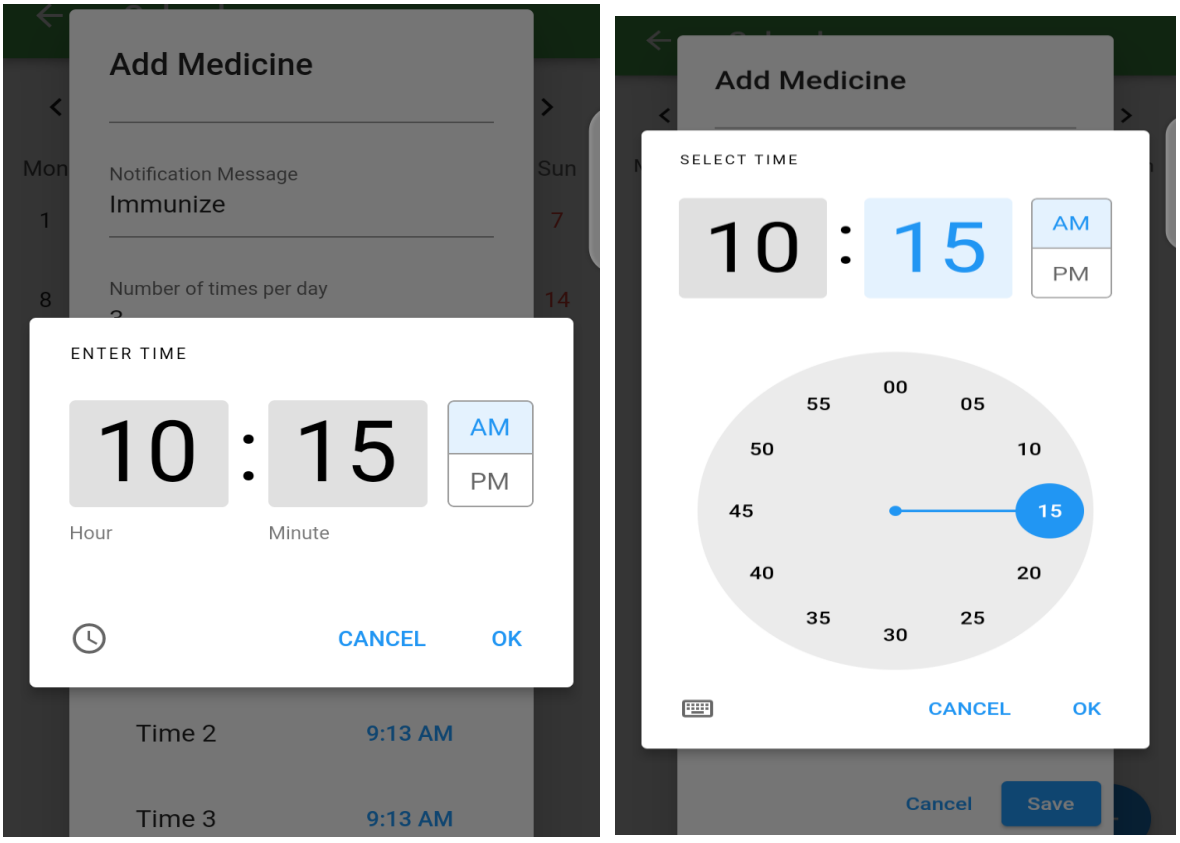


Figure 21: Reminder clock input

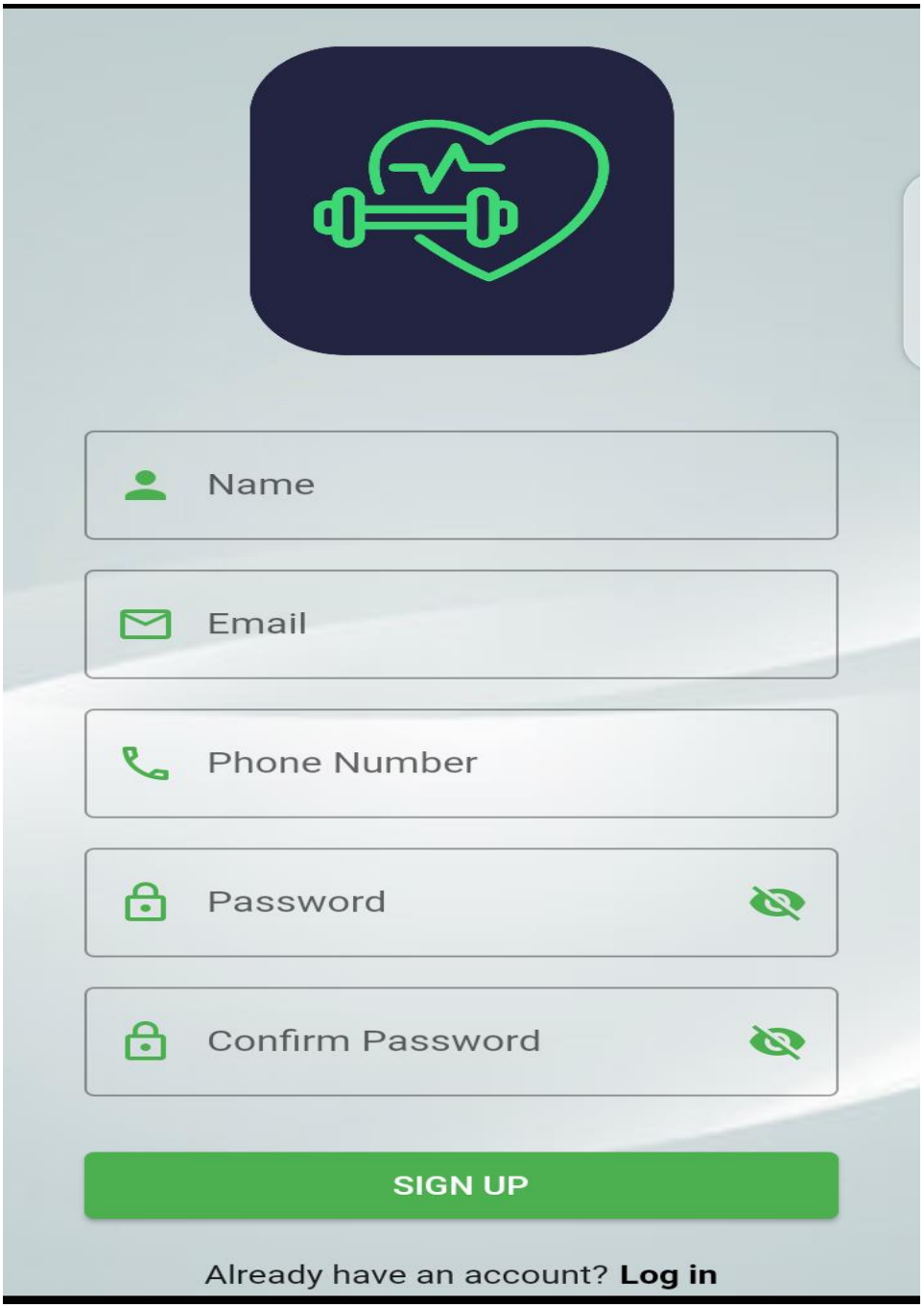


Figure 22: Sign Up form

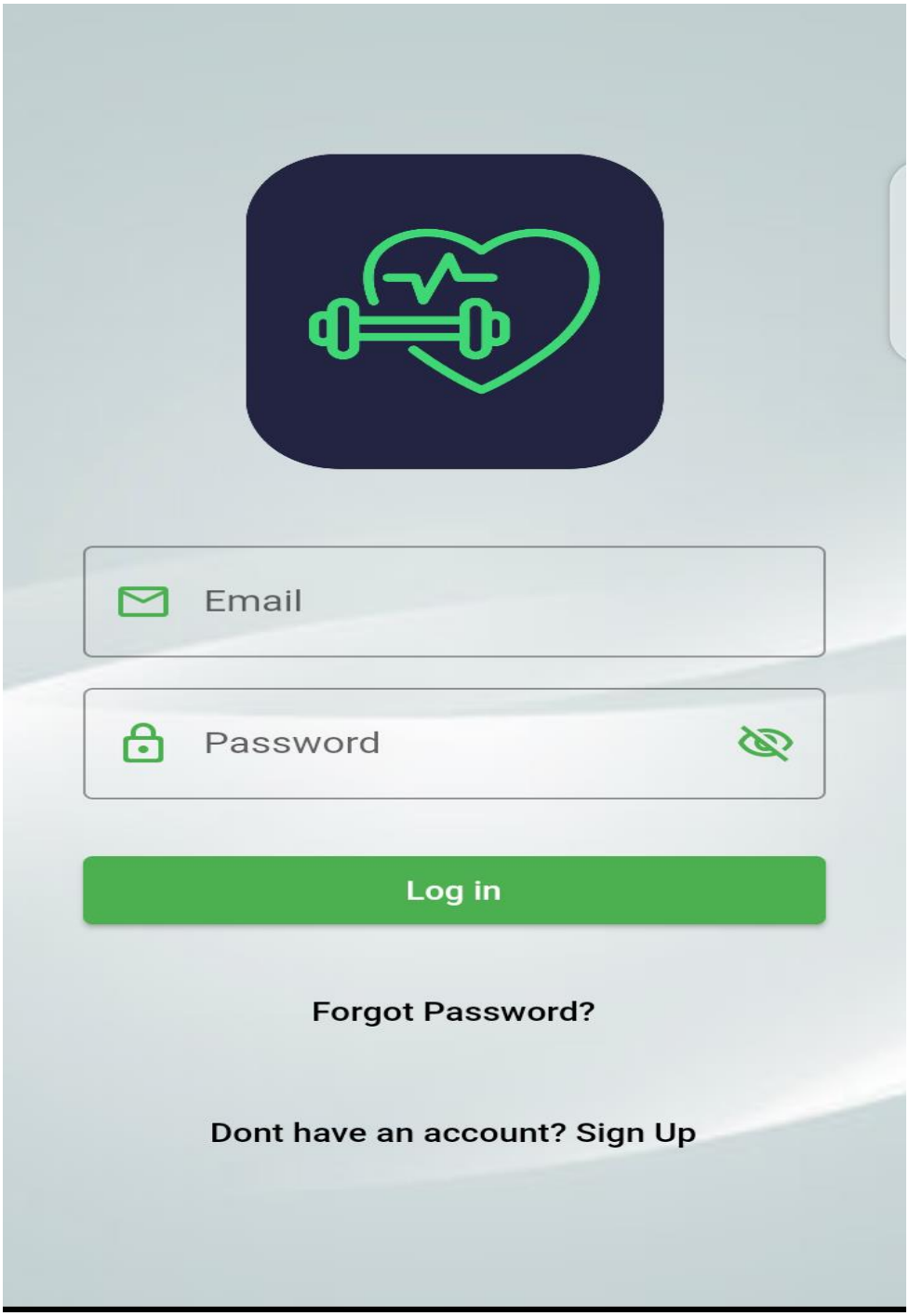


Figure 23: Log in Form

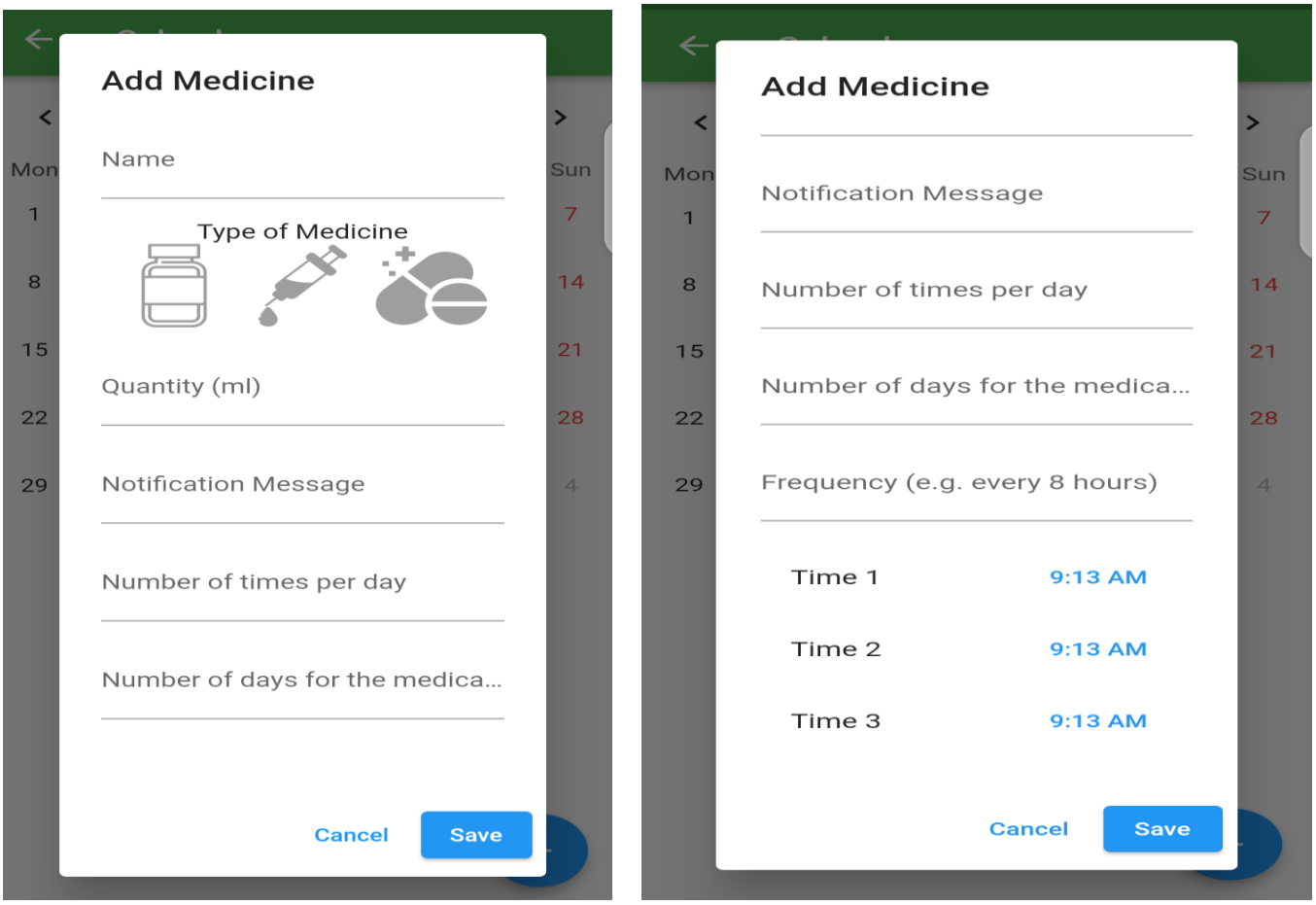


Figure 24: Add medicine Form

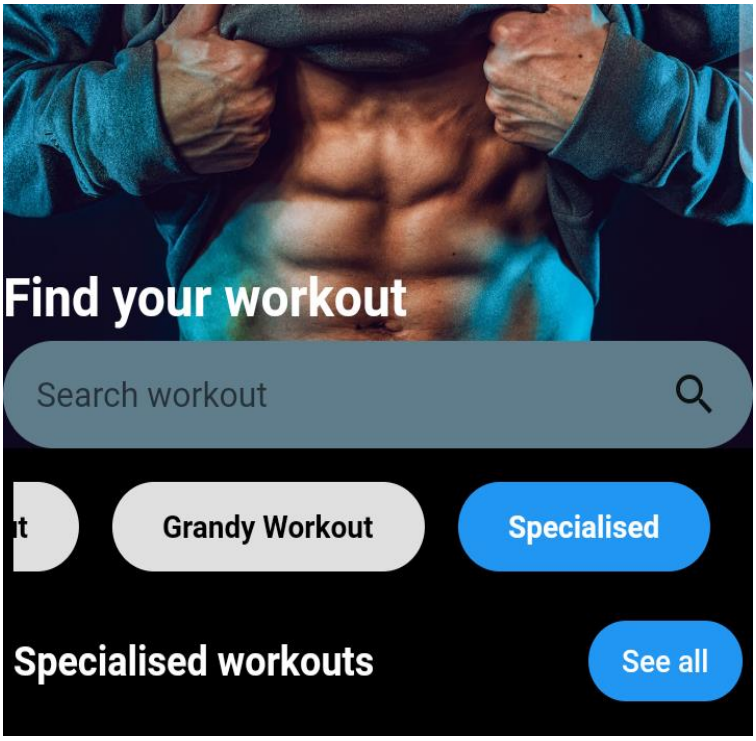


Figure 25: Workout search

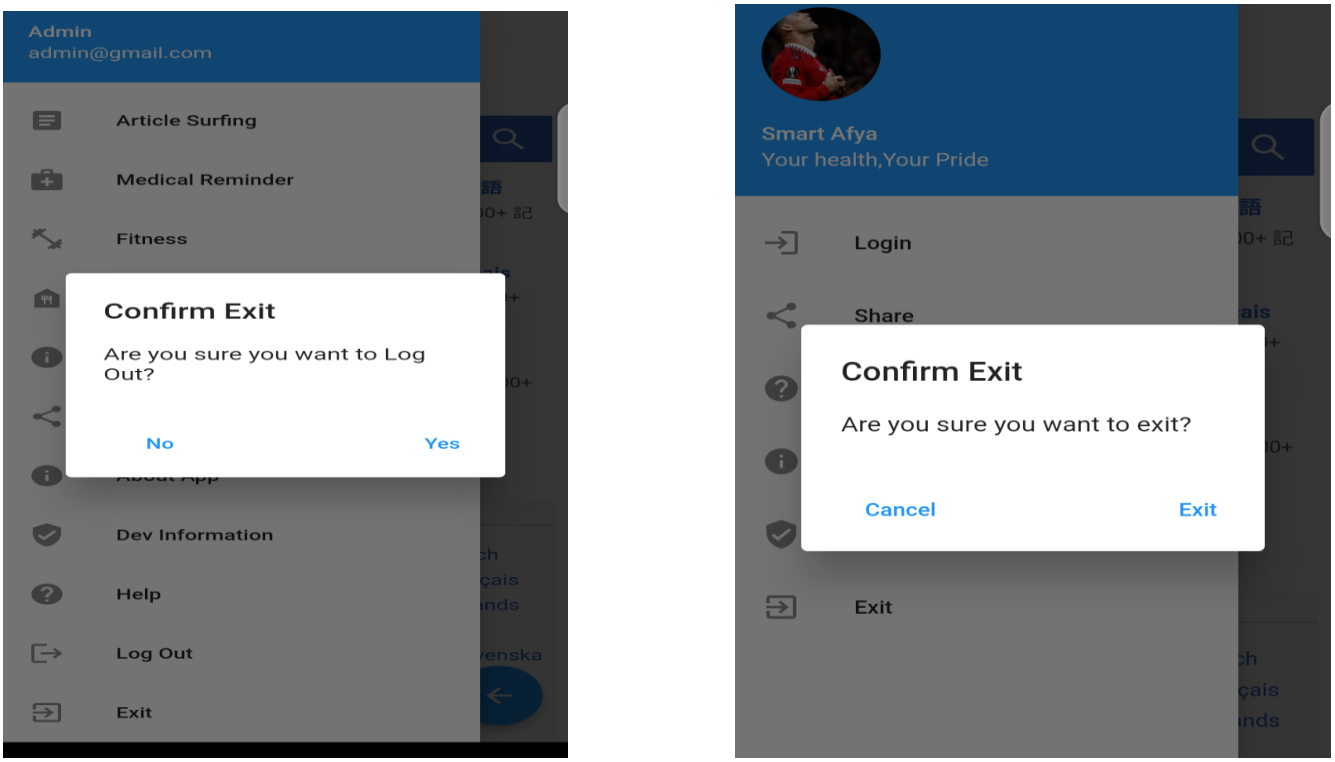


Figure 26: Button inputs

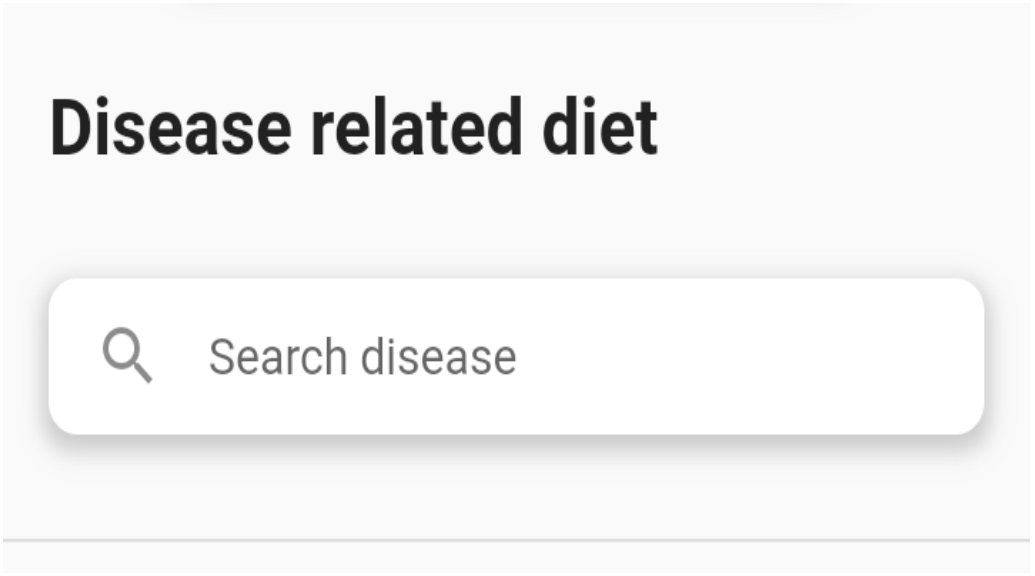


Figure 27: Disease related diet search input

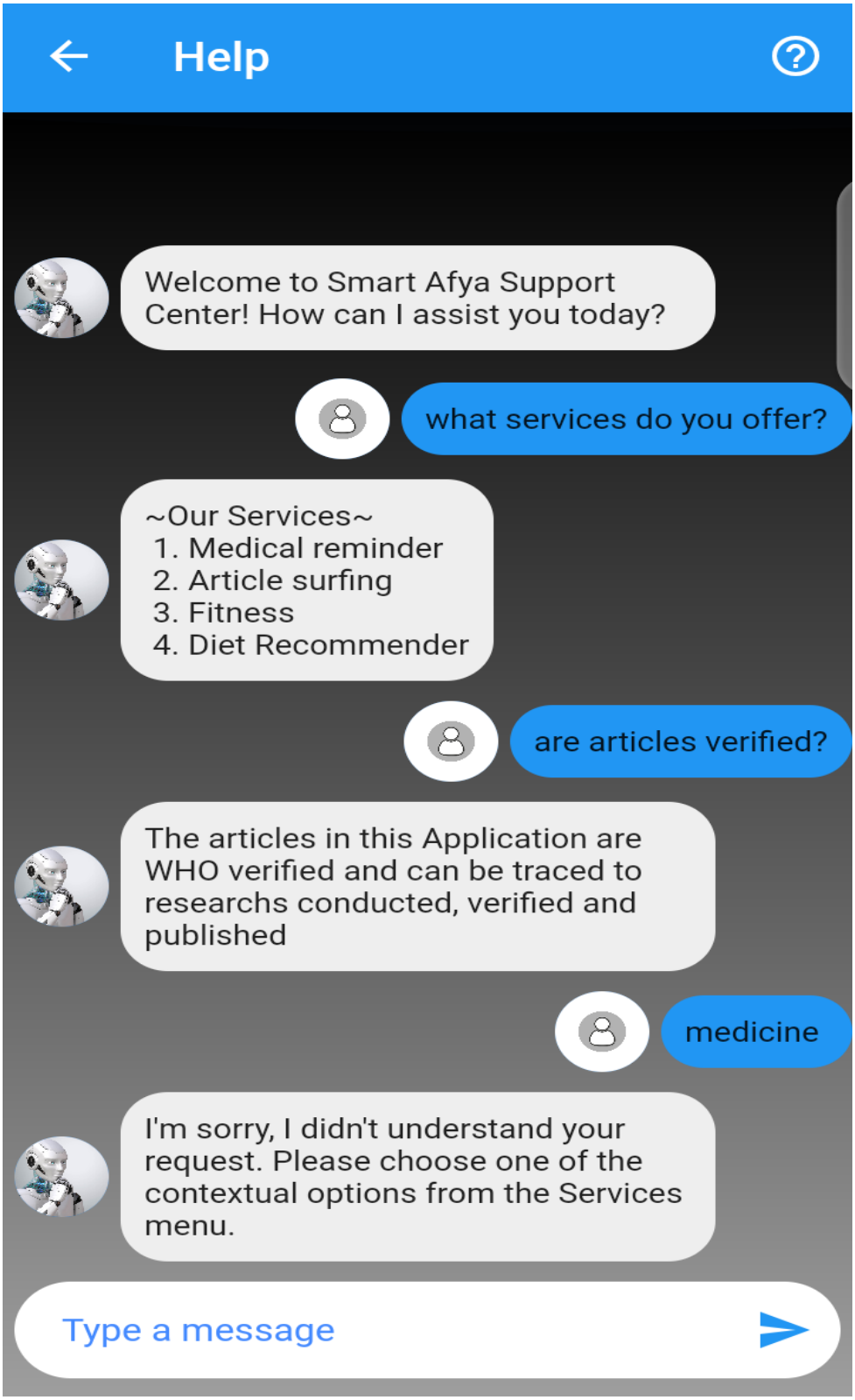


Figure 28:Help facility input

3.8.3.2 Output Design

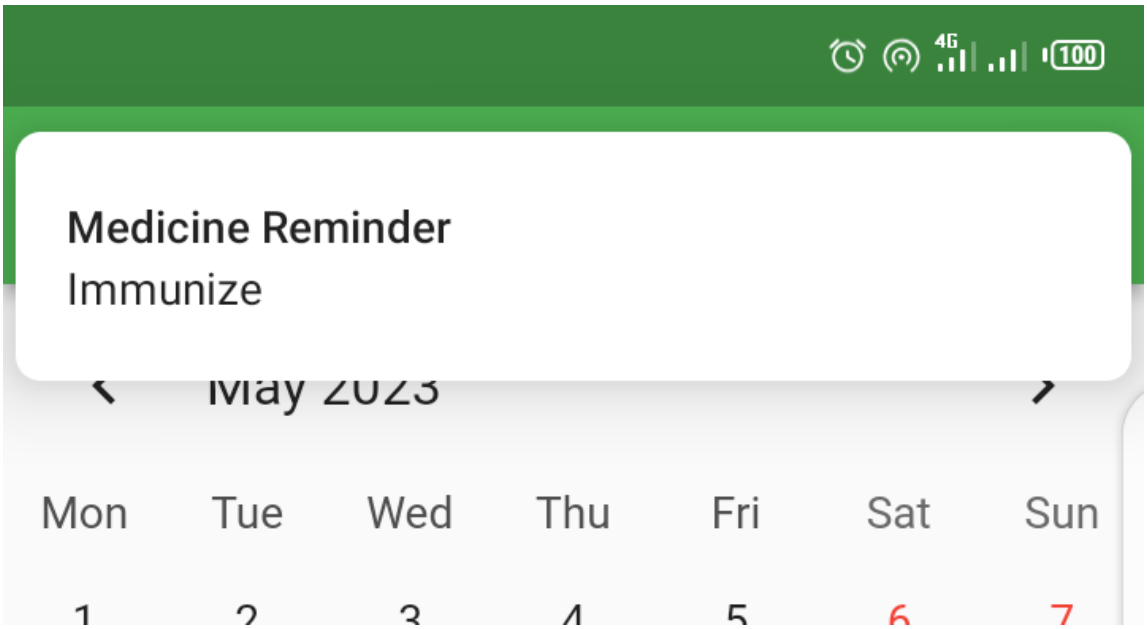


Figure 29: Reminder Notification

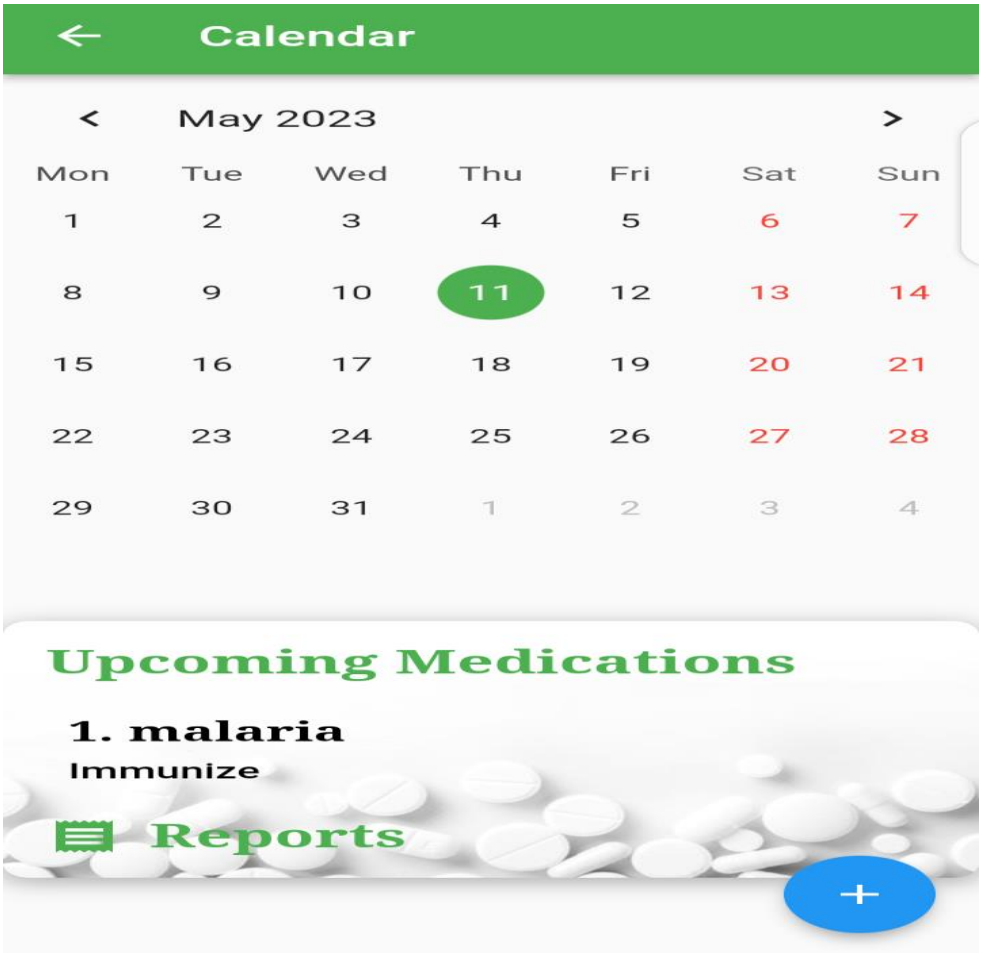


Figure 30: Notification Reports

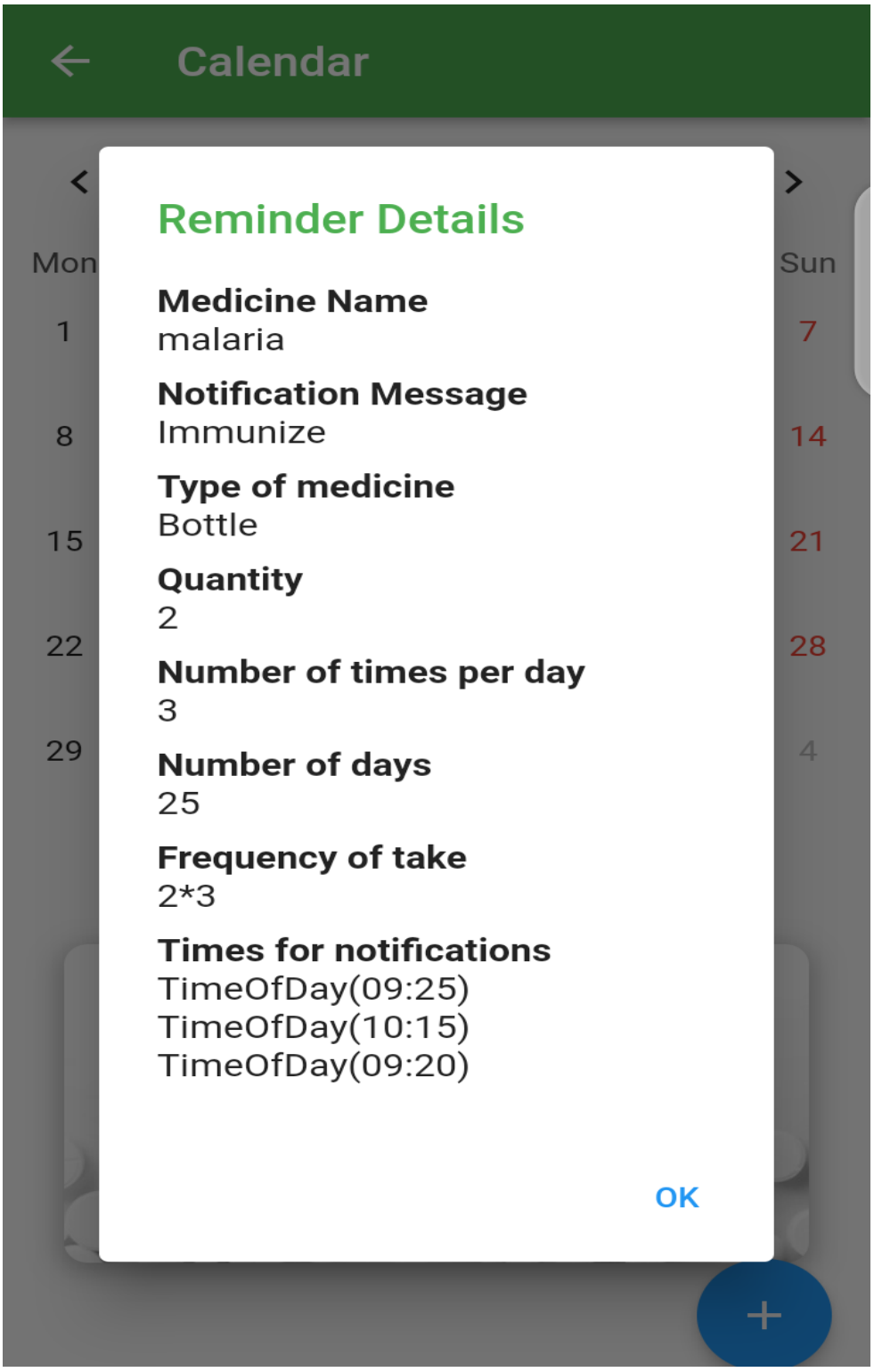


Figure 31: Reminder Output

3.9 Summary

This chapter has shown all the activities to be carried out in the analysis and design of the cross platform mobile application. The chapter begins with selecting a methodology and carefully analyzing system requirements by reviewing existing documents and providing questionnaires for evaluation and feedback on some proposed system features. The responses were mostly positive, with 69% of the users agreeing to the features, compared to 16% of those whose reviews were considered harmful. However, some 14 percent of the users provided responses that neither agreed nor disagreed with the feature. The chapter describes the system requirements and specifications, including a description of modules in the system. A feasibility study was conducted to determine the project's viability across the organizational, technical, financial, and operational dimensions. The project is easily feasible. The system's functional and non-functional requirements were then listed, upon which user use cases were also defined and diagrams of the same described using use case descriptions in table formats. The flow of events in significant use cases was listed to determine sequence diagrams, which aided in the design of data flow diagrams describing how information moves across the system.

Activity diagrams were created to depict the overall flow of activities for each user in the system. The entity relationships and database design were finalized using a schema created to complete the design.

Lastly, Input and output diagrams were used to show some of the various input forms that will be used and how they would look like. Outputs designs in form of tables were also used to depict how the data would be organized in the relevant tables. These designs would aid in the development of the system forma and reports to be generated.

In a nutshell, the chapter has successfully analyzed and brought out the design aspects of the system, making it easier to visualize and analyze the system later.

CHAPTER FOUR: SYSTEM CODE GENERATION AND TESTING

4.1 Introduction

System code generation and testing are integral phases in the development of a health-based system. During code generation, the actual code for the system is written using the proposed technologies, while adhering to the design approaches established in the earlier design phase. This process ensures that the system is implemented accurately, translating the design specifications into functional code. The interfaces are specifically designed to cater to the unique needs of the users, as described in the functional analysis report. By considering user requirements, the interfaces are tailored to provide a seamless and intuitive user experience.

System testing plays a vital role in verifying the correctness and reliability of the health-based system. It involves executing the program and meticulously examining its behavior to identify any errors, discrepancies, or malfunctions. In the case of the health-based system, testing focuses on evaluating the effectiveness of the reminder functions and report generation. Test cases are developed based on the system specifications, particularly the functional requirements. Through rigorous testing, faults and defects are identified and rectified, ensuring that the system meets the desired quality standards.

The testing phase is crucial as it helps eliminate potential flaws and enhances the overall quality of the system. By identifying and resolving errors early on, the health-based system can provide accurate and reliable functionalities, instilling confidence in its users. The combination of thorough code generation and meticulous testing contributes to the development of a high-quality, dependable health-based system.

4.2 System Code Generation

During the initial stages of system development, careful observation and identification of the various actors who will directly engage with the system were conducted. The primary actor identified at the outset of the project is the user, as the system is designed to be a single-user system.

To ensure a smooth and enjoyable user experience, the development of the mobile application followed a systematic approach. The first step involved creating a visually appealing splash screen page, which serves as an introductory screen that captures the user's attention and sets the tone for their interaction with the application. This splash screen page is designed to provide a seamless transition into the main Wikipedia landing page, which serves as the central hub for user interactions within the system.

By implementing this systematic approach, the development team aims to create a mobile application that not only meets the functional requirements but also engages and captivates the user from the moment they launch the application. The splash screen and Wikipedia landing page play vital roles in establishing a positive initial impression and setting the stage for a seamless and immersive user experience throughout the application.

After establishing the splash screen and Wikipedia landing page, the development team focused on ensuring a user-friendly and visually appealing user interface for the system. This involved creating screen pages for each of the modules, considering the design considerations in detail. To streamline the development process and save time on coding, Flutter, a cross-platform framework, and APIs were employed. This combination facilitated the integration of front-end and back-end functionality, enhancing the efficiency of the programming process.

Moving forward, the subsequent phase of system development entailed identifying the key modules that would be vital for the single user of the system. Special attention was given to understanding the specific requirements and preferences of each user. With the database schema carefully designed to align with these specifications, the next step involved the actual creation of the database. This process ensured that the database structure corresponded accurately to the needs of the system and its users.

By systematically addressing the design considerations and employing efficient technologies, the development team aimed to deliver a well-designed user interface that met the functional requirements of the system. The integration of Flutter and APIs allowed for a streamlined and effective development process, while the thoughtful identification of key modules and database creation ensured a robust foundation for the system.

After implementing the necessary modules, users were able to utilize various functionalities within the system. These included surfing, setting medical reminders, engaging in exercises, receiving diet recommendations, playing therapeutic games, and accessing reports. By following a systematic approach throughout the development process, the mobile application was designed to cater to the diverse needs of its users, ensuring a user-friendly and efficient experience.

In terms of the backend development, Node.js was utilized as the programming language. Node.js is known for its scalability and event-driven architecture, making it suitable for handling concurrent connections and real-time applications. REST APIs (Representational

State Transfer) were employed to facilitate communication between the front-end and back-end components, enabling data retrieval, manipulation, and interaction with the system.

As for the database, a NoSQL (non-relational) approach was adopted. NoSQL databases, such as MongoDB or Firebase Firestore, provide flexibility and scalability, allowing for the storage and retrieval of unstructured or semi-structured data. This choice was made to accommodate the dynamic and evolving nature of mobile applications, ensuring efficient data management and system performance.

By employing Node.js, REST APIs, and a NoSQL database, the development team aimed to create a robust and responsive backend infrastructure that supported the system's functionalities and user interactions. This technology stack offered flexibility, scalability, and efficient data handling capabilities, contributing to the overall success and performance of the application.

To retrieve or manipulate data in Firebase, considerations were made on query operations such as filtering, sorting, and limiting the results. These queries are typically performed using Firebase's Realtime Database or Cloud Firestore. You can also use Firebase Authentication for user authentication and Firebase Cloud Functions for serverless computing.

After completion of individual module development, they were integrated to form the system as a whole. It was important to maintain the logic of the system during the process of linking different modules to ensure that they were understandable and consistent. Therefore, a click of a button should redirect to the next screen, and an action that should lead to an equal change in another module should do so.

4.2.1 User Interfaces

4.2.1.1 Splash Screen

A splash screen is an introductory screen that appears briefly when an application or system is launched. It serves as a visual representation of the application's branding, providing an initial impression to the user. Typically, a splash screen includes the logo, name, or any other relevant visual elements associated with the application.

The purpose of a splash screen is twofold. Firstly, it provides feedback to the user, indicating that the application is loading and preparing for use. This helps manage user expectations and prevents them from perceiving a delay as an application malfunction. Secondly, a splash screen creates a sense of continuity and professionalism by establishing a consistent visual identity for the application.

A well-designed splash screen contributes to the overall user experience by capturing the user's attention and setting the tone for their interaction with the application. It can create anticipation, generate brand recognition, and provide a cohesive visual transition into the main interface. However, it is essential to keep the duration of the splash screen brief to avoid frustrating the user with unnecessary delays. For the app it is set on display for 5 seconds.

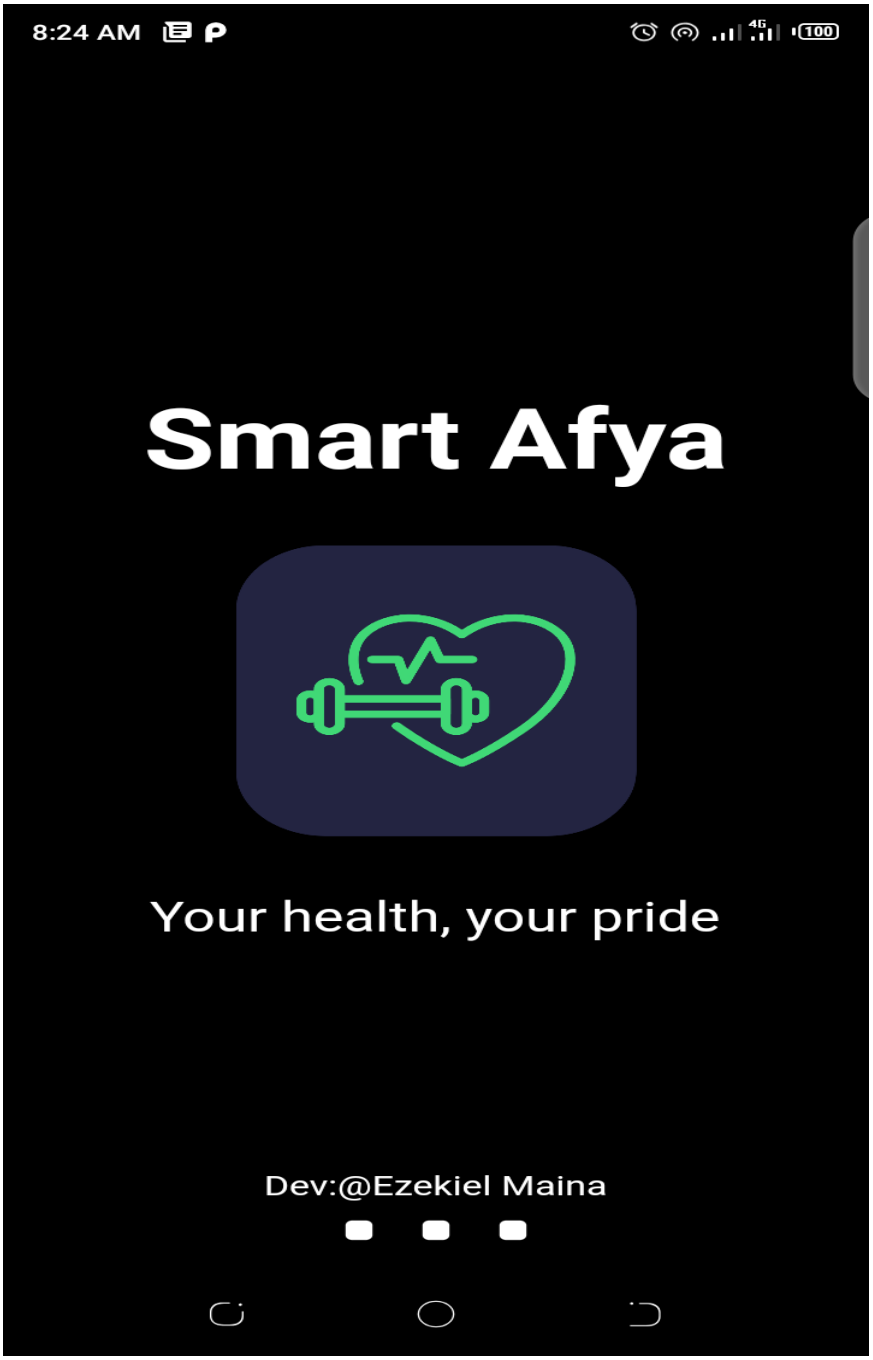


Figure 32: Splash Screen

4.2.1.2 Home Page

This is also the landing page, it has an interactive and responsive layout to enable users to interact with information on health. The Home page is the Wikipedia as a result of Wiki API. It relays vast information on health. It is the main connector as it has all the links to other modules in the application thus it is the starting point to access all the other modules.

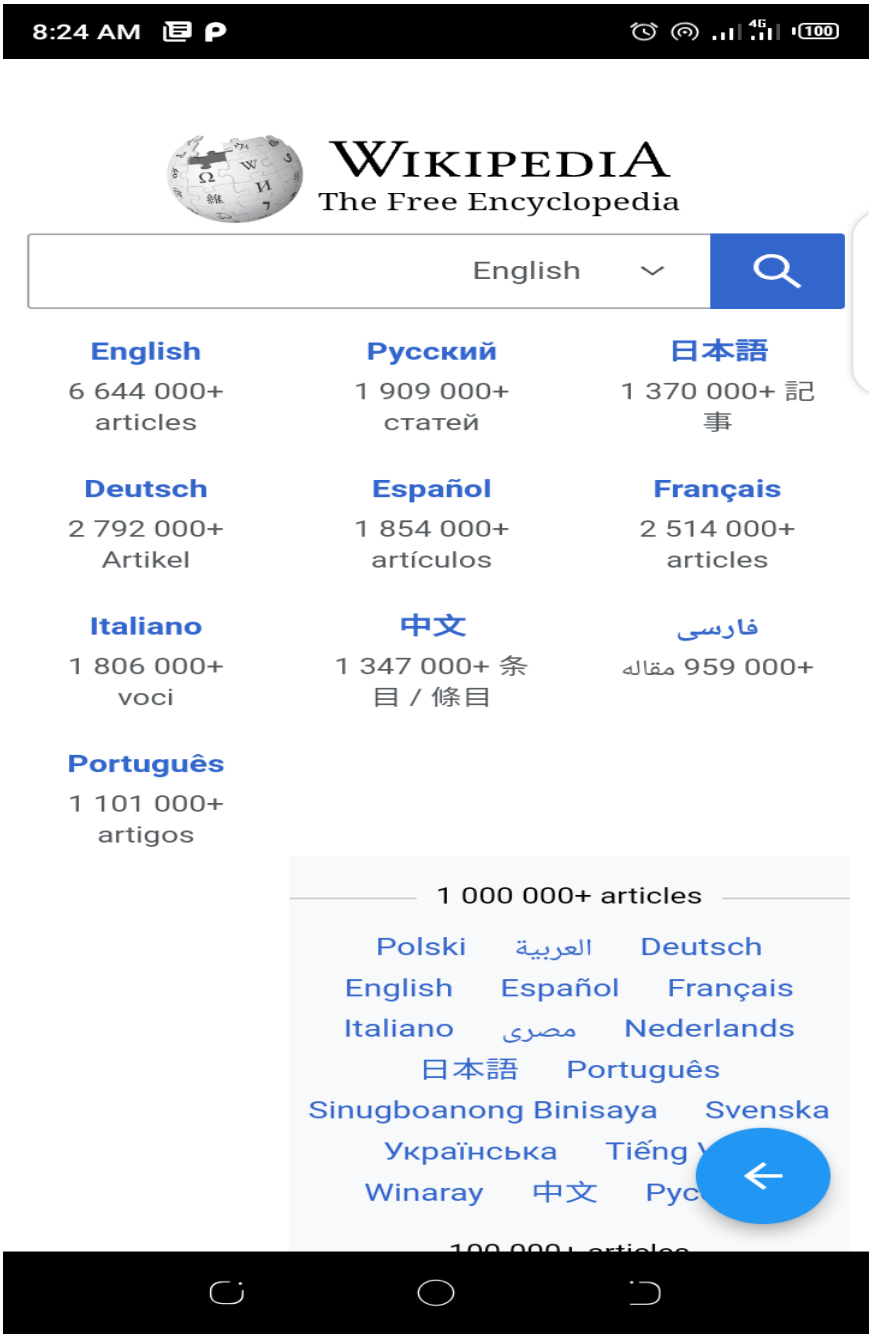


Figure 33: Home Page

4.2.1.3 Drawer

The drawer serves as a centralized hub for navigation, organizing app sections, settings, and other relevant content in a hierarchical or categorized manner. It allows users to easily switch between different sections of the app or access additional features, providing a convenient and efficient way to explore and interact with the app's functionalities.

The drawer is invoked by an arrow pointer in the Home page or by sliding left to right from the left hand side of the screen

The app has 2 drawers:

1. **Drawer for in-active session**

   This is the drawer presented when no one has logged into the system

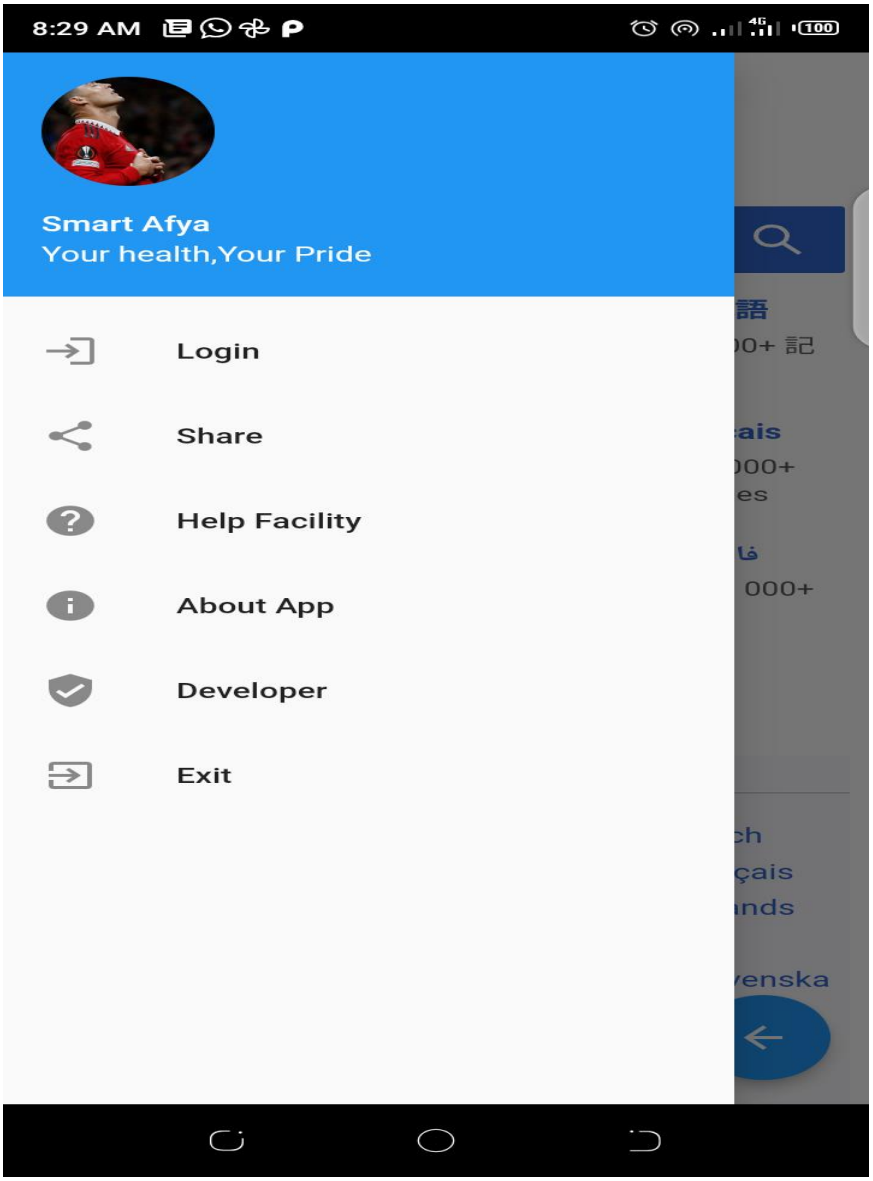


Figure 34: Drawer for inactive user

2. **Drawer for active session**

This is the drawer presented when one has logged into the system

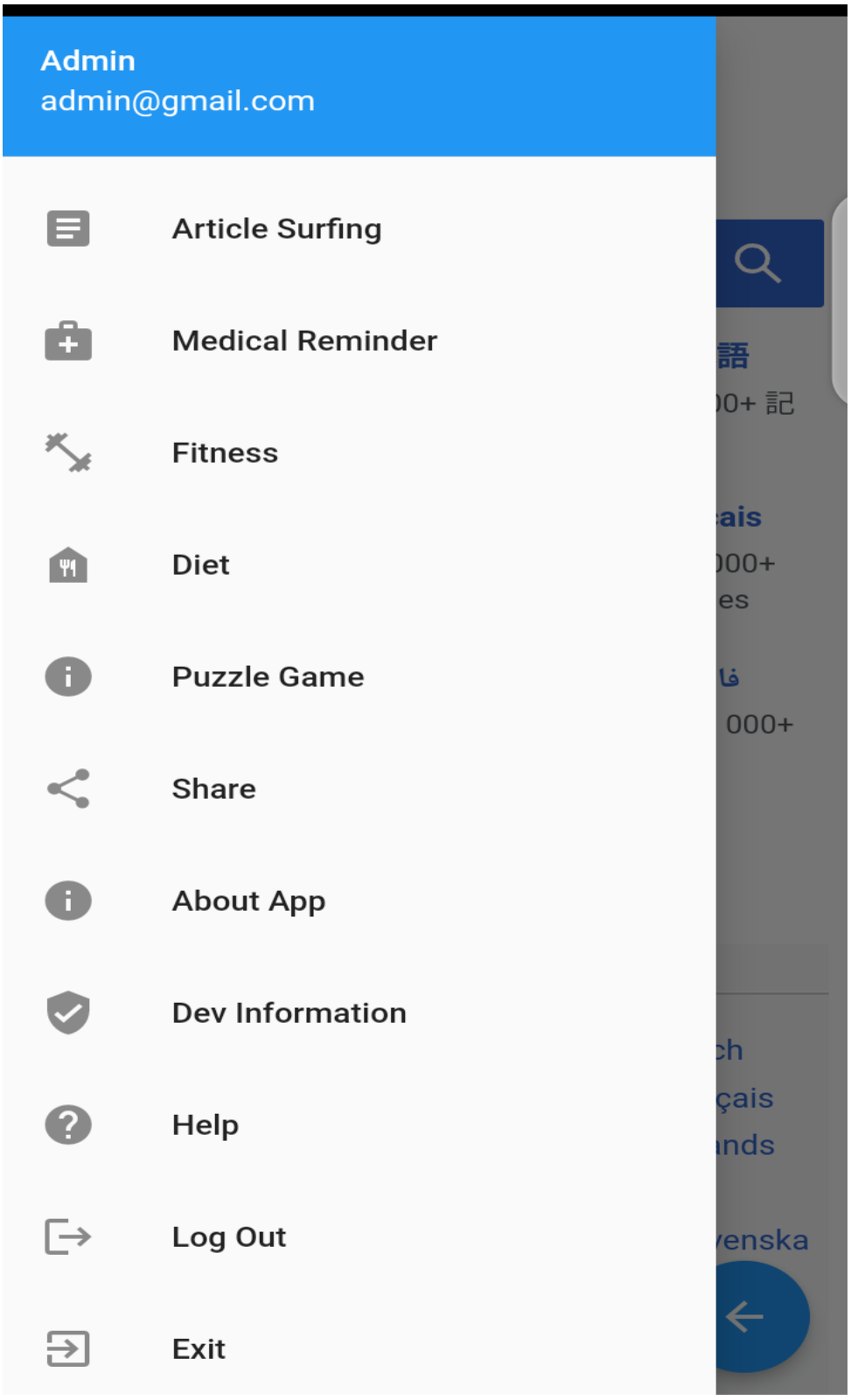


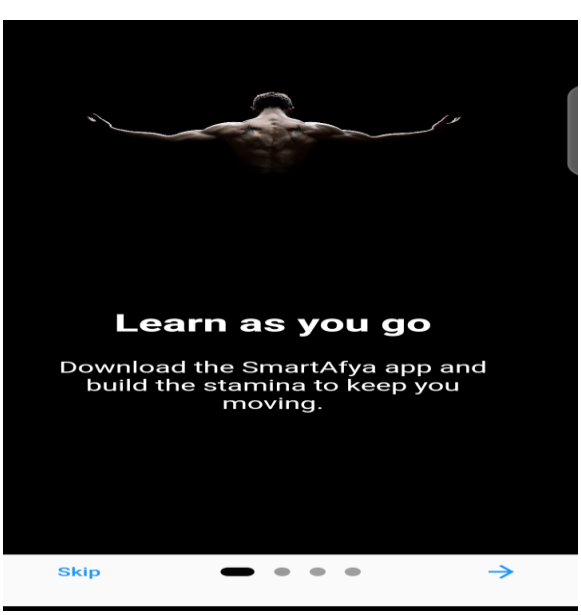
Figure 35: Drawer for active user

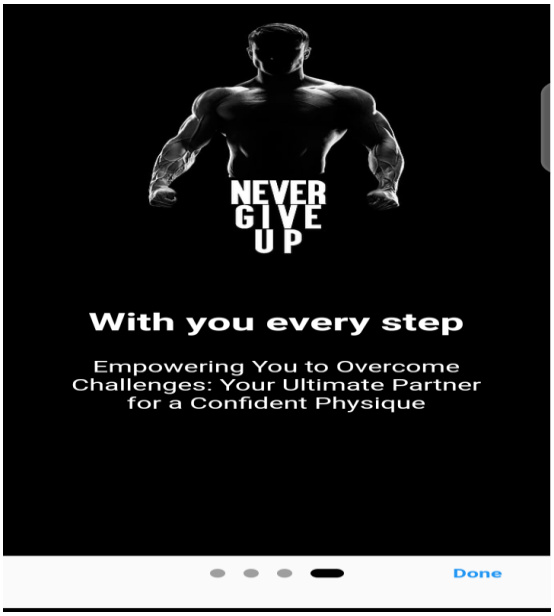
4.2.1.4 Screens

A visual interface or layout that is displayed on a mobile device's screen. It represents a specific view or state of the application, presenting information, interactive elements, and user interface components for users to interact with. Screens in the application include the following:

Figure 36:Articles screen

Figure 37: Article View





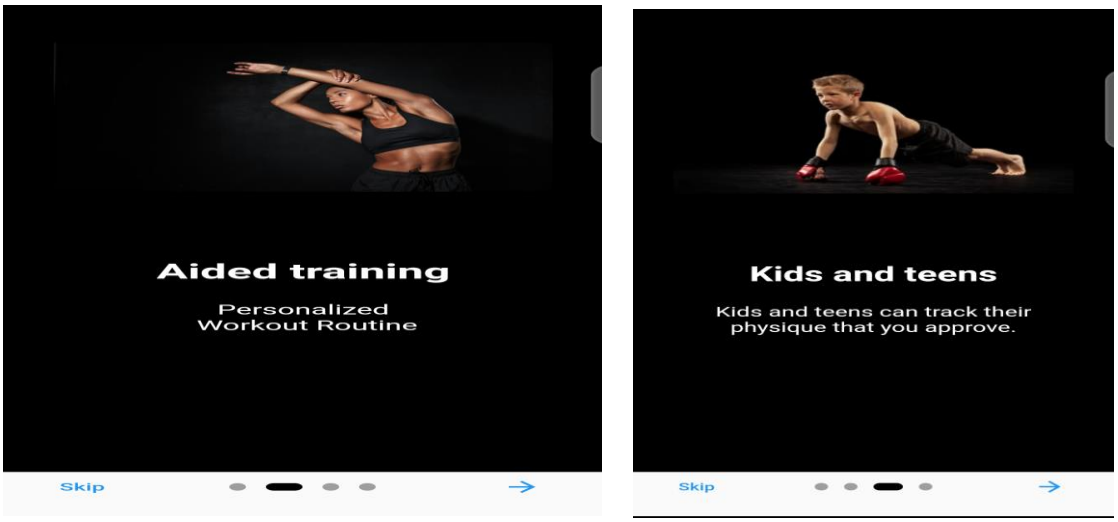


Figure 38: exercise boot screens

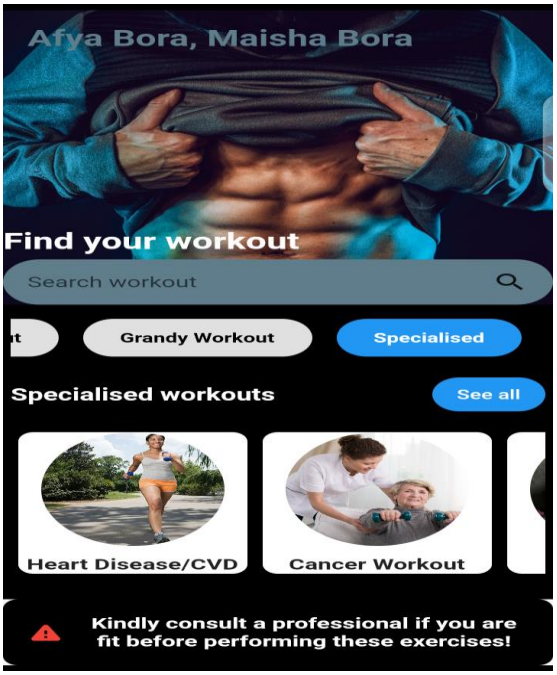


Figure 39: Specialized workout

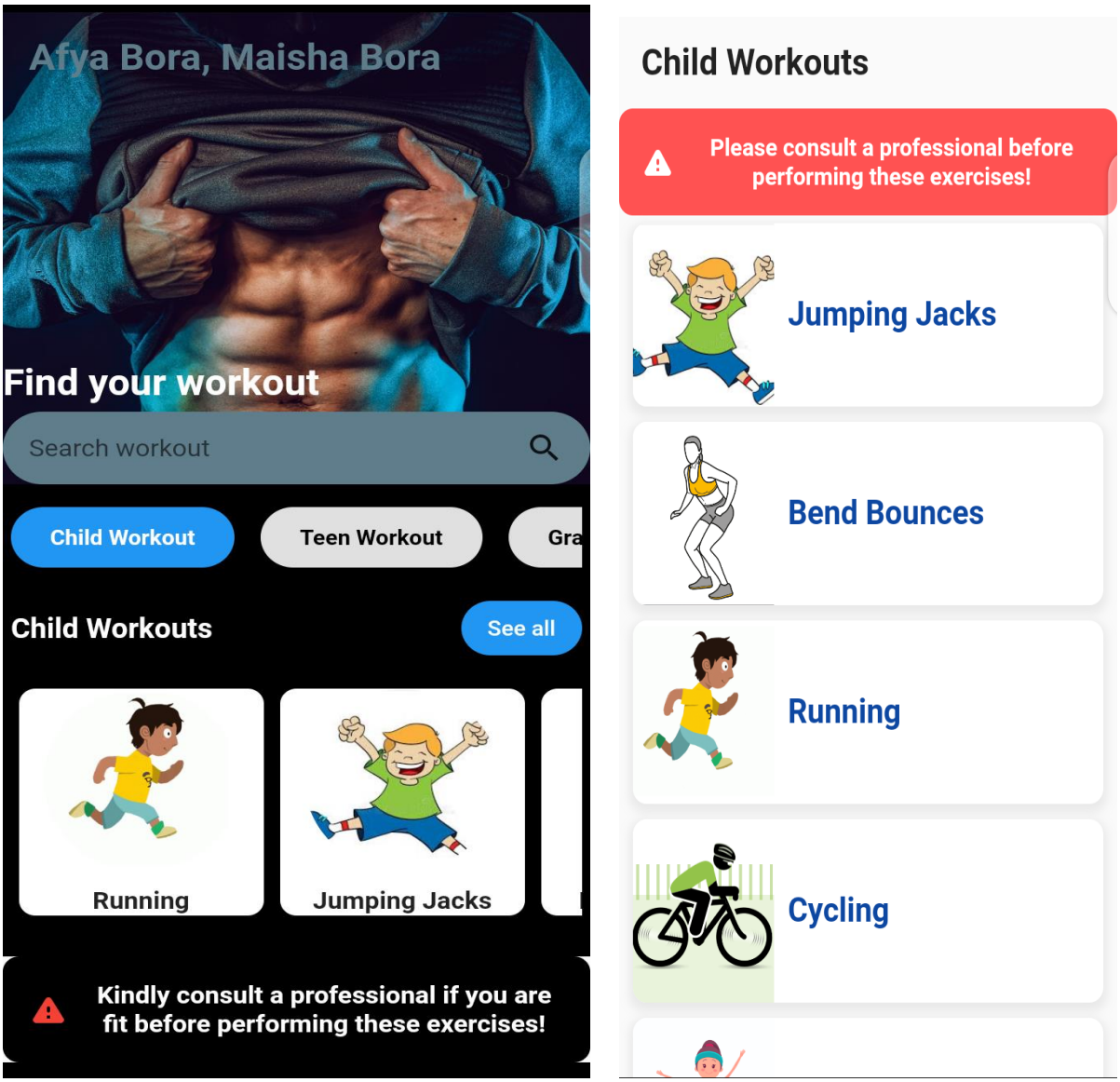


Figure 40: child Workout

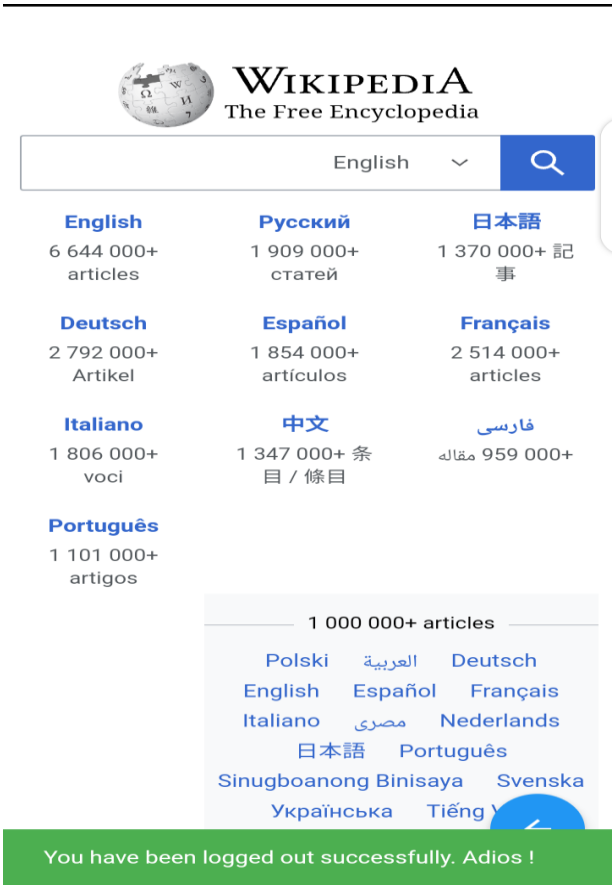
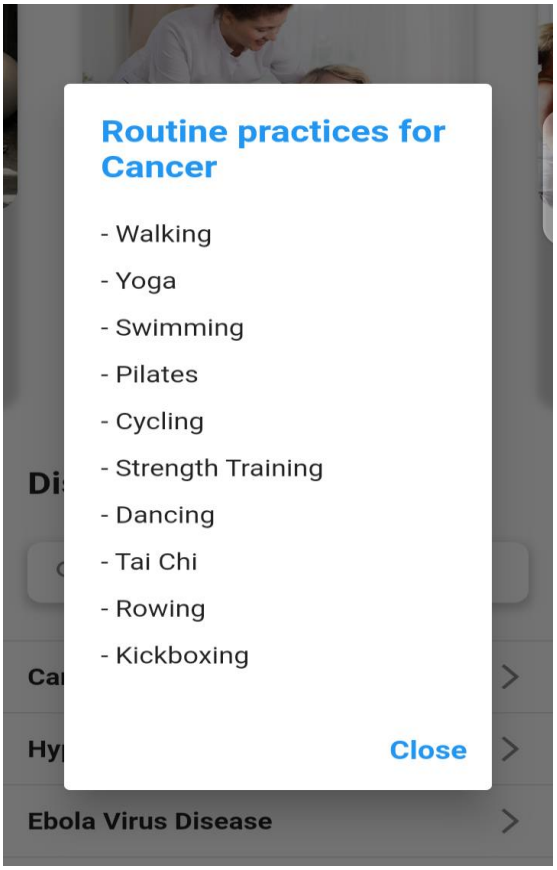


Figure 41:logout screen

Figure 42:about app



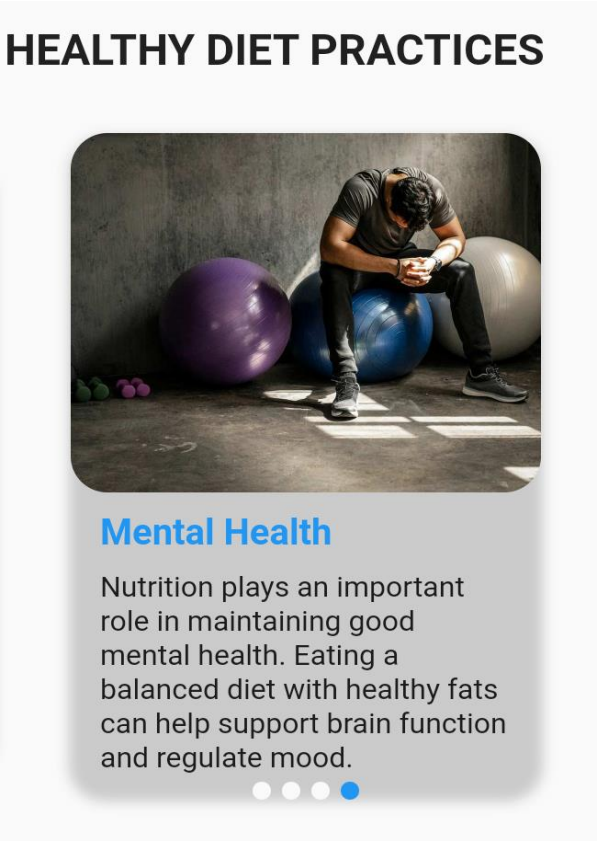


Figure 43:Routine practise

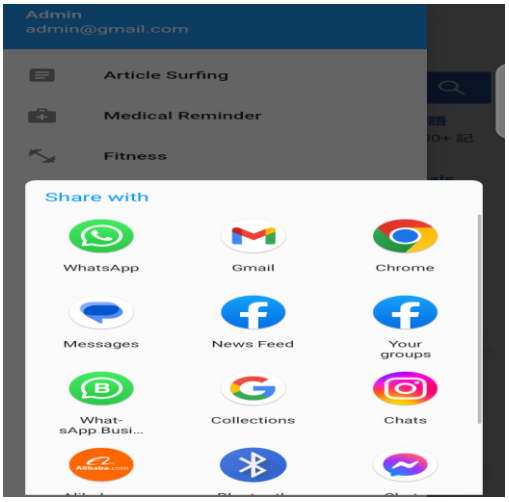


Figure 44: share app

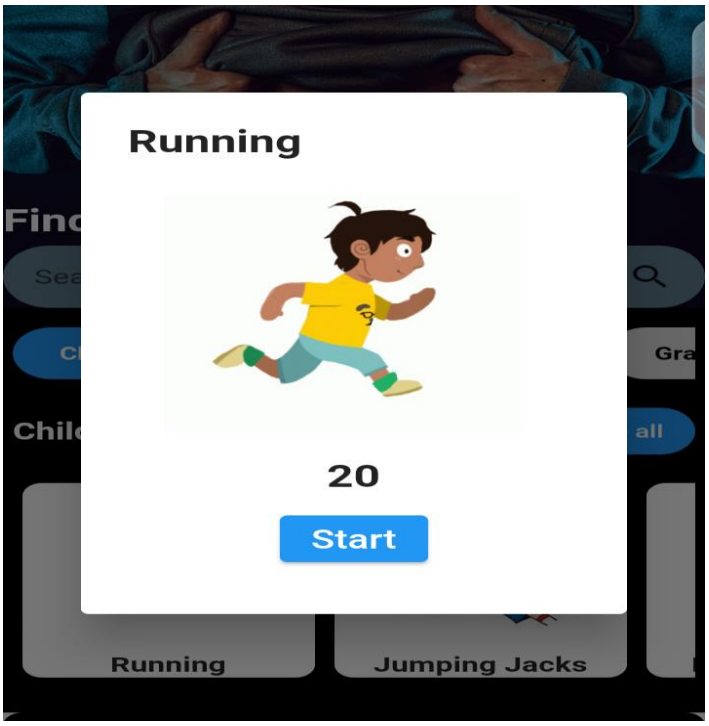


Figure 45: Running

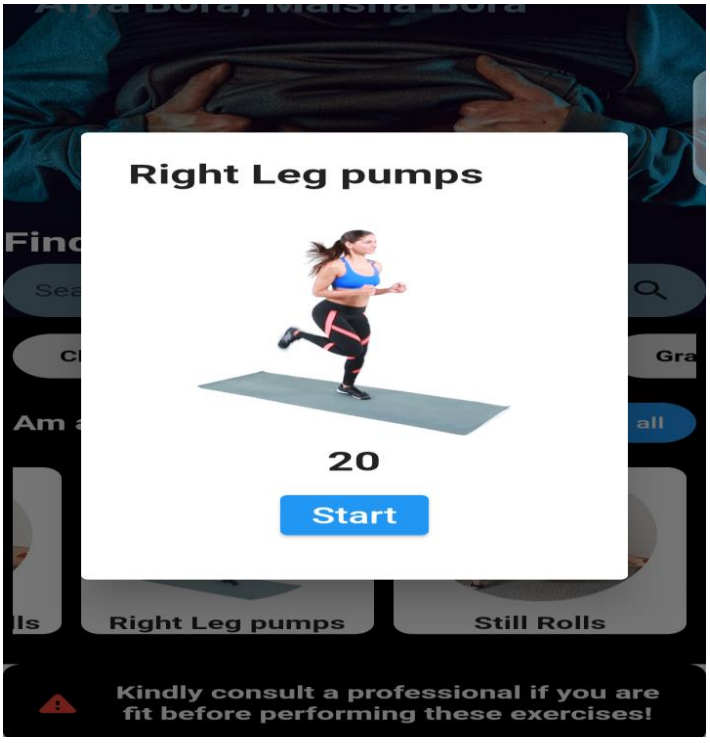


Figure 46: Leg pumps

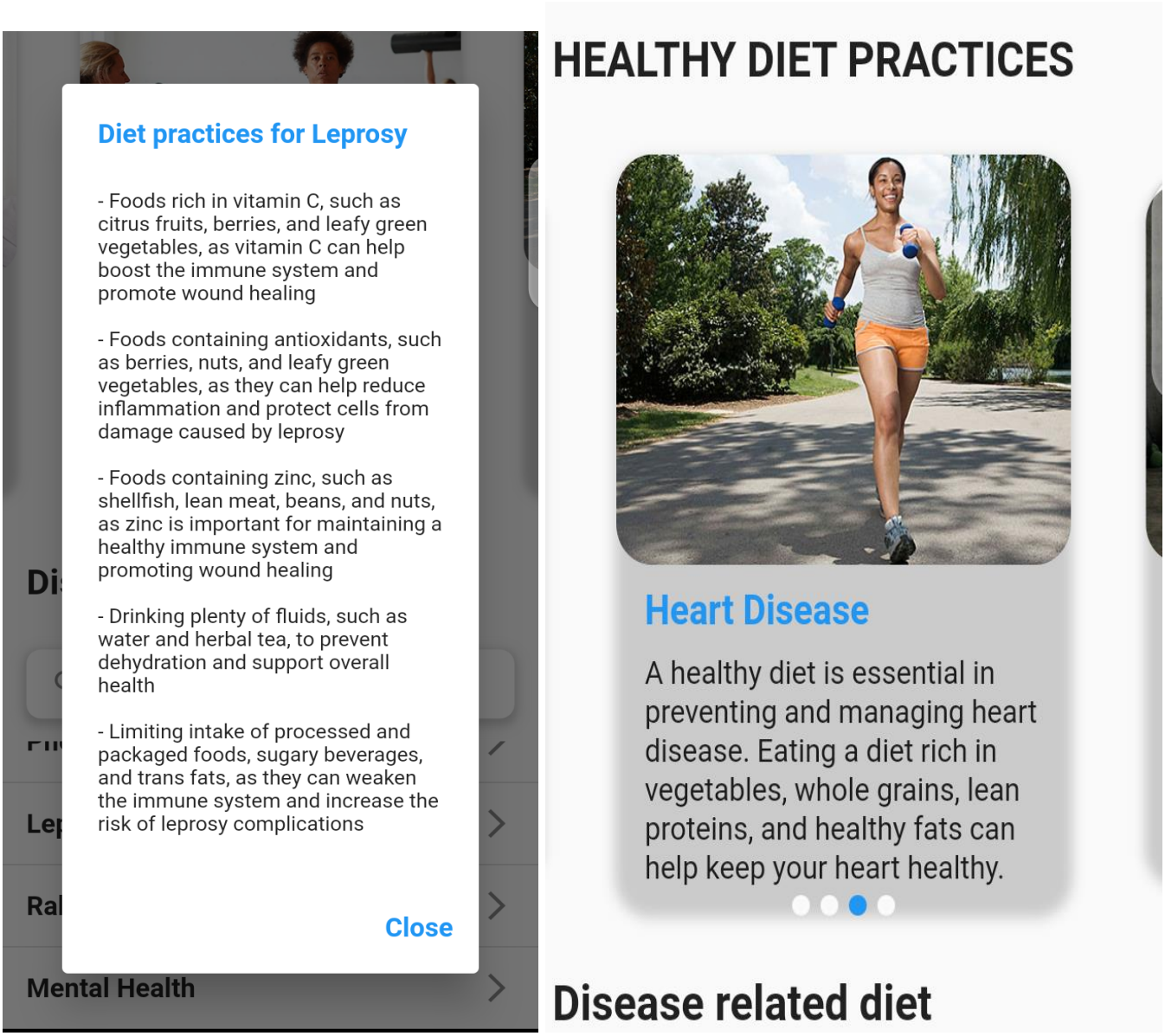


Figure 47: Leprosy diet recommendation

4.3 Testing

4.3.1 Testing Approaches

The system testing approaches are used to check the functionality of the system based on the objectives declared. System testing was carried out on the whole system either to check on the system requirement specifications or functionality requirement specification. These tests were carried out to detect and solve the errors experienced during the system development. These tests ensure the system is free from errors and has high reliability. The following are the tests carried out on the system:

i. Unit testing

ii. Modular testing

iii. System Testing

iv. Acceptance testing

4.3.1.1 Unit Testing

Unit testing involves testing the smallest components of the system individually to assess if they are operating as required. The unit test involves testing the input fields to ensure the correct data format is entered into the system. The hyperlinks and buttons are fully functional to the respective pages. Validation for all the input fields can also be done at this stage. All functions that have been used in the system are also tested to ensure a valid output is received by the users.

4.3.1.2 Modular Testing

Modular testing takes into consideration specific modules of the system. The mobile application comprises of modules such as article surfing, medical reminder, diet recommendation and fitness. A single module was taken at a time and tests are carried out to ensure the module was fully functional. Once all the modules were tested and approved then the next testing phase was carried out.

4.3.1.3 System Testing

In system testing, all the modules will be integrated and the system tested as a whole. In Smart Afya Cross platform Mobile Application there are some specific modules that are dependent

on each other. Tests were carried out to see if those modules are synchronous with each other. The medical reminder module and reports module work hand in hand. The system testing approach used a specific method to ensure the operation of the system is optimal. The following steps were used:

Creation of test cases - This outlines the specific types of tests that are to be carried out.

Create test data - The data to be used for testing is then selected. Extreme data are chosen to see how the system will respond in extreme cases.

Execute test case - After the creation of the test data the test cases are then executed.

Report generation - The results of the test case execution are noted down. All the success and failed test cases are listed.

Regression Testing - This is carried out to test the side effects of the testing process.

Log defects - The failures incurred during testing are then addressed and corrected at this stage.

Retest - After the log defects is done, a retest is done to check if the errors encountered were fixed.

4.3.1.4 Acceptance Testing

Acceptance testing deals with presenting the system to the client once all the other tests have been successfully carried out. The client will then check if the system developed check out with the system and functional requirements specified. If the system fulfils all the required specifications then it is ready to be deployed and be used.

4.3.2 Test Environment

The web application system was developed and designed using specific software and hardware conditions that led to its successful implementation.

4.3.2.1 Software Requirements

Software	Description
Operating System	Microsoft Windows 11 Pro 10.0.22000 N/A Build 22000
Programming Languages	Flutter , dart
Internet Browser	Google Chrome, Microsoft Edge, Mozilla
Integrated Development Environment	Visual Studio Code , Android Studio
Mobile phone API	• Google APIs intel x86_64Atom system image • Intel x86 Emulator Accelerator • Patch Applier v4
Cloud Server	Firebase
DBMS	NoSQL

Table 11: Software Requirements

4.3.2.2 Hardware Requirements

Hardware	Description
Processor	Intel(R)Core(TM) i5-3427U
Memory	4.00 GB RAM
CPU clock speed	1.80GHz 2.30 GHz
Resolution	1366 × 768 (Recommended)

Table 12: Hardware Requirements

4.3.3 Test Cases Used

The following table shows the test cases that were carried out and the results gotten.

Test case Number	Test Case Description	Test Data	Expected Results	Actual Results	Pass/Fail
1	Login with wrong/invalid credentials	Username: Root Password: Qwerty01	User should not be able to login	User did not login	***Pass***
2	Submit Information with a blank input	No data at on or all of the inputs fields	No information will be submitted	As expected	***Pass***
3	Using letters in numeric input fields	Using letter while setting number of times for pills	The input will not be filled	The input was not filled. It did not accept the word input.	***Pass***
4	Selection of past date for pill scheduling	Previous date in the calender as shown	The buttons will be inactive if tried to be clicked	Successful defense on the dates	***Pass***
5	Time addition on the exercise module at the 0 margin	Pause and continue buttons of the exercise	No new record of the exercise will be made in the fitness module. An increment of the counter of the exercise if it is existing or a new record of the exercise in then module	No new record of the exercise was made in the fitness module. .	***Pass***
6	Input validation; only valid input will be accepted and pushed to the database.	Enter medicine quantity as string in the med form	Reject the data and prompt you to enter data in the correct format.	Prompt the user to enter the data, product name in the correct format.	***Pass***

Table 13:Test Cases

4.3.4 Test Summary

In summary, the testing process was an integral part of the system development, ensuring that the system meets the quality standards specified and that it works as expected. The testing process helped to build quality into the system, demonstrate its working capabilities, and assess its progress and suitability.

Through the development of the system and the subsequent testing, it can be concluded that the system has been able to solve the client's problem to a significant extent. The system has successfully met the design requirements and has been tested to ensure that it functions as intended.

The system provides a user-friendly interface and allows for easy communication between user and the help facility as well as user interaction with the application. The different modules of the system have been tested and proven to function as expected, with the expected results obtained in all the 6 test cases.

CHAPTER FIVE: CONCLUSIONS AND RECOMMENDATIONS

5.0 Introduction

The primary aim of this project was to create a comprehensive cross-platform mobile application suitable for general use by individuals. The project commenced with an evaluation of existing mobile health applications and an identification of user needs. These requirements served as the basis for defining both functional and non-functional requirements, which guided the design of the prototype solution. This chapter summarizes the conclusions drawn by the development team upon implementing the solution and provides recommendations for future enhancements and the system's future roadmap.

5.1 Conclusion

In conclusion, the development of the cross-platform mobile application "Smart Afya" has been a significant endeavor aimed at enhancing healthcare management and promoting well-being. The application's diverse modules, including medical reminders, article surfing, diet recommendations based on diseases, and exercise categorization by age and illnesses, have been designed to cater to the specific needs of users in a user-friendly and accessible manner.

Through the implementation of medical reminders, users are empowered to stay on track with their healthcare routines and never miss important medication or appointment schedules. The article surfing feature provides users with a wealth of health-related information, enabling them to stay informed and make educated decisions regarding their well-being.

The diet recommendation module, tailored to different diseases, ensures that users have access to personalized nutritional guidance based on their specific health conditions. This feature aims to promote healthy eating habits and support individuals in managing their dietary requirements effectively.

Additionally, the exercise categorization feature takes into account users' age and illnesses, providing them with exercise routines that are safe, effective, and suitable for their unique circumstances. By offering tailored exercise programs, the application aims to encourage regular physical activity and improve overall fitness levels.

Throughout the development process, user experience and user interface design have been prioritized to ensure a seamless and intuitive interaction with the application. The cross-platform nature of "Smart Afya" allows users to access its functionalities across different devices and operating systems, maximizing its reach and convenience.

In summary, the "Smart Afya" mobile application serves as a comprehensive tool for health management, providing users with a range of features and functionalities to support their well-being. By incorporating medical reminders, article surfing, diet recommendations, and personalized exercise routines, the application aims to empower users to take control of their health, make informed decisions, and lead healthier lifestyles.

5.2 Recommendations

Based on the development and evaluation of the "Smart Afya" mobile application, the following recommendations can be made to further enhance its functionality and user experience:

1. Continuous Improvement and Updates: It is essential to maintain regular updates and improvements to the application based on user feedback and emerging technologies. This includes fixing bugs, adding new features, and enhancing existing functionalities to meet evolving user needs.

2. Expand Disease Database: Consider expanding the disease database for diet recommendations to include a wider range of conditions. Collaborating with healthcare professionals and nutritionists can help ensure accurate and up-to-date information for users.

3. Integration with Wearable Devices: Explore the integration of "Smart Afya" with wearable devices such as fitness trackers or smartwatches. This integration would enable users to track their physical activity, vital signs, and health data seamlessly, providing a comprehensive view of their overall health.

4. Gamification and Rewards: Introduce gamification elements within the application to make health management and exercise more engaging. Implementing challenges, achievements, and rewards can motivate users to stay consistent with their health routines and foster a sense of accomplishment.

5. Localization and Language Support: Consider providing localization options and language support to cater to a wider audience. Adapting the application's content and interface to different regions and languages can enhance its usability and accessibility for users worldwide.

6. Collaboration with Healthcare Providers: Establish partnerships with healthcare providers, clinics, and hospitals to enable seamless integration with their systems. This collaboration can facilitate better patient-doctor communication, appointment scheduling, and data sharing for a more holistic healthcare experience.

7. Privacy and Security: Continuously prioritize user privacy and data security. Implement robust security measures and adhere to industry standards to safeguard users' personal health information and ensure their trust in the application.

8. User Education and Support: Provide comprehensive user education resources, including tutorials, FAQs, and in-app support, to help users navigate the application effectively. Clear and accessible guidance will empower users to make the most of the application's features and functionalities.

By implementing these recommendations, "Smart Afya" can continue to evolve into a highly valuable and user-centric mobile application, supporting individuals in their health management journey and promoting overall well-being.

Appendix A: References

1. Krupinski EA, et al. (2016) "Use of a consumer health mobile application: A crosssectional study." Journal of Medical Internet Research. 18(3):e52.

2. Frimming, R., Polsgrove, M. and Bower, G. (2011). Evaluation of a Health and Fitness Social Media Experience. *American Journal of Health Education*, 42(4), pp.222-227.
3. Riehemann, K., Schneider, S. W., Luger, T. A., Godin, B., Ferrari, M., & Fuchs, H. (2009). Nanomedicine-challenge and perspectives. *Angewandte Chemie International Edition*, *48*(5), 872-897.
4. Nosrati, M., Karimi, R., & Hasanvand, H. A. (2012). Mobile computing: principles, devices and operating systems. *World Applied Programming*, *2*(7), 399-408.
5. Ohayon, M. M. (2002). Epidemiology of insomnia: what we know and what we still need to learn. *Sleep Medicine Reviews*, *6*(2), 97-111.
6. Leonard, T. C., Richard, H., Thaler, C. R., & Sunstein, N. (2008). Improving decisions about health, wealth, and happiness. *Proceeding of Pervasive Technologies Related to Physical Health*, *1*(3), 11-18.
7. Hughes, C. M. (2004). Medication non-adherence in the elderly. *Drugs & Aging*, *21*(12), 793811.
8. Groves, P., Kayyali, B., Knott, D., & Van Kuiken, S. (2013). The big data revolution in healthcare. *McKinsey Quarterly*, 2(3), 112-121.

9. Fong A, et al. (2017) "Evaluating the accuracy of a symptom checker application for self-diagnosis: a cross-sectional study." Journal of Medical Systems. 41(12):170.

10. Saveski S, et al. (2018) "The Impact of a Mobile Health App on Chronic Disease Management: A Randomized Controlled Trial." Journal of Medical Internet Research. 20(6):e170.

11. Fox, S., Jones, S. B., & Thomson, M. (2015). Health information seeking and use: the role of the Internet and health professionals. Journal of medical Internet research, 17(3), e50.

12. Kontopantelis, E., Ashworth, M., Reed, N., & Smith, J. (2015). The strengths and weaknesses of the WebMD symptom checker. Journal of medical Internet research, 17(3), e52.

13. World Health Organization, 2020b :External situation report

14. Millstein, R. A., Nguyen, T., & Kvedar, J. C. (2017). Medisafe: a medication management and symptom tracking mobile application. Journal of medical Internet research, 19(2), e36.

15. Huang, C., Lee, J. M., Kim, J., & Park, S. (2016). The use of mobile health applications for weight loss: a systematic review. Journal of medical Internet research, 18(9), e232.

16. Kamal, M., Khan, R., & Al-Qahtani, A. (2019). Impact of mobile health apps on health outcomes: A systematic review. Journal of Medical Systems, 43(10), 210.

17. Kim, J., Kim, Y., & Kim, J. (2019). The effectiveness of mobile health applications for self-management of chronic conditions: A systematic review. Journal of Medical Internet Research, 21(4), e13158.

18. Kullberg J, et al. (2018) "A systematic review of the quality of mobile health apps for diabetes management." Journal of Medical Internet Research. 20(11):e12087.

19. Kansagara D, et al. (2019) "Diagnostic Accuracy of Symptom Checkers for Commonly Occurring Conditions: Systematic Review." Journal of American Medical Association. 322(9):834-843.

20. Ng J, et al. (2018) "Privacy and security in mobile health: A systematic review." Journal of Medical Internet Research. 20(11):e11057.

21. Zanjal, S. V., & Talmale, G. R. (2016). Medicine reminder and monitoring system for secure health using IOT. *Procedia Computer Science*, *78*(5), 471-476.

22. Zao, J. K., Wang, M. Y., Tsai, P., & Liu, J. W. (2010). Smart phone based medicine in-take scheduler, reminder and monitor. *In the Proceeding of the e-Health Networking Applications and Services (Healthcom), 2010 12th IEEE International Conference on*, *23*(8), 162-168.

23. "World Health Organization The global burden of disease: 2004 update". (2013, May).(Online).Available:http://www.who.int/healthinfo/global_burden_disease/GBD_report_2004update_full.

24. United States. Department of Health and Human Services, 1996. *Physical activity and health: a report of the Surgeon General*. diane Publishing.

25. World Health Organization, 2000. *Obesity: preventing and managing the global epidemic* (No. 894). World Health Organization.

Appendix B: Timeline of Activities (Gantt Chart)

ACTIVITY		PERIOD YEAR 2022 - 2023																				
SN		November			December			January			February			March			April				
1	Expression of need																					
2	Requirement Analysis																					
3	Technical Specifications																					
4	Design																					
5	Verification																					
6	Prototyping																					
7	Testing																					
8	Sketching diagrams																					
9	Progress presentation and submission																					
10	Refinement																					
11	Deployment																					
12	Documentation																					
	Submission																					

Table 14: Gantt Chart

Appendix C: Budget Incurred

SN	Required Item	Unit	Quantity	Unit Price	Total Price
1	Laptop	Each	1	Kshs.22000	Kshs.22000
2	Flash Disk	32 GB	1	Kshs.2500	Kshs.2500
3	Internet charges	monthly	5	Kshs.500	Kshs.2500
4	Software Installations	each	3	Kshs.500	Kshs.1500
5	DVD costs	each	1	Kshs.100	Kshs.100
6.	Printing and photocopying	page	150	Kshs.10	Kshs.1500
7	Report binding	each	2	Kshs.200	Kshs.400
8	Contingency	20% of total			Kshs.5200
TOTAL				**Kshs.31200**	

Table 15: Budget Cost estimation

Appendix D: Code Snippets

1. **Splash Sreen**

```
import 'package:flutter/material.dart';

class SplashScreen extends StatefulWidget {
  @override
  _SplashScreenState createState() => _SplashScreenState();
}

class _SplashScreenState extends State<SplashScreen> {
  @override
  void initState() {
    super.initState();
    _navigateToHome();
  }

  void _navigateToHome() async {
    await Future.delayed(Duration(seconds: 5)); // Simulate a loading process
    Navigator.pushReplacementNamed(context, '/home');
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Colors.black,
      body: Stack(
        children: [
          Center(
            child: Column(
              mainAxisAlignment: MainAxisAlignment.center,
              crossAxisAlignment: CrossAxisAlignment.center,
              children: [
                Text(
                  'Smart Afya',
                  style: TextStyle(
                    color: Colors.white,
                    fontSize: 48,
                    fontWeight: FontWeight.bold,
                  ),
                ),
                SizedBox(height: 16),
                Image.asset(
                  'assets/mainlogo.png',
                  width: 200,
                  height: 200,
                ),
                SizedBox(height: 16),
```

```
                Text(
                  'Your health, your pride',
                  style: TextStyle(
                    color: Colors.white,
                    fontSize: 24,
                  ),
                ),
              ],
            ),
          ),
          Positioned(
            bottom: 16,
            left: 0,
            right: 0,
            child: Column(
              children: [
                Text(
                  'Dev:@Ezekiel Maina',
                  style: TextStyle(
                    color: Colors.white,
                    fontSize: 16,
                  ),
                ),
                SizedBox(height: 8),
                Row(
                  mainAxisAlignment: MainAxisAlignment.center,
                  children: [
                    LoadingBox(),
                    SizedBox(width: 16),
                    LoadingBox(),
                    SizedBox(width: 16),
                    LoadingBox(),
                  ],
                ),
              ],
            ),
          ),
        ],
      ),
    );
  }
}

class LoadingBox extends StatefulWidget {
  @override
  _LoadingBoxState createState() => _LoadingBoxState();
}
```

```
class _LoadingBoxState extends State<LoadingBox>
    with SingleTickerProviderStateMixin {
  late AnimationController _controller;

  @override
  void initState() {
    super.initState();
    _controller = AnimationController(
      vsync: this,
      duration: Duration(milliseconds: 1000),
    )..repeat(reverse: true);
  }

  @override
  Widget build(BuildContext context) {
    return ScaleTransition(
      scale: Tween(begin: 0.5, end: 1.0).animate(_controller),
      child: Container(
        width: 16,
        height: 16,
        decoration: BoxDecoration(
          color: Colors.white,
          borderRadius: BorderRadius.circular(4),
        ),
      ),
    );
  }

  @override
  void dispose() {
    _controller.dispose();
    super.dispose();
  }
}
```

2. Add Medicine Form

```
import 'package:flutter/material.dart';
import 'package:flutter_time_picker_spinner/flutter_time_picker_spinner.dart';

class AddMedicineInputForm extends StatefulWidget {
  const AddMedicineInputForm({Key? key}) : super(key: key);

  @override
  _AddMedicineInputFormState createState() => _AddMedicineInputFormState();
}
```

```
class _AddMedicineInputFormState extends State<AddMedicineInputForm> {
  final _formKey = GlobalKey<FormState>();
  int? _timesToTake;
  List<TimeOfDay> _selectedTimes = [];
  TimeOfDay? _reminderTime;

  @override
  Widget build(BuildContext context) {
    return AlertDialog(
      title: Text('Add Medicine'),
      content: Form(
        key: _formKey,
        child: Column(
          mainAxisSize: MainAxisSize.min,
          children: [
            TextFormField(
              decoration: InputDecoration(labelText: 'Times to take'),
              keyboardType: TextInputType.number,
              validator: (value) {
                if (value == null || value.isEmpty) {
                  return 'Please enter a number';
                }
                if (int.tryParse(value) == null) {
                  return 'Please enter a valid number';
                }
                return null;
              },
              onSaved: (value) {
                _timesToTake = int.parse(value!);
              },
            ),
            SizedBox(height: 16),
            Text('Selected times: ${_selectedTimes.join(', ')}'),
            SizedBox(height: 16),
            Expanded(
              child: ListView.builder(
                itemCount: _timesToTake ?? 0,
                itemBuilder: (BuildContext context, int index) {
                  return ListTile(
                    title: Text('Time ${index + 1}'),
                    trailing: TimePickerSpinner(
                      is24HourMode: false,
                      normalTextStyle: TextStyle(fontSize: 18),
                      highlightedTextStyle:
                          TextStyle(fontSize: 22, color: Colors.blue),
                      spacing: 50,
                      itemHeight: 40,
                      isForce2Digits: true,
```

```
                onTimeChange: (time) {
                  setState(() {
                    _selectedTimes[index] = TimeOfDay.fromDateTime(time);
                  });
                },
              ),
            );
          },
        ),
      ),
      SizedBox(height: 16),
      Text(
          'Reminder time: ${_reminderTime?.format(context) ?? 'Not set'}'),
      SizedBox(height: 16),
      ElevatedButton(
        onPressed: () async {
          if (_formKey.currentState!.validate()) {
            _formKey.currentState!.save();
            setState(() {
              _selectedTimes = List.generate(
                  _timesToTake!, (_) => TimeOfDay(hour: 0, minute: 0));
            });
          }
        },
        child: Text('Save'),
      ),
      SizedBox(height: 16),
      ElevatedButton(
        onPressed: () async {
          final time = await showTimePicker(
              context: context,
              initialTime: _reminderTime ?? TimeOfDay.now());
          if (time != null) {
            setState(() {
              _reminderTime = time;
            });
          }
        },
        child: Text('Select reminder time'),
      ),
    ],
  ),
),
actions: [
  TextButton(
    onPressed: () => Navigator.of(context).pop(),
    child: Text('Cancel'),
  ),
```

```
        ElevatedButton(
          onPressed: () {
            // TODO: Add medicine to database or state
            Navigator.of(context).pop();
          },
          child: Text('Save'),
        ),
      ],
    );
  }
}
```

3. Log in code

```
import 'package:flutter/material.dart';
import 'signup.dart';
import 'forgotpassword.dart';
import 'main.dart';

class LoginScreen extends StatefulWidget {
  const LoginScreen({Key? key}) : super(key: key);

  @override
  _LoginScreenState createState() => _LoginScreenState();
}

class _LoginScreenState extends State<LoginScreen> {
  final _formKey = GlobalKey<FormState>();
  final _emailController = TextEditingController();
  final _passwordController = TextEditingController();
  bool _showPassword = false;
  static bool _showdisplay = false;
  String _email = '';
  String _pass = '';

  @override
  void dispose() {
    _emailController.dispose();
    _passwordController.dispose();
    super.dispose();
  }

  void _handleLogin() {
    if (_formKey.currentState!.validate()) {
      // Login logic here
      print(
                'Logging in with email: ${_emailController.text} and password:
${_passwordController.text}');
```

```
  }
}

@override
Widget build(BuildContext context) {
  return SafeArea(
    child: Scaffold(
      body: Container(
        decoration: BoxDecoration(
          image: DecorationImage(
            image: AssetImage('assets/log.jpg'),
            fit: BoxFit.cover,
          ),
        ),
        child: Center(
          child: SingleChildScrollView(
            padding: const EdgeInsets.symmetric(horizontal: 32.0),
            child: Form(
              key: _formKey,
              child: Column(
                mainAxisAlignment: MainAxisAlignment.center,
                crossAxisAlignment: CrossAxisAlignment.stretch,
                children: [
                  Image.asset(
                    'assets/mainlogo.png',
                    width: 200,
                    height: 200,
                  ),
                  const SizedBox(height: 24.0),
                  TextFormField(
                    controller: _emailController,
                    keyboardType: TextInputType.emailAddress,
                    decoration: const InputDecoration(
                      labelText: 'Email',
                      border: OutlineInputBorder(),
                      prefixIcon: Icon(
                        Icons.email_outlined,
                        color: Colors.green,
                      ),
                    ),
                    validator: (value) {
                      if (value!.isEmpty) {
                        return 'Email is required';
                      }
                      if (!RegExp(r'^[\w-\.]+@([\w-]+\.)+[\w-]{2,4}$')
                          .hasMatch(value)) {
                        return 'Please enter a valid email address';
                      }
```

```
                    _email = value!;
                    return null;
                  },
                ),
                const SizedBox(height: 16.0),
                TextFormField(
                  controller: _passwordController,
                  obscureText: !_showPassword,
                  decoration: InputDecoration(
                    labelText: 'Password',
                    border: const OutlineInputBorder(),
                    prefixIcon: const Icon(
                      Icons.lock_outline_rounded,
                      color: Colors.green,
                    ),
                    suffixIcon: IconButton(
                      icon: Icon(
                        _showPassword
                            ? Icons.visibility_outlined
                            : Icons.visibility_off_outlined,
                      ),
                      color: Colors.green,
                      onPressed: () {
                        setState(() {
                          _showPassword = !_showPassword;
                        });
                      },
                    ),
                  ),
                  validator: (value) {
                    if (value!.isEmpty) {
                      return 'Password is required';
                    }
                    if (!RegExp(
                                                r'^(?=.*?[A-Z])(?=.*?[a-z])(?=.*?[0-
9])(?=.*?[!@#\$&*~]).{8,}$')
                        .hasMatch(value)) {
                      return 'Password must include at least 8 characters,\n1 upper
case letter, 1 lower case letter, 1 number\nand 1 special character (!@#\$&*~)';
                    }
                    _pass = value;
                    return null;
                  },
                ),
                const SizedBox(height: 24.0),
                ElevatedButton(
                  onPressed: () {
```

```
    _handleLogin();
    if (_email == 'admin@gmail.com' &&
        _pass == '@Admin001') {
      GoogleSearchPage.display = true;
      Navigator.push(
        context,
        MaterialPageRoute(
            builder: (context) => GoogleSearchPage()),
      );
    } else {
      _handleLogin();
    }
  },
  child: const Text('Log in'),
  style: ButtonStyle(
    backgroundColor:
        MaterialStateProperty.all<Color>(Colors.green),
  ),
),
const SizedBox(height: 16.0),
TextButton(
  onPressed: () {
    Navigator.push(
      context,
      MaterialPageRoute(
          builder: (context) => ForgotPasswordScreen()),
    );
    // Implement forgot password logic here
  },
  style: ButtonStyle(
    foregroundColor:
        MaterialStateProperty.all<Color>(Colors.black),
  ),
  child: const Text('Forgot Password?'),
),
const SizedBox(height: 16.0),
TextButton(
  onPressed: () {
    Navigator.push(
      context,
      MaterialPageRoute(
          builder: (context) => SignupScreen()),
    );
  },
  style: ButtonStyle(
    foregroundColor:
        MaterialStateProperty.all<Color>(Colors.black),
  ),
```

```
                    child: const Text('Dont have an account? Sign Up'),
                  ),
                ],
              ),
            ),
          ),
        ),
      ),
    );
  }
}
```

Appendix E: Attachments

[1] Project Student Log

[2] Sample mail Report

[3] CD Having Project Source code