VISVESVARAYA TECHNOLOGICAL UNIVERSITY

JNANA SANGAMA, BELAGAVI – 590018, KARNATAKA

A Mini Project Report on

"HEALTHY DIET"

Submitted in partial fulfillment of the requirements for the VI Semester of degree of ***Bachelor of Engineering in Information Science and Engineering*** *of Visvesvaraya Technological University, Belagavi*

Submitted by

R G MANVITHA	**1RN19IS111**
RAKSHITA S	**1RN19IS116**
ROHAN SANJAY SAVALGI	**1RN19IS121**

Under the Guidance of

Ms. Priyanka M R

Assistant Professor

Dept. of ISE, RNSIT

Department of Information Science and Engineering

RNS Institute of Technology

Dr. Vishnuvardhan Road, Rajarajeshwari Nagar Post,
Channasandra, Bengaluru – 560 098

2021 -2022

RNS Institute of Technology

Dr. Vishnuvardhan Road, Rajarajeshwari Nagar Post,

Channasandra, Bengaluru – 560098

DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING

CERTIFICATE

This is to certify that the mini project report entitled **"HEALTHY DIET"** has been successfully completed by **R G MANVITHA (1RN19IS111), RAKSHITA S (1RN19IS116)** and **ROHAN SANJAY SAVALGI (1RN19IS121),** bonafide students of **RNS Institute of Technology** in partial fulfillment of the requirements for the award of the degree in **Bachelor of Engineering in Information Science and Engineering of Visvesvaraya Technological University, Belagavi** during academic year **2021 – 2022.** The mini project report hasbeen approved as it satisfies the academic requirements as a part of Mobile ApplicationDevelopment Laboratory.

Ms. Priyanka M R	**Dr. Suresh L**	**Dr. M K Venkatesha**
Mini Project Guide	Professor and HOD	Principal
Assistant Professor	Department of ISE	RNSIT
Department of ISE	RNSIT	

External Viva

Name of the Examiners	**Signature with date**
1. _____	_____
2. _____	_____

DECLARATION

We, **R G MANVITHA [USN:1RN19IS111], RAKSHITA S [USN: 1RN19IS116]** and **ROHAN SANJAY SAVALGI [USN: 1RN19IS121],** students of VI Semester BE, in Information Science and Engineering, RNS Institute of Technology hereby declare that the Project entitled "**HEALTHY DIET"** has been carried out by us and submitted impartial fulfillment of the requirements for the *VI Semester of degree of* ***Bachelor of Engineering in Information Science and Engineering*** *of Visvesvaraya Technological University, Belgavi* during academic year 2021-2022.

Place: Bengaluru
Date:

R G MANVITHA (1RN19IS111)
RAKSHITA S (1RN19IS116)
ROHAN SANJAY SAVALGI (1RN19IS121)

ABSTRACT

Generally, in case patients are supposed to be monitored continuously for their BMI, Water intake. In the earlier methods, the doctors need to be present physically or in several cases SMS will be sent using GSM. In the earlier case the history of the patient cannot be displayed, only current data is displayed. In the current project, we are using a novel idea for continuous monitoring patient's health conditions. The health care scheme is focus on the measurement and Monitoring various biological parameters of patient's body like BMI, Water tracking and Calorie tracker android application, where person can continuously monitor his/her condition on his smart phone using an Android application.

ACKNOWLEDGMENT

The fulfillment and rapture that go with the fruitful finishing of any assignment would be inadequate without the specifying the people who made it conceivable, whose steady direction and support delegated the endeavors with success.

We would like to profoundly thank **Management** of **RNS Institute of Technology** for providing such a healthy environment to carry out the project work.

We would like to express our sincere thanks to our beloved Principal **Dr. M K Venkatesha** for his support and inspired us towards the attainment of knowledge.

We wish to place on record our words of gratitude to **Dr. Suresh L,** Professor and Head of the Department, Information Science and Engineering, for being the enzyme and master mind behind our project work.

We would like to express our profound and cordial gratitude to our Mini Project Guide**, Ms. Priyanka M R,** Assistant Professor, Department of Information Science and Engineering for her valuable guidance, constructive comments, continuous encouragement throughout the Mini Project Work and guidance in preparing report.

We would like to thank all other teaching and non-teaching staff of Information Science & Engineering who have directly or indirectly helped us to carry out the project work.

Also, we would like to acknowledge and thank our parents who are source of inspiration and instrumental in carrying out this Mini Project Work.

Place: Bengaluru
Date:

R G MANVITHA **(1RN19IS111)**
RAKSHITA S **(1RN19IS116)**
ROHAN SANJAY SAVALGI **(1RN19IS121)**

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

API - Application Programming Interface

HTML - Hyper Text Markup Language

IDE - Integrated Development Environment

iOS - iPhone Operating System

Inc. - Incorporated

JVM - Java Virtual Machine

OS - Operating System

RAM - Random Access Memory

SDK - Software Development Kit

UI - User Interface

XML - Extensible Markup Language

Chapter 1

INTRODUCTION TO ANDRIOD

1.1 History

In past mobile phones were used only to make calls but with the introduction of smart phone the mobile phone has evolved to a low powered hand held processing system. This evolution was caused by the operating system for the mobile phones making them smart that have processing and storage of their own. Now the mobile provides numerous functionalities from calling to texting, multimedia sharing, emails, socializing applications, word processor, excel sheets to various multiplayer games and much more.

The operating system for this hand held devices are iOS by Apple Inc., Windows by Windows Inc. and Android by Google. Among the competitors in smartphone operating system industry Android holds the largest market share in terms of units shipped worldwide and number of users.

Android is an open source operating system based on linux kernel on which applications run on an application framework that controls the activities supported by the libraries and Dalvik virtual machine which compiles and converts all java class files into a single file. There can be number of virtual machines running simultaneously on a single device handling different applications or instances of an application.

Android operating system provides memory management, process management to the applications and services running. Each release of android improved user experience and brought enhanced features. In 2012 Android became the most popular operating system for mobile devices, surpassing Apple's iOS, and, as of 2020, about 75 percent of mobile devices run Android.

1.2 Android Versions

The development of the Android operating system was started in 2003 by Android, Inc. Later on, it was purchased by Google in 2005. The beta version of Android OS was released on November 5, 2007, while the software development kit (SDK) was released on November 12, 2007.

The first Android mobile was publicly released with Android 1.0 of the T-Mobile G1 (aka HTC Dream) in October 2008. The first Android version which was released under the numerical order format was Android 10.

Code name	Version numbers	API level	Release date
No codename	1.0	1	September 23, 2008
No codename	1.1	2	February 9, 2009
Cupcake	1.5	3	April 27, 2009
Donut	1.6	4	September 15, 2009
Eclair	2.0 - 2.1	5 - 7	October 26, 2009
Froyo	2.2 - 2.2.3	8	May 20, 2010
Gingerbread	2.3 - 2.3.7	9 - 10	December 6, 2010
Honeycomb	3.0 - 3.2.6	11 - 13	February 22, 2011
Ice Cream Sandwich	4.0 - 4.0.4	14 - 15	October 18, 2011
Jelly Bean	4.1 - 4.3.1	16 - 18	July 9, 2012
KitKat	4.4 - 4.4.4	19 - 20	October 31, 2013
Lollipop	5.0 - 5.1.1	21- 22	November 12, 2014
Marshmallow	6.0 - 6.0.1	23	October 5, 2015
Nougat	7.0	24	August 22, 2016
Nougat	7.1.0 - 7.1.2	25	October 4, 2016
Oreo	8.0	26	August 21, 2017
Oreo	8.1	27	December 5, 2017
Pie	9.0	28	August 6, 2018
Android 10	10.0	29	September 3, 2019
Android 11	11	30	September 8, 2020

Table 1.1 Android Versions

1.3 Architecture of Android

Android operating system is a stack of software components which is roughly divided into five sections and four main layers as shown below in the architecture diagram.

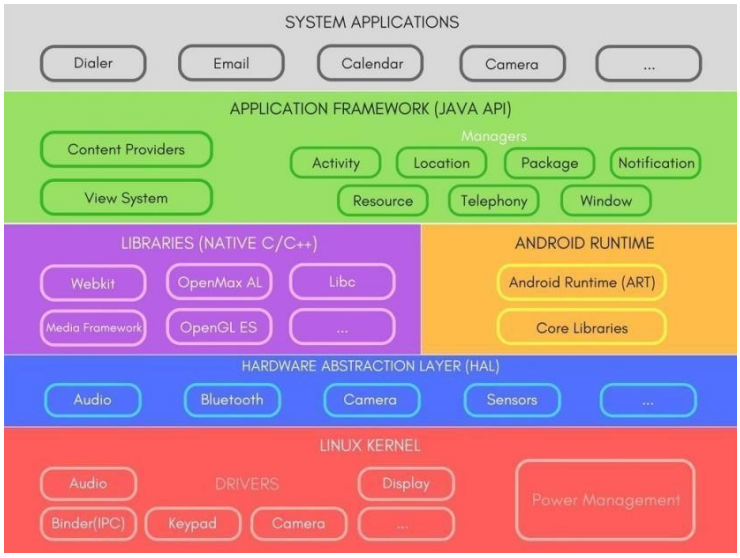

Figure 1.1 Android Architecture diagram

- **Linux kernel:** This is the layer at the very bottom of the Android architecture. All other layers run on top of the Linux kernel and rely on this kernel to interact with the hardware. This layer contains all the essential hardware drivers which help to control and communicate with the hardware. It provides the basic functionality like Process Management, Memory Management and Device Management like Camera, Display, and Flash etc.

- **Libraries:** This is a set of common functions of the application framework that enables the device to handle different types of data. Some of the most important set of libraries that are included are – Web kit which is the browser engine to display HTML, OpenGL used to render 2- D or 3-D graphics on to the screen, SQLite which is a useful repository for storing and sharing of application data.

  A summary of some key core Android libraries available to the Android developer is as follows

  - *android.app* − Provides access to the application model and is the cornerstone of all Android applications.

- *android.content* − *Facilitates content access, publishing and messaging between* applications and application components.
- *android.database* − Used to access data published by content providers andincludes SQLite database management classes.
- *android.opengl* − A Java interface to the OpenGL ES 3D graphics rendering API.
- *android.os* − Provides applications with access to standard operating systemservices including messages, system services and inter-process communication.
- *android.text* − Used to render and manipulate text on a device display.
- *android.view* − The fundamental building blocks of application user interfaces.
- *android.widget* − A rich collection of pre-built user interface components such as buttons, labels, list views, layout managers, radio buttons etc.
- *android.webkit* − A set of classes intended to allow web-browsing capabilities to be built into applications.

❖ **Android Runtime:** The Android runtime mainly consist of the Dalvik Virtual Machine (DVM). DVM is very much like the standard Java Virtual Machine (JVM) except that it is optimized for mobile devices that have low processing power and low memory. DVM generates a.dex file from the .class file at compile time and provides higher efficiency in low resources devices. Each application has its own process and an instance of DVM. Android runtime also provides core libraries that enable the Android developers to create applications using the Java language.

❖ **Application Framework:** The Android runtime mainly consist of the Dalvik Virtual Machine (DVM). DVM is very much like the standard Java Virtual Machine (JVM) except that it is optimized for mobile devices that have low processing power and low memory. DVM generates a.dex file from the .class file at compile time and provides higher efficiency in low resources devices. Each application has its own process and an instance of DVM. Android runtime also provides core libraries that enable the Android developers to create applications using the Java language.

❖ **Applications:** This is the topmost layer in the architecture and the layer where the application that we develop fits in. This layer provides several pre-installed applicationsthat are default for certain things like Contacts Books, Browser etc.

1.4 Android Studio Installation

Android Studio is the official integrated development environment (IDE) for Android application development. It is based on the IntelliJ IDEA, a Java integrated development environment for software, and incorporates its code editing and developer tools.

To support application development within the Android operating system, AndroidStudio uses a Gradle-based build system, emulator, code templates, and Github integration. Every project in Android Studio has one or more modalities with source code and resource files. These modalities include Android app modules, Library modules, and Google App Engine modules.

PROCEDURE TO BE FOLLOWED TO DOWNLOAD AND INSTALL ANDROIDSTUDIO:

STEP 1: Android Studio and the Software Development Kit can be downloaded directly from any web browser using the below link.

https://developer.android.com/studio

STEP 2: Android Studio is available for Mac, Windows, and Linux desktop platforms.

Windows

To install Android Studio on Windows, proceed as follows:

i. If you downloaded an .exe file (recommended), double-click to launch it. If you downloaded a .zip file, unpack the ZIP, copy the android-studio folder into your Program Files folder, and then open the android-studio > bin folder and launch studio64.exe (for 64-bit machines) or studio.exe (for 32-bit machines).

ii. Follow the setup wizard in Android Studio and install any SDK packages that it recommends.

Mac

To install Android Studio on your Mac, proceed as follows:

i. Launch the Android Studio DMG file.

ii. Drag and drop Android Studio into the Applications folder then launch AndroidStudio.

iii. Select whether you want to import previous Android Studio settings, and then click OK.

iv. The Android Studio Setup Wizard guides you through the rest of the setup, which includes downloading Android SDK components that are required for development.

Linux

To install Android Studio on Linux, proceed as follows:

i. Unpack the .zip file you downloaded to an appropriate location for your applications, such as within /usr/local/ for your user profile, or /opt/ for shared users. If you're using a 64-bit version of Linux, make sure you first install the required libraries for 64-bit machines.

ii. To launch Android Studio, open a terminal, navigate to the android-studio/bin/ directory and execute studio.sh.

iii. Select whether you want to import previous Android Studio settings or not, then click OK.

iv. The Android Studio Setup Wizard guides you through the rest of the setup, which includes downloading Android SDK components that are required for development.

Chapter 2

INTRODUCTION TO PROJECT

2.1 Overview of the Project

Android Studio IDE

Nutrition point app source code is built in Android Studio IDE. This android studio IDE helps us in various different processes like designing of UI, coding of logic for the application , build and run the application. This complete execution of program from the start to finish is made easy by the IDE.

Splash Screen

Show a splash screen to user when user start your Android App. This screen is colorful and attractive in order to grab the attention of the user and enables the user to explore the application in detail. This screen stays for 3 seconds and returns to home page after the page is loaded.

API

API's is used for fetching data from backend server. We used it to calculate calories for all food types and fetch hundreds of food recipes compatible with all Diet types and health cases.

Navigation Drawer Menu

In Navigation Drawer menu the user will see many options to get most out of the App and moving between activities and Fragments. This provides easy ready to use navigation bar which is available throughout the application because it's a overlay and is displayed on the top of all other screens.

Advance Push Notification

Nutrition Point App can send notifications for different Reminders. It calculates the amount of water your Body needs and the periods between notifications. Also the times of your diet meals during the day.

Exit Confirmation Toast

When user try to exit the App, the confirmation toast will be displayed on screen asking user to click once again back button to exit. This prevents the user to accidently exit out the application.

Scroll View

Nutrition Point content Vertically Scrolling. make the content vertically scrollable.

2.2 Aim of the Project

The aim of the project is to ensure everybody has an efficient and prosperous life. It keepsa track of the daily activities and helps the user to maintain a healthy and nutritious journey.

BMI: BMI is a useful measure of overweight and obesity. It is calculated from our height and weight. BMI is an estimate of body fat and a good gauge of our risk for diseasesthat can occur with more body fat.

Water remainder notification: will help us to calculate, how much water our body needs, will track our hydration and gently remind you to drink water to fulfill your goal. Improve your health with personal drink notification.

Calorie tracker: Calorie tracking apps have made it much easier to count calories. Typically, such apps gather basic information such as age, gender, activity levels, goal (weightless/gain/maintenance) and how quickly you want to achieve our goal, when determining a net calorie recommendation.

.

Chapter 3

SYSTEM DESIGN

3.1 System Requirements

Hardware Requirements:

- Processor : Pentium IV or above
- RAM : GB or more
- Hard Disk : 2GB or more

Software Requirements:

- Operating System : Windows 7 or above
- IDE : Android Studio - Chipmunk
- API Level : 19 or above

3.2 User Interface

XML Code for the initial UI screen:

```
<?xml version="1.0" encoding="utf-8"?>

<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".SplashActivity">

    <ImageView
        android:id="@+id/splashIV"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:foregroundGravity="center"
        app:srcCompat="@drawable/splash" />
```

```
</android.support.constraint.ConstraintLayout>
```

Figure 3.1 Main Screen UI

Chapter 4

IMPLEMENTATION

4.1 Implementation of BMI Calculator

```
bmi = (Integer.valueOf(weight) * 10000) /
(Integer.valueOf(height) * Integer.valueOf(height));
tv1.setText("Your BMI: " + Math.round(bmi * 10d) / 10d);
tv1.setTypeface(tf);
if (bmi >= 30){
    tv2.setText("Obesity");
}else if ((bmi >= 25) && (bmi < 30)){
    tv2.setText("Overweight");
}else if (bmi <= 18){
    tv2.setText("Under Weight");
}else if((bmi > 18) && (bmi < 25) ){
    tv2.setText("Normal");
}
seekBar = view.findViewById(R.id.seekBar);
seekBar.setProgress((int)Math.round(bmi));
```

For the working of the application, we have to calculate the BMI(body mass index) of a person based on the height and weight of a particular person. There are steps in which this has to be done.

1. The Input regarding the weight and height of the individual has to be taken, and this can be done by having an EditText fields and fetching data from them.
2. After we input from both the fields, we can calculate the BMI by using the formula - (weight * 10000) / (height * height).
3. Now this BMI value is shown inside the app, with the help of a TextView.
4. Seeing the value of BMI, a person can be categorized into Overweight, Normal and Underweight.
5. All this categories can also be shown to the user by using the TextView.

4.2 Implementation of Water Tracker

```
if (Integer.valueOf(age) <= 30){
        water = (Integer.valueOf(weight) * 42 * 2.95) /
(28.3 * 100);}

else if (Integer.valueOf(age) > 30 &&
Integer.valueOf(age) <= 35){
        water = (Integer.valueOf(weight) * 37 * 2.95) /
(28.3 * 100);}

else if (Integer.valueOf(age) > 35){

water = (Integer.valueOf(weight) * 32 * 2.95) / (28.3 *
100);
}
    tv6.setText("You need: " + (Math.round(water * 10d)
/ 10d) + " L/day");

    return view;
}
```

The water intake is calculated on the basis of the age of the individual. The water intake is further classified and calculated as to how much water is needed for that individual. The formula which is applied to calculate the water is split into 3 sections:

1. If the age is less than 30 or equal to 30, the formula applied is
   water Intake = (weight * 42 * 2.95) / (28.3 * 100)
2. If the age is between 30 and 35, in this case the formula is
   water Intake = (weight * 37 * 2.95) / (28.3 * 100)
3. At last if the age is greater than 35, the formula is used is
   water Intake = (weight * 32 * 2.95) / (28.3 * 100)

The water Intake value which is calculated in the previous step is updated and shown inside the textView so that it can be displayed to the user.

4.3 Implementation for Food Database API

In order to get information about the various food items, we fetch the information from a global database which is created and integrated by a team. In order to fetch data from this database we use free and open Api of that database.

This is how we can find the various food items and get all information about that food.

```
URL myUrl = new
URL("https://api.nutritionix.com/v1_1/search/" + food
+"?fields=item_name%2Citem_id%2Cnf_calories%2Cnf_total_fat"
+"&appId=3fe5fa47&appKey=61729b9d2d8612a629467f0cdbbd6d2c");
HttpURLConnection connection =(HttpURLConnection)
myUrl.openConnection();
connection.setConnectTimeout(700);
connection.connect();
```

Chapter 5

TESTING

5.1 Unit Testing

Test case ID	Test cases	Action	Result
1	On clicking Male/Female buttons	Takes you to BMI calculation	Pass
2	Entering details: Age, Weight and Height. Press Analyze Data	Takes the user to the profile summary	Pass
3	Click Set Reminder	Takes to Water Reminder	Pass
4	Choose the time interval.	Sets the time interval	Pass
5	Click start reminder	Sets the starting time.	Pass
6	Click Stop reminder	Ends the water reminder	Pass
7	Click calculate calorie	Takes to calories calculator	Pass
8	Enter food item. Click calculate calories.	Displays the total calorie and fat.	Pass
9	Click show analysis.	Takes to profile summary	Pass
10	Click Restart Body Analysis,	Starts the whole analysis from home page.	Pass

Table 5.1 Unit Testing

Chapter 6

RESULTS

All the Activities provided in the application and its operations have been presented in as snapshots. A detailed view of all the snapshots of the application is given in this section.

Figure 6.1 Eat Clean Stay Positive

Fig 6.1 shows the first page which is shown to the user. Here we have a motivating statement quoting "EAT CLEAN STAY POSITIVE". This page will display for 3 seconds after which the user is taken to the next page.

In this page, as shown in *Fig 6.2* users has to select his/her gender which leads to physical activity page that asks for their exercise details.

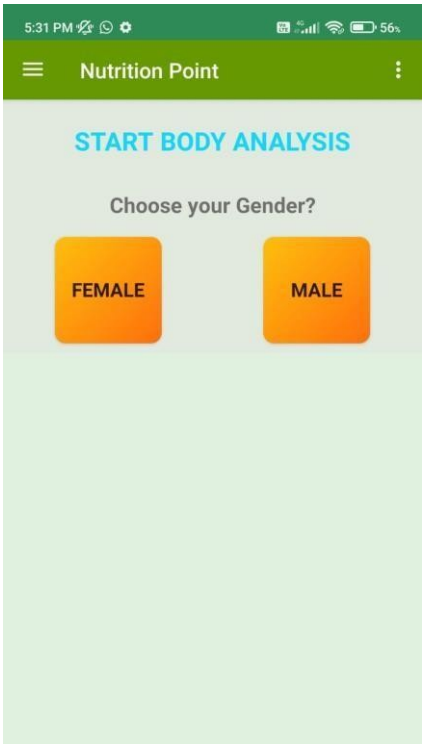

Figure 6.2 Start Page

Figure 6.3 Physical Activity

As described in *Fig 6.3,* the user has to choose the kind of physical activity they are actually doing, i.e., Almost no physical activity, Frequently walking, Exercise 1-2 days a week, Exercise 3-5 days a week or Exercise every day. Once chosen it leads to the next page that asks for the users details

In this page, i.e., *Fig 6.4*, the user must enter his/her details, i.e., age, weight and height. When the "Analyze Data" button is pressed it takes us to the next page where the BMI is calculated accordingly

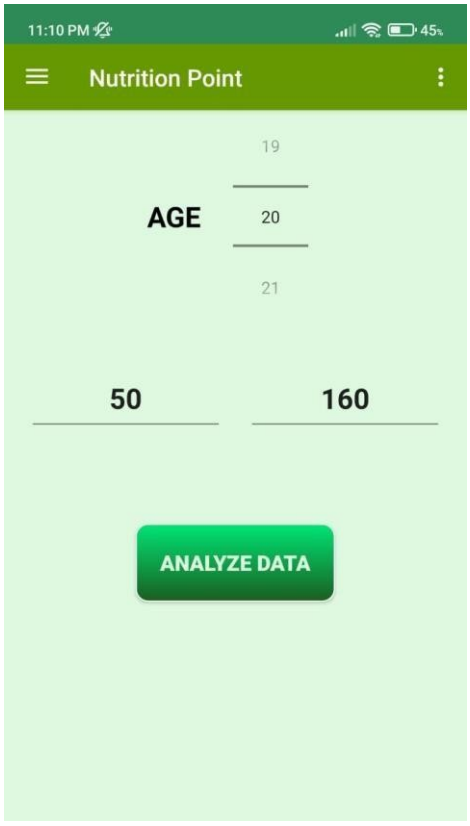

Figure 6.4 Entries of Details

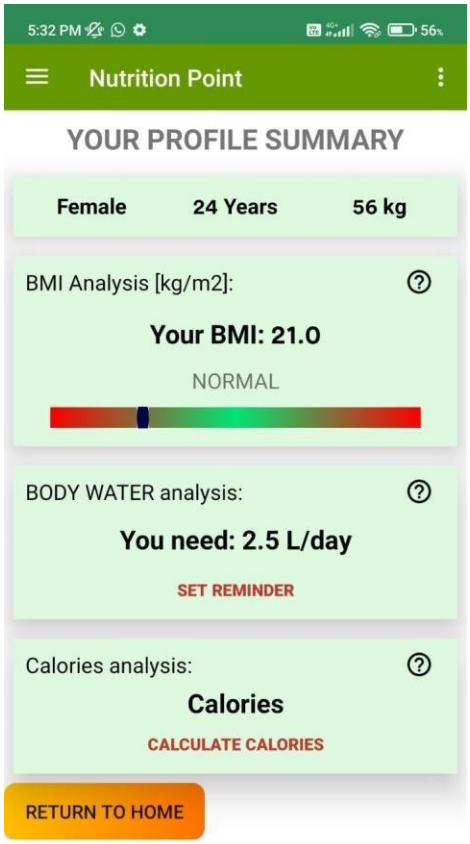

Figure 6.5 Profile Summary

The Profile Summary page as shown in *Fig 6.5,* shows the body analysis of the user based on the details entered. It also has water analyzer which tracks the water intake and a calorie analyzer, tracking the calories consumed.

In this page as shown in *Fig 6.6* the user has various time intervals to choose, for setting their water reminder; they have just start and stop the reminder, according to which the user will get an alarm.

Based on these reminders the user can control or regulate his/her water intake.

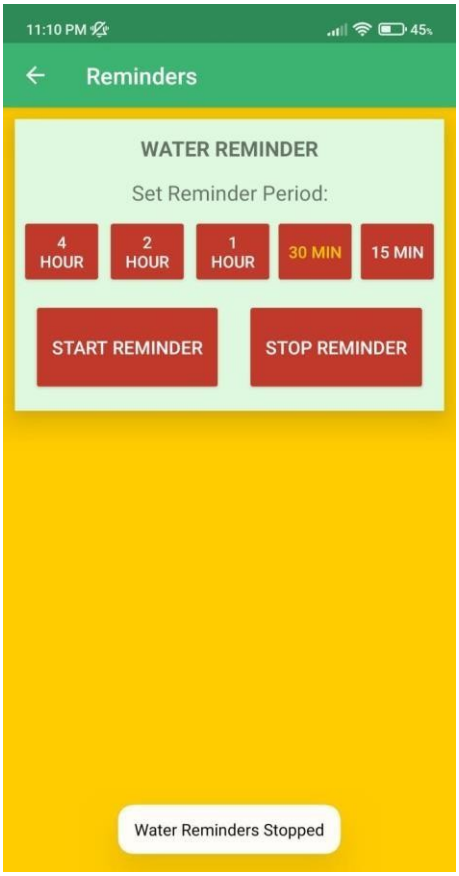

Figure .6 Reminders

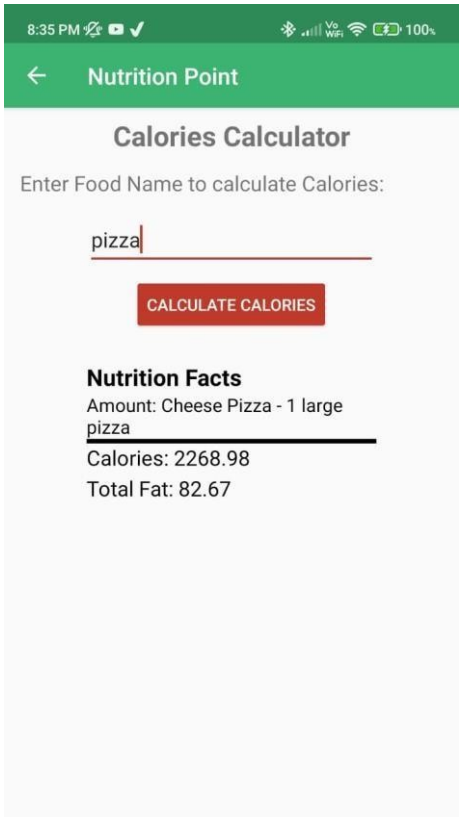

Figure 6.7 Calories Calculator

Calories calculator as described in *Fig 6.7*, helps the user to track the calories of food they have consumed or want to consume.

The user can enter any food item as per their wish; the calories and fat associated will be displayed.

The page in *Fig 6.8* is just an interface that leads to the Profile Summary if the "Show Analysis" button is pressed and leads to the main page on pressing the "Restart Body Analysis" button.

Figure 6.8 Navigation to Analysis or Restart

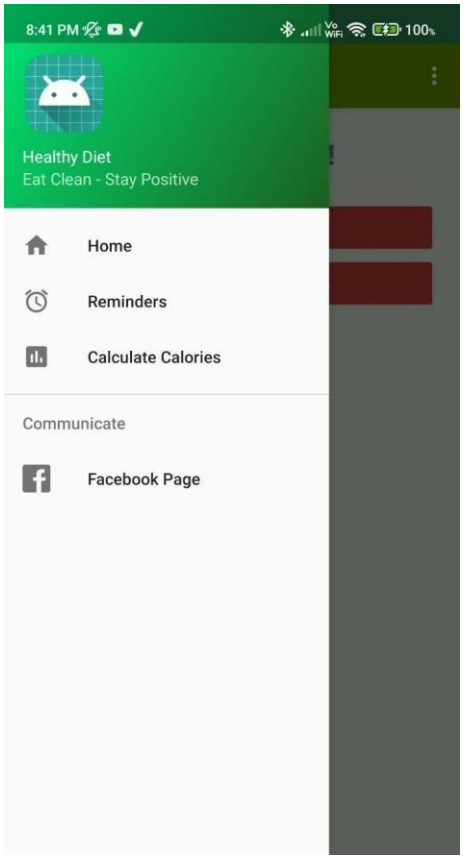

Figure 6.9 Navigation Menu

The *Fig 6.9* shows a navigation page from which the user can directly access the home page, calories calculator or water remainder page without following any order.

Chapter 7

CONCLUSION AND FUTURE ENHANCEMENT

7.1 Conclusion

In order to improve diet and nutrition and combat obesity in the general population, smartphone apps are anticipated to be an effective and affordable intervention. In general, it has been observed that measurements of diet and nutrition taken using mobile devices have good accuracy.

Applications that are quick to use, simple to administer, and that raise awareness of calorie intake and weight management are preferred by participants. Good diet has been shown to support academic performance and cognitive development in addition to greater physical health and a decreased risk of disease.

A balanced diet and appropriate eating habits should be instilled in children from an early age. Children should be taught the importance of consuming nutritious food and the negative impacts of junk food because such habits can't be changed quickly.

7.2 Future Enhancement

- Implementation of heart rate tracker.
- Count of footsteps by a person in a day.
- Customized workout sessions / yoga sessions.
- Usage of previous data in-order to create comprehensive reports.

REFERENCES

- Dawn Griffiths and David Griffiths, “Head First Android Development”, 1st Edition,O‟Reilly SPD Publishers, 2015.
- Google Developer Training, "Android Developer Fundamentals Course – Concept Reference”, Google Developer Training Team, 2017.
- https://www.tutorialspoint.com/android/android_architecture.htm
- https://www.javatpoint.com/android-versions