TotalHealth: An android application to help you achieve fitness goals.

A Project Report
Presented to
The Faculty of the College of
Engineering

San Jose State University
In Partial Fulfillment
Of the Requirements for the Degree
Master of Science in Computer Engineering
Master of Science in Software Engineering

By
Gaurav Chandra Debnath, Kajal Agarwal, Spriha Deshpande
Batch of May 2019.

APPROVED

DocuSigned by:

Gheorghi Guzun

F8287D169BF047E...

Gheorghi Guzun, Project Advisor

David Bruck, Director, MS Computer Engineering

Dan Harkey, Director, MS Software Engineering

Xiao Su, Department Chair

ABSTRACT

TotalHealth
By Gaurav Chandra Debnath, Kajal Agarwal, Spriha Deshpande

The rigorous professional environment in the current world scenario has led to diverse health issues in a person life. To maintain a balance in work schedule along with a healthy personal life, is of foremost importance. According to a research, approximately $147 billion is spent as an added medical expense per year within the United States of America. Current health anomalies can be prevented with social awareness and by self-motivation to achieve them.

Lack of legitimate resources leads to exhaust human time and money and complicates health goals. To learn about the vitals for health monitoring, itself is a compelling task. Furthermore, it is important to observe health parameters and analyze to avoid outbreak of any adverse impairments. It takes a good diet plan and a healthy work out to ensure a desirable lifestyle.

In this project, we propose to develop an Android application which allows users to venture a customizable self target or a goal, to attain fitness intent. The user will be able monitor their workout routine, along with running a dietary record and realize the ideal BMI (Body Mass Index) and WHR (Waist-Hip Ratio). The use of self-visualization and use of videos is in pursuit of encouraging the user to have a balanced diet and achieve their fitness goals. The aim of the application Total Health is to help users contain their fitness goals according to convenience and have a healthy balance with nourishment and routine.

Acknowledgments

The authors are deeply indebted to Prof. Gheorghi Guzun for his invaluable guidance, assistance and planning of this project, Prof. Dan Harkey for scheduling the timelines for proper implementation, and San Jose State University for providing an opportunity to making this a culminating experience.

Table of Contents

List of Figures

List of Tables

Chapter 1. Project Overview

1.1 Introduction

Health is always being a concern for Human Beings. The gradual deterioration of health in an environment of rigorous professional wing is of critical concern. Many factors are responsible for the degradation of a person's health. A few of the reasons include the type of food intake and lifestyle habits such as smoking and drinking. The need of the hour is to encourage healthy lifestyle choices, maintain the calorie intake, and a workout routine. Chronic diseases such as cardiovascular illnesses, cancers, chronic respiratory illnesses, and diabetes, are the primarily causes of death in the world. The mortality rate of these chronic diseases is of about 68% of the world's population[1,2]. Health concerns have led to the rise of applications that focus on systematic improvement of physical and mental health. Applications that help in regularizing the health stability are anticipatedly growing in popularity and informs their respective user of channelizing their energy into a healthy diet and workout regime. It is equally essential for people with chronic ailments to monitor their vital ailments [1].

The prime objective of this project is to develop an android based application for mobile devices to acquaint users with regard to tracking their calorie intake and burn-out along with maintaining their nutrient needs. Some of the features that a fitness application should have are customized diet plans, workout programs uniquely designed for the user, daily activity tracking, community based aspect and a simple user interface design.

Android based platform is preferred because phones with this platform is available to larger masses due to its low cost and reasonability. Besides, these phones are easy to operate to a widespread of age groups without any commotion [3].

1.2 Proposed Areas of Study and Academic Contribution

Moving forward with the proposition of implementing a regression based model in the application Total Health, we here define chief dependent values i.e. Body Mass Index (BMI) and Waist to Hip Ratio. These two parameters are cellulared by the dynamic user input and also present an ideal value based on the surveys conducted and data sets taken to advantage [4]. The experimental data is as suggested by Kaggle [5] and more surveys are contemplated using the Google forms within people of various age groups. In this project report, we devote our research into how mobile medical industry has reached every corner of the world. A development of a User Interface siding with Android Framework is an important task since it is responsible for the information exchange between the device and user. A UI is responsible for converting the information into the internal system and generate the output to present to the user[6]. A methodology is by its very nature, demanding. This requires continuous efforts to develop and maintain the application in an environment suitable for arduous and frequent software updates and installations. The Android and Server terminals are connected through a intermediate transport service based on which, output results are obtained [5,6]. We have the User Interface Control Layer allied with the System Service Layer that is accountable for data services and networking. All this is combined with the Data Access Layer responsible for accessing data through Databases. The whole system is therefore connected to a server providing obligated services [6].

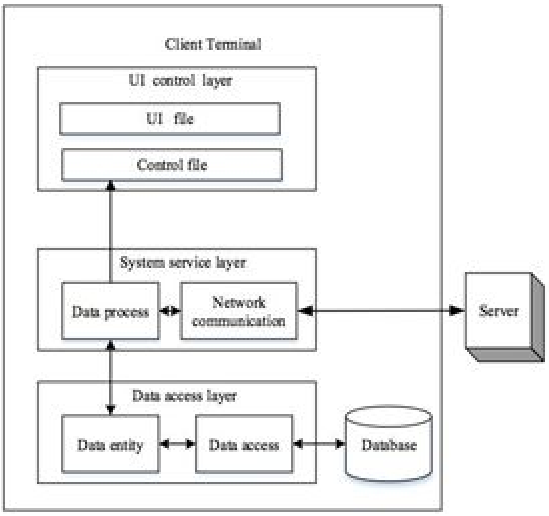


Figure 1. Planned Design of Proposed System Architecture

The design and implementation is based on client-server architecture and compelling algorithm like Weighted Majority in Machine Learning which is designed to provide the user to realize their accuracy with least of mistakes in consideration [7]. The growing need for open source order to have a 2D/3D visualization can be used to explain a complex solution and concepts [8]. This is to design an API with high level 3D graphics/videos for motivational goals in support of Android platform [9].

Significant tasks in the application Total Health is to create a user profile going forth with customizable goals/target with the target steps and calories. The goal is classified as an accomplished one based on the legitimate parameters provided by the user. In addition to the legitimacy, implementation of sensor based on geolocation access helps the user guide through the progress in their routine and workout. The use of predictive analysis combats the challenges to provide user with details with regard to their respective Body-Mass-Index and Waist-to-Hip ratio. The customized goals and targets aids to an interactive application for user contentment.

Going forth with the development phase, the application is developed on an Android SDK (Software Development Kit) with Java programming language. SDK is a fast, reliable, and a free open-source with different features made available with different libraries and tools[11]. Data sets are acquired by Kaggle [5] and surveys conducted using Google forms.

Databases and UI are integrated within the architecture with the help of an Android Framework. Different function calls, protocols, and returning peculiar data types will be maintained by RESTful API [12] and Google API as well. The purpose of this application is to make use of the cutting-edge technologies and implement it to provide an easy-go-goal for the user that is not been provided so far. To attain this exceptional objective, we will gain ground with Machine Learning based Weighted Majority Algorithm for having a predictive analysis on the ideal key values to recommend a user with the amount of time and energy they would need be of [13].

Legitimacy is an important concern while we play with the Application. The user goals are dedicated to benefit themselves. By means of providing a user data is to accomplish goals and get back the credit invested. Upon completion of target/goals, a user may also share their ‘Run’ with a friend. Following the Agile Methodology for Development, during the planned Sprints, sanity testing for each scenario will be conducted to realize the modifications in the due process. We make use of Perfecto for prototype development which is a cloud-based platform to achieve continuous and integrative testing of the application on a variety of platform and environment [14].

Many of the current applications are devoid of the integrative system with user based data and run on sensors which make it an expensive affair. Post which, applications need to be in sync with devices to provide an analysis. In our application, an integrative methodology combining Weighted Majority Algorithm along with Visual effects standout for an easy to use, and friendly application. HOOPS 3D Graphic system is a tool that will benefit developing the application by providing rich 3D graphic capabilities across different platforms and Large Model Visualization for real-time viewing of large datasets [15].

Model View Controller (MVC) is a framework for developing mobile applications. The model represents the application core, View displays the data and Controller handles the user input. The Model controls the flow of the application and it comprises the overall architecture of the application. The controller controls all the input from different users. Fragments are the part of an activity's user interface. It contributes its own layout to the activity. The user can create a callback method to provide fragment layout. Whenever the fragment layout is drawn, the android system calls the method. The library opencv is imported to strongly focus on developing real-time applications.

Thus, we are devoted to our research made for developing an application that is in best interest of people and their health, also by bringing innovation and development in the mobile medical health industry. Contributing to the same is done by researching on various technologies, articles, and resources made available on internet. A lot of knowledge and lessons have been gained through this project and has been paid in terms of developing ideas to bring a larger picture to practice in future.

1.3 Current State of the Art

Due to the large availability of the users as discussed above, the number of applications available in the android market is increasing. Developer of the application introduce the functionalities based on the target audience. Each of these applications tries to provide maximum functionalities with a user friendly interface. the application should also be lightweight and compatible with different android versions available. Storing data on the cloud provides scalability and reliability to the application. Introduction of machine learning has increased the functionalities of the application. Machine learning will improve the features of the application by analyzing the data about the physical attributes of the person. These values will help in training of the model and increasing the reliability and accuracy of the application. By learning the dataset values the person can be suggested with the options by which he can achieve fitness goals with minimal applied effort. Design model could provide choices of physical exercises and dietary supplements to the user. These alternatives will encourage the user to achieve his goal. Research and development is being carried out to achieve the above functionalities.

“Health care”, “monitoring vitals”, and “exciting visuals” are three aspects covered by Total Health adding an advantage to a User friendly Application Interface. Obesity, in concern is glaring amongst 39.8% of American population and affecting about 90 million people [10]. With the rise in popularity in mobile medical industry and the easy access to low-cost of Android device, has given us this opportunity to develop an application that will be easily accessible to a wide variety of people of different age groups and health to make use of the application to endure a better lifestyle.

Android application development can be done with Android Studio on which different languages are supported. Database management has moved from device storage to cloud environment. Servers have high computation engines where complex models can run thereby decreasing the computation on user-end machines.

Chapter 2. Project Architecture

2.1 Introduction

We have developed a smartphone Android Application and hence it requires Android operating system. The Application's architecture has the following essential modules: Client-server communication module, Database Module, Authentication and Validation, and Machine Learning module that uses Weighted Majority Algorithm. We shall be using the Model View Controller (MVC) design pattern to develop our application that involves three significant aspects: Controller, Model, and View.

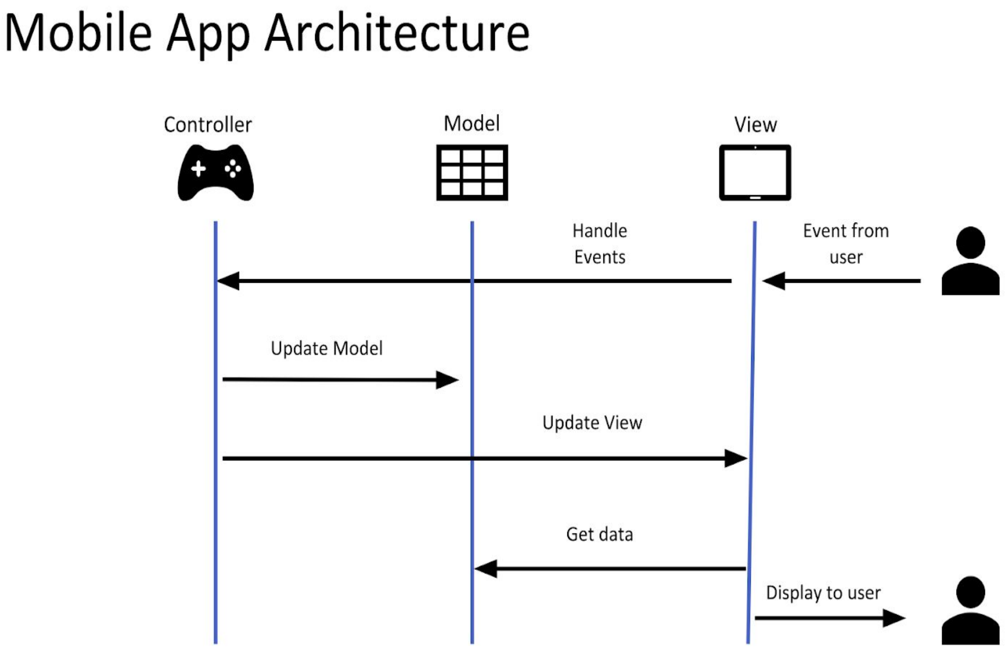


Figure 2. Architecture of the android mobile application.

2.2 Architecture Subsystems

The flow of data begins once the user opens the application for sign up and logs into the system. This involves various UI Fragments and database components that store and authorize user credentials. The different features of the various modules of the system are described below :

2.2.1 User Profile: Create the user profile by allowing the user to update all the personal details and body dimensions. It also includes bank account details to allow the user to stake money for the goals.

2.2.2 Set Goals: The user shall see some pre-defined fitness goals or create a new one that gets added to the database and can be used later as well. The duration of the goal i.e, the start and end of the goal shall be detailed by the user who sets the goal.

2.2.3 Nutritional Diary: Keep track of diet nutritional value and aggregate information on a daily, weekly and monthly basis. The following image explains the detailed implementation:

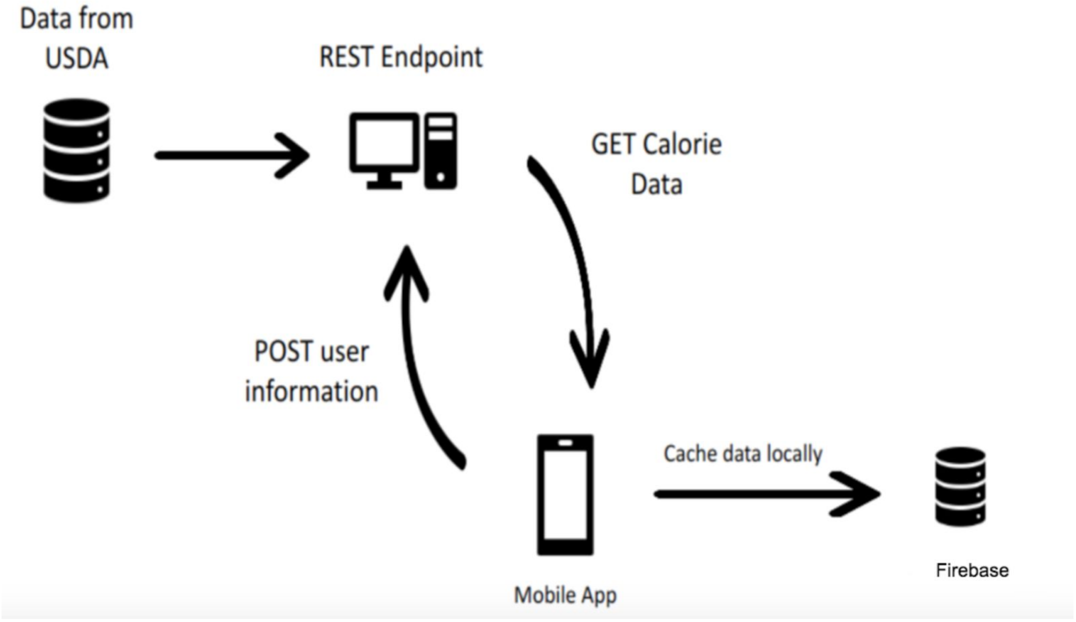


Figure 3. Detailed implementation of nutritional diary.

2.2.4 Monitor Activity: This ensures the user keeps updating the calories intake to maintain the user's personal health diary. Also, this requires the user to update the workout history and update the status of the goals that are set.

2.2.5 Creating Groups by sharing contacts: The set goals are shared amongst the friends to allow them to vote for the credibility of completion of it and hence the money is split accordingly or re-claimed by the user who sets or completes the goal.

2.2.6 Notification pop-ups: Based on the users involved in the goal, reminders are sent as notification. It involves the database access to allow timely pop-ups.

2.2.7 Predict BMI: Using the input data like height, weight and gender, BMI shall be predicted. Using a part of dataset as test data, we shall train the model using machine learning algorithm and then perform predictive analysis to predict BMI.

2.2.8 Predict WHR: The calculation of WHR suggests the waist-to-hip size ratio which is another way to realize the potential risk of user's health.

2.2.9 Analytical dashboard: From all the input data, we shall create an analytical dashboard that allows the user to graphically analyze the changes over time and help the user take accurate measures.

The UI for the application is divided into various Fragments(over activity) to improve performance. Speeding up the DB creation by using SQLite Statement and check the Database schema using an emulator. We shall be using Navigation bar and Expandable list view for effective user experience.

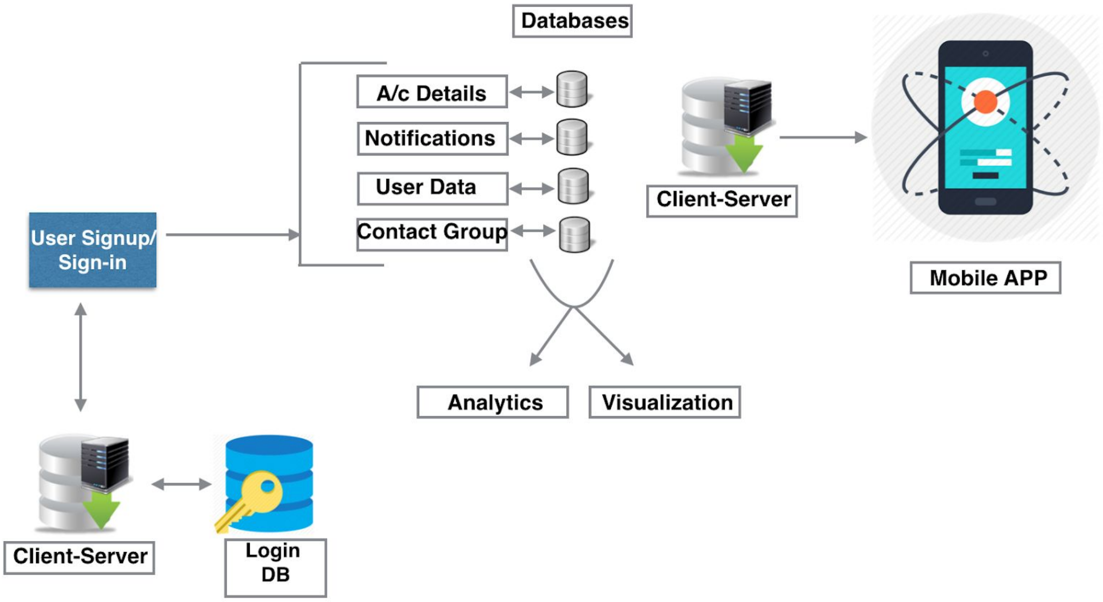


Figure 4. High-level architecture of the Application.

Chapter 3. Technology Descriptions

This section covers the different technologies that we have included in developing the application Total Health. The front-end is developed using XML language for recognizing the elements and their attributes. The realtime firebase database is used as a hosted database. Data is basically stored as JSON format and is synchronized in realtime to everytime a client tries connection to server. The cross platform applications developed are as such that all the clients share one real-time database and are capable of receiving all data updates all at once. We can use Google API to basically call the functionality of login with Google or Facebook, and login authentication are verified at Firebase. We use Google API in plenty to call responses and videos in publication to be viewed in the user profile, for motivation purpose. The Speech to Text conversion functionality is implemented using Google API. Apart from database and application deployment, Machine Learning Algorithm i.e. Weighted Majority Algorithm makes use of entities that are user defined and on Google Cloud platform. Below are the list of technologies used with a brief description of their implementation and usage.

3.1 Client Technologies

3.1.1 Android Studio: The use of Android Studio for the purpose of software development kit, combining the code and making use of Emulators for different Android Versions and deploying in an environment similar to debugging on a real-time device.

3.1.2 XML: This describes the kind of data being displayed. We work on Android Studio in XML language for the front-end basis. This kind of a file can be processed as data or by a program to store with similar data on another machine.

3.1.3 Google API: We use Google API to consolidate the working functionality/features like Login credentials for the user to be able to login with their Facebook credentials or to be able to share YouTube videos in their account settings.

3.1.4 NDB API: The API fetches data from USDA in REST to access USDA Food Composition Database. This kind of API reverts back the reports fetched for Food and Nutrient. Every list of food, will comprise of relevant nutrient value.

3.2 Machine Learning Technologies

3.2.1 Python: Python libraries such as NumPy and Pandas also with memory management for Long Short-Term Memory(LSTM). To implement the speech to text conversion, we use NLTK to perform Machine Learning algorithms on a specific set of data that is an input.

3.2.2 Amazon Comprehend: This function is basically used to fetch data from speech and search the database with the data as Key to provide an insight and relationship.

3.2.3 Libraries: The libraries used are Numpy and Pandas that are used to pre-compile the code, and observe the complex functionality of calculation and aggregation.

3.3 Server Technologies

3.3.1 Firebase: This is a cloud service to synchronize between the client and server data. They are scalable and incorporate well within existing services. Firebase stores all of its data in MongoDB and offers built-in capability for applications to support scalability and provide each piece of data stored as a JSON document.

3.3.2 MongoDB: This type of database system is hosted on the cloud to ensure reliable database connection for storing the objects in the form of collections.

3.3.3 JSON: Javascript Object Notation will revert back the object and their values pertaining to the element identified along with their attributes.

3.3.4 Hoops 3D: To encourage visualization and have their effects in 2D, we use this technology for us to design a character using unified API. This tool uses platform independent input architecture and interfaces with Java, a technology used at Backend.

3.3.5 Java: The back-end of the application is implemented with Java programming, defining packages, libraries, and with functions defined to calculate BMI and call integration with API accessing the Firebase database.

3.4 Data-Tier Technologies

3.4.1 Firebase: This database is real time available offline, available to the client, and open for modifications. Referencing the JSON Data as follows:

users/user:1234/set_user_goal

The above creation of goal, will result in creation of a goal as an index that is set with values typically an numerical value.

3.4.2 In-built Sensor: This sensor is linked to the application and the user device, such as his phone to be able to track the step count that will variably add the step counts, increment the values automatically.

3.4.3 ProgressView: The UIProgressView class provides us with properties to manage the progress bar and various styles for getting/setting values to pin onto a particular task.

3.4.4 AppIntro: This is a library introduced to provide an introductory page with a slide through options. The activities declared are done through the following syntax:

```
<activity android:name="com.example.example.intro"
          android:label="@string/app_intro" />
```

Chapter 4. Project Design

Add additional chapters if necessary to keep chapters at a reasonable length. This chapter should describe the important design elements of your project. Describe elements that are key to project and that are innovative. The topics below are for a typical MS Software Engineering project. Adjust the topics in this chapter to meet the needs of your project.

4.1 Design Flow

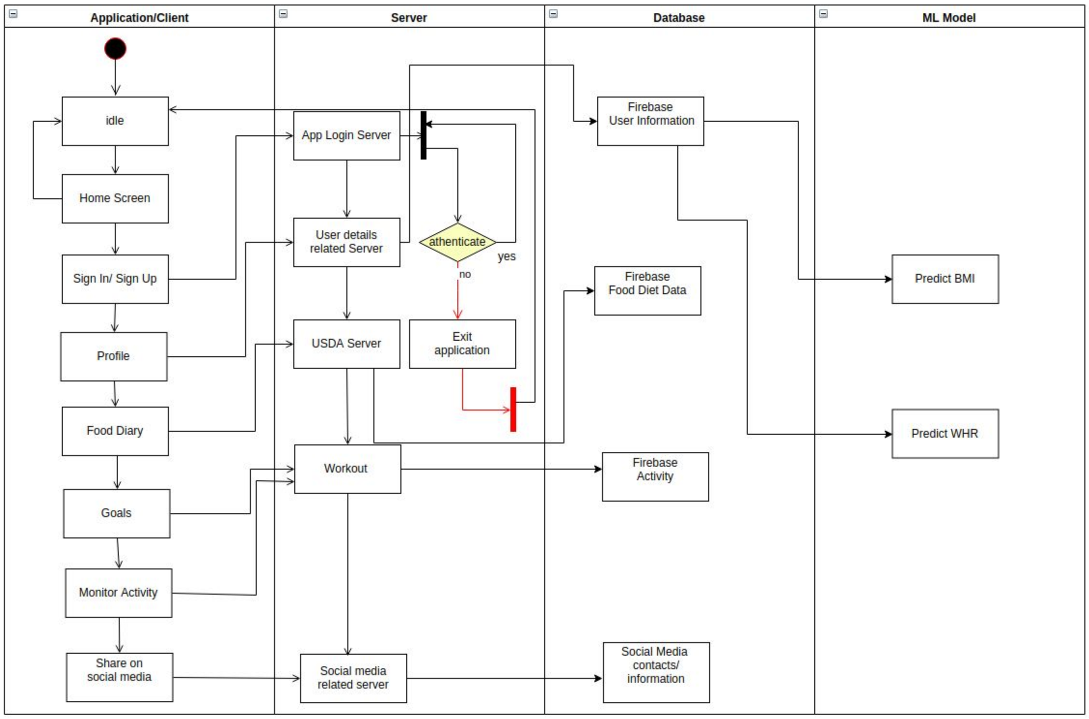


Figure 5. Project Flow Diagram.

The project flow diagram comprises of Application interacting with server, which provides elements based interaction with Login and authentication, USDA Server records, workout workflow all of which interacting with databases on Firebase along with social media interactions published.

At application level, we have user defined parameters and functions like creating an account, display of homepage, setting the profile with Goal Name and Task Count values, Monitor activity by interactive routine workout and calorie count, drawing from USDA calorific instances.

The local server is in charge of drawing application credentials and to retrieve the profile in the previous state. Enabling the user to be able to set goals/target, calculate the calorie specifics and return the value of burnout/added calories.

The database specifics relate extracting user information on a high level, and authenticating the login with self credentials or credentials obtained from Google APIs. Also the USDA API reflect the calorific values of the dietary items consumed by the user and adds to their profile, following which Firebase Activity is monitored by the routine of the user based on Workout/calorie count burn obtained by inbuilt sensors.

The Machine Learning Model uses supervised learning model that will fragment the modules according to decision tree model. It realizes the data, extracting the fragments, and return a value for predictive analysis on the basis of the algorithm being trained. This would in turn return an BMI and WHR Ratio, helping the user realize the amount of weight that they should gain/lose.

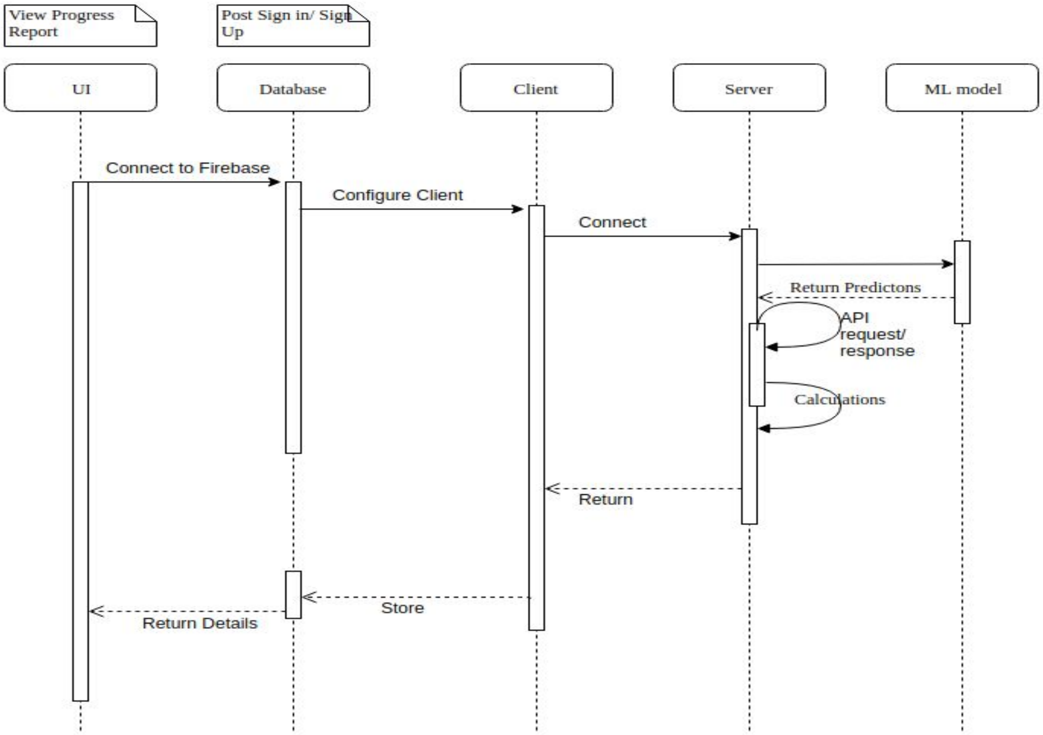


Figure 6. Predict BMI and WHR.

The BMI and WHR prediction are done using ML Model. The calculations of actual BMI

and WHR are done on the server, where functions are defined with a return in the form of predictive analysis from the ML algorithm. The API fetch data from server and interact with a set of continual calculations on the whole returned to the client.

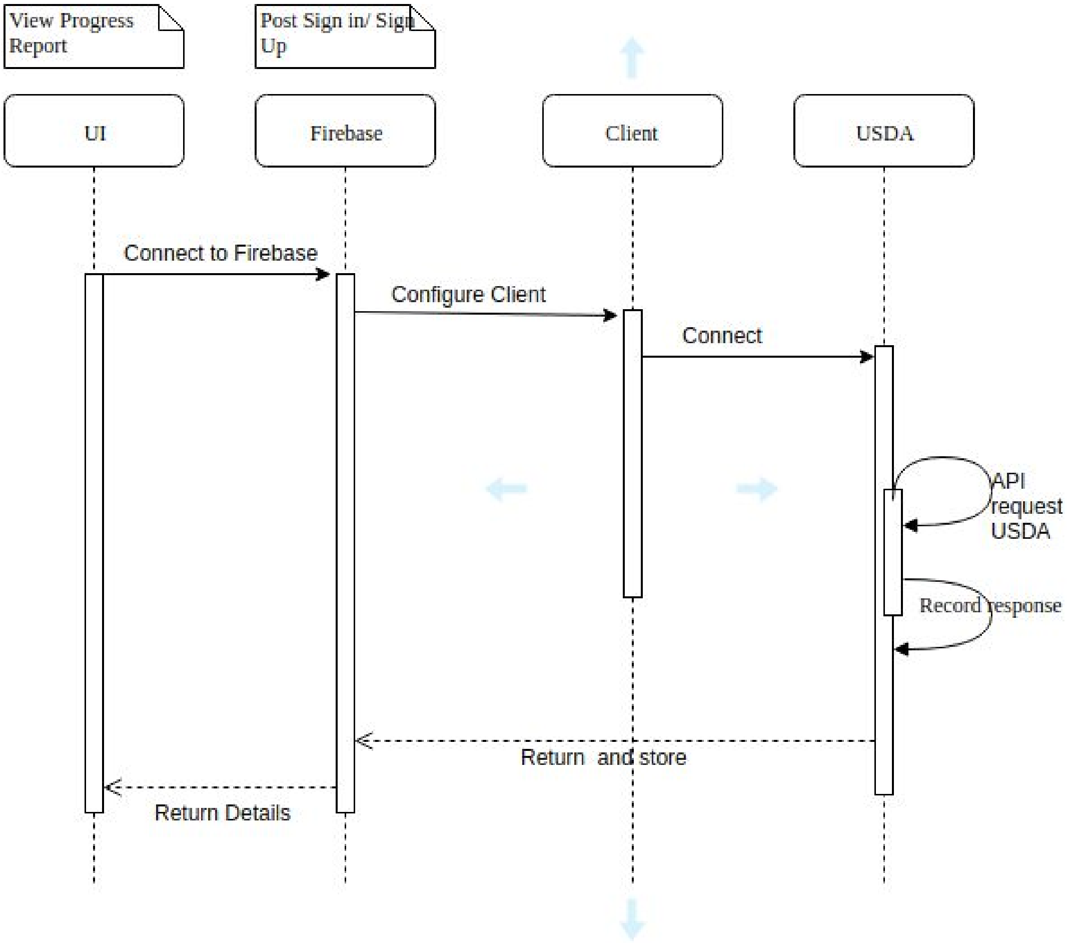


Figure 7. Daily Food Activity

The daily food activity links the UI to firebase interaction and connect to client server. This basically fetches USDA calorific values using API post connecting from USDA which will return the response of the values of the key element or Items. All the values are stored in Firebase of the result fetched and proceed to add to the user values and results. This will return the object and values, and find the corresponding calorie count and goals and the state is returned upon login or access at regular intervals by means of creating a session.

4.2 Database Entity Diagram

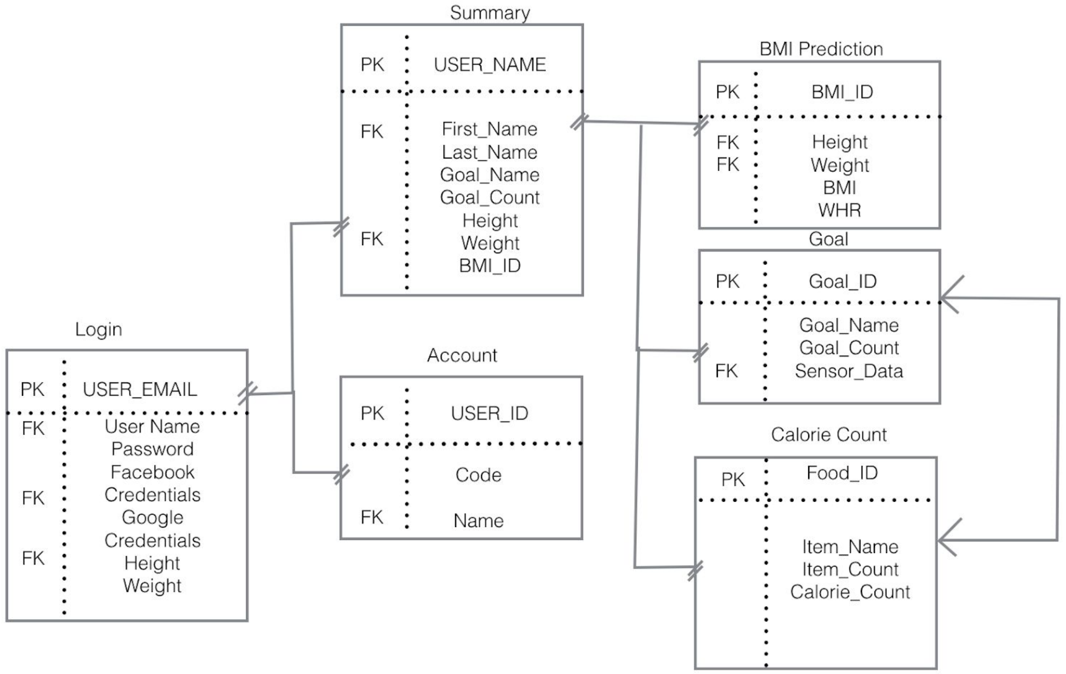


Figure 8. Database entity diagram .

4.3 UI Mockups

Following are some of the use cases for our application. The use cases include user registration, record daily diet, measure daily fitness and graphical view of data.

Use Case Element	Description
Use Case Number	1
Application Use Case	User registration
Use Case Description	New user shall be able to sign up into the application and provide correct details in the fields.
Primary Actor	User
Preconditions	The Application has been installed by the user.
Basic Flow	● The user opens the application and signs up using email id or using facebook profile or using the phone number. ● These are authenticated and the user logs in. ● The user records its details of age, gender, body measurements, profile picture, etc. which is then saved in the database.

Table 1. Registering new user in the application.

Use Case Element	Description
Use Case Number	2
Application Use Case	Daily food diary.
Use Case Description	The user shall be able to record daily food and create its own personal food diary.
Primary Actor	User
Preconditions	The user has logged in to the application and has a profile that has been created with accurate data.
Basic Flow	● The user enters the daily food intake to record the diet. ● Each entry is matched with the USDA database to fetch the calories. ● These details create the daily nutrition statistics for the user. ● The user can monitor the calorie consumption for each day.

Table 2. Recording daily diet in the application.

Use Case Element	Description
Use Case Number	3
Application Use Case	Record daily workout details.
Use Case Description	The user shall be able to set workout goals and enter daily workout data against those goals into the app to monitor fitness.
Primary Actor	User
Preconditions	The user has logged in to the application and has a profile that has been created with accurate data.
Basic Flow	● The user can set goals for workout each day by entering the details of exercise in the application. ● On completion of the goal set, they can be marked as complete. ● The data is verified against the smart watch that is linked to the application.

Table 3. Measure daily fitness details of the user.

Use Case Element	Description
Use Case Number	4
Application Use Case	View graphical progress of the workout data.
Use Case Description	The user shall be able the view the numeric data entered in graphical form with analysis results by week, bi-week or by year.
Primary Actor	User
Preconditions	The user has logged in to the application and has a profile that has been created with accurate data.
Basic Flow	● The user shall be able to view the statistical report based on the data entered by the user. ● The user shall be able to monitor weekly, monthly and yearly consolidated data in form of graphs.

Table 4. View graphical statistics of the user.

4.4 Data And Methodology

Following are the key metrics that are essential to the application performance. The android application shall be tested for the following and improvised to get better performance to users :

- **Total Availability:** Total availability reveals the ability of our end users to access to our service or mobile app through the day and from day to day over a longer period.
- **Time to First Byte (TTFB):** This measures the responsiveness of our mobile app. It indicates how fast our app is able to launch and get the first responses from the back end servers.
- **Reaction Time:** It's the measurement that reflects the speed at which our mobile application reacts to user inputs. For example, "how fast your application starts showing search results," "How fast the business transactions are completed," etc.
- **Time to Load:** This is a metric that implies the time elapse between the moment users launch your app and the moment users can start interacting with our app. It tells how fast the application becomes responsive to user queries.

Since we are also using machine learning principles for predictive analysis of the data, following performance metrics will also be taken into consideration to measure application performance:

- **Classification Accuracy:** It is the ratio of number of correct predictions to the total number of input samples.It works well only if there are equal number of samples belonging to each class.

  Accuracy = Number of Correct Predictions / Total Number of Predictions made

- **Mean Absolute Error:** Mean Absolute Error is the average of the difference between the Original Values and the Predicted Values. It gives us the measure of how far the predictions were from the actual output. However, they don't give us any idea of the direction of the error i.e. whether we are under predicting the data or over predicting the data. Mathematically, it is represented as :

$$\text{Mean Absolute Error} = (1/N)\ \Sigma\ \text{abs}\ (y_j - \text{vector}(y_j))$$

- **F1 Score:** F1 Score shall be used to measure a test's accuracy. It is the Harmonic Mean between precision and recall. The range for F1 Score is [0, 1]. It tells us how precise our classifier is (how many instances it classifies correctly), as well as how robust it is (it does not miss a significant number of instances).High precision but lower recall, gives you an extremely accurate, but it then misses a large number of instances that are difficult to classify. The greater the F1 Score, the better is the performance of our model. Mathematically, it can be expressed as :

F1 = 2 * ((recall * precision) / (recall + precision))

- Precision : It is the number of correct positive results divided by the number of positive results predicted by the classifier.
- Recall : It is the number of correct positive results divided by the number of all relevant samples (all samples that should have been identified as positive).

F1 Score tries to find the balance between precision and recall.

4.4.1. Data Sets :

So far, we have collected about 1000 data sets lies in similarity to our application and can be used for analytical prediction. Sample table is as shown below:

Data Set No	**Gender**	**Height (cm)**	**Weight (kg)**	**Index**	**BMI**
1	Male	174	96	4	31.7
2	Male	189	87	2	24.4
3	Female	185	110	4	32.1

4	Female	195	104	3	27.4

4.4.2. Metrics Explanation:

Column 1: Gender - This column gives us a reference of the gender and the basis on which the data is accumulated.

Column 2: Height (cm) - This column defines the height of each user and with the unit centimeter.

Column 3: Weight (kg) - This column defines the weight of each user and with the unit in kilograms.

Column 4: Index - Index of the person is defined by the calculation of height and weight described by the user. The scale is as follows:

0 - Extremely Weak

1 - Weak

2 - Normal

3 - Overweight

4 - Obesity

5 - Extreme Obesity

Column 5: BMI - The Body Mass Index of a person is the simple division of height/weight wherein the values of range have definitions as follows:

Underweight = <18.5

Normal weight = 18.5–24.9

Overweight = 25–29.9

Obesity = BMI of 30 or greater

Note that BMI and Index have a dependency and we rate them on scales interdependent on each other.

Also extendable scope includes the waist to hip size ratio that would include various suggestions for an ideal measurements of a person. The WHR or the Waist-to-Hip ratio is a measure to inform the user about the risk for future health in support of correcting them immediately. The indicators are well supported in scientific fields and detect major health risks.

The Data Sets so far collected is 1000 on an approximate that will deal with the relevance of exposing a person to his health risk factors and help them decide to get to the ideal measurements they have to set to their goals.

Gender	**Waist Measurement (cm)**	**Hip Measurement (cm)**	**WHR Ratio**
Female	66	69	0.95
Male	74	80	0.92
Female	70	86	0.81

The risk factor and the other tabular factors of the above table are defined as below:

Female	**Male**	**Health Risk Factors**
0.80 or lesser	0.95 or lesser	Low Health Risk
0.81 - 0.84	0.96 - 1	Moderate Risk
0.85 or higher	1.0 or higher	High Risk

What are we predicting? And why?

In this application Total Health, we are essentially calculating two things : BMI (Body Mass Index) and WHR (Waist to Hip Ratio). Coming to answer what are the essential inputs required by the user are the following fields:

1. Gender
2. Height in cm
3. Weight in kg
4. Waist Measurements in cm
5. Hip Measurements in cm

In our application, we are calculating the BMI and WHR for essentially classify a user of a certain category of health risk. Based on the experiments and regression model, we classify and suggest the user to be able to realize the amount of weight to be lost in order to be in the recommended waist size for their gender.

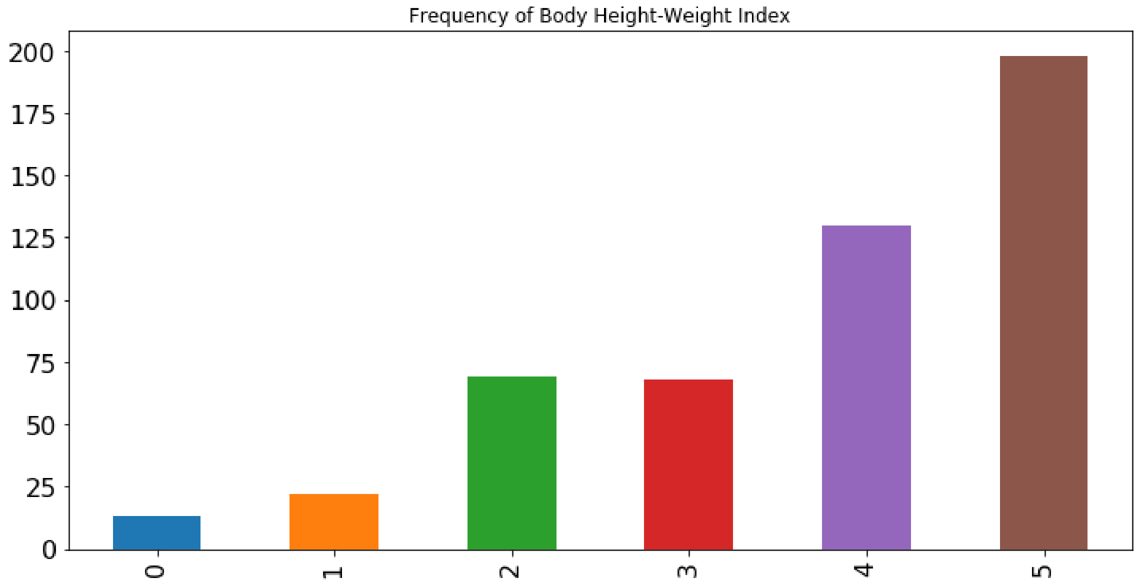
We will be able to set target goals and have a diet monitoring system that will be of suggestion based on the BMI of a user to fall into category of prone to lesser health risk. For instance, we have a female user with a BMI 33 and WHR 0.9. These are the parameters suggest high risk of health concerns. The application will suggest the following in return:

1. Suggestion and recommendation for BMI and WHR approximately falling within range 24 and 0.8 respectively. Note that this is implemented using Machine Learning Algorithm for recommendation and suggestions.
2. A workout target goal that suggests 1000 calories burn-out every day.
3. A diet plan that will suggest calorie intake of about 1500 calories a day.

This prediction will help the user realize the need to stay fit and visualize themselves in a more obvious and healthy way. For prevention being better than cure, we help the user in realizing what health risk they are at.

4.4.3. Analysing the Data and the Experiments on dataset :

Analysing the datasets using a bar plot and finding the frequency of body height-weight index :

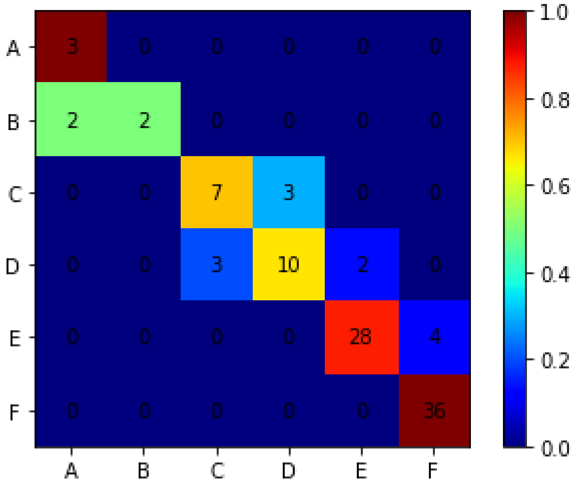


The reference index used in the above plot describes people who are 0 - Extremely Weak, 1 - Weak, 2 - Normal, 3 - Overweight, 4 - Obesity, 5 - Extreme Obesity

Preparing and splitting data as 80% training data and 20% testing data.

Now moving onto Classification models, we will try 4 basic methods of classification of the data :

- **K-NN Classification**

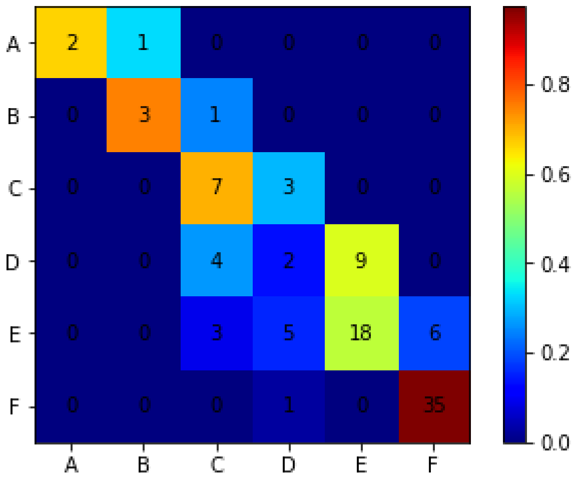


	A	B	C	D	E	F
A	3	0	0	0	0	0
B	2	2	0	0	0	0
C	0	0	7	3	0	0
D	0	0	3	10	2	0
E	0	0	0	0	28	4
F	0	0	0	0	0	36

K-NN gives **85%** right prediction on the test set .

The k-fold cross validation removes the data bias and gives an accuracy of 82.28% ± 4.27%

- **Naive Bayes**

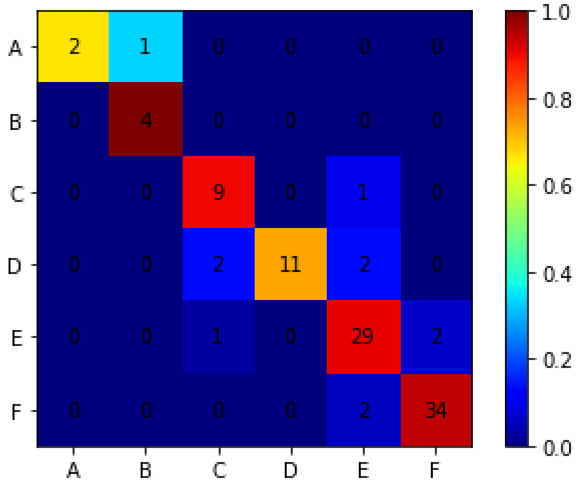


	A	B	C	D	E	F
A	2	1	0	0	0	0
B	0	3	1	0	0	0
C	0	0	7	3	0	0
D	0	0	4	2	9	0
E	0	0	3	5	18	6
F	0	0	0	1	0	35

Naive Bayes gives **67%** right prediction on the test set .

The k-fold cross validation gives an accuracy of 71.54% ± 6.05%

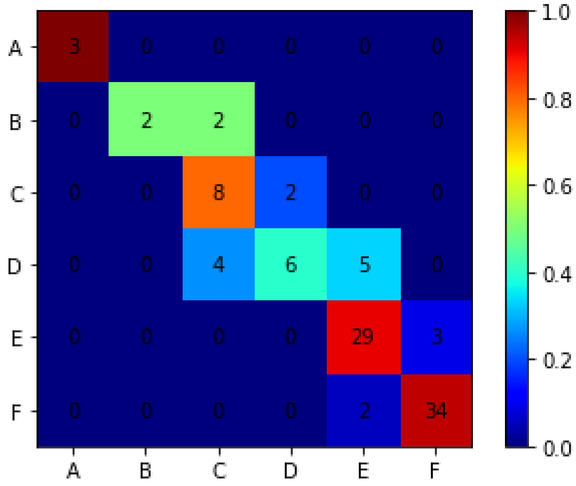
- **Random Forest**



Random Forest Classifier gives **89%** right prediction on the test set .

The k-fold cross validation gives an accuracy of 84.79% ± 4.25%

- **XGB Classifier**



XGB Classifier gives **82%** right prediction on the test set .

The k-fold cross validation gives an accuracy of 84.90% ± 4.75%

Comparison:

Classification Model	**% of correct predictions**	**Mean**	**Standard Deviation**
(1)K-NN Classification	85%	0.8228994082107048	0.04274651325985314
(2)Naive Bayes	67%	0.7154329627628728	0.06052155443975069
(3)Random Forest	**89%**	**0.8479215672668818**	**0.04255105534539179**
(4)XGB Classifier	82%	0.8490967902617454	0.047523062588980536

On comparing all the models performance from it's confusion matrix and through k-fold cross validation method we shall select the most suited model i.e, Random Forest, which has the highest accuracy of 89% for the prediction of the data.

Chapter 5. Project Implementation

The application has been designed with Android Studio 3.3.2. Android Studio 3.3.2 supports database management with Google Firebase. Firebase provides functionalities such as authentication, database management, cloud messaging and crash reporting. Authentication features can be supported by adding the sdk components to project level build.gradle file. Select the feature of authentication from the popup window and authenticate through same credentials that were used to create the firebase account. The application is connected to Firebase successfully and Java Classes are designed using the Page Object Model.

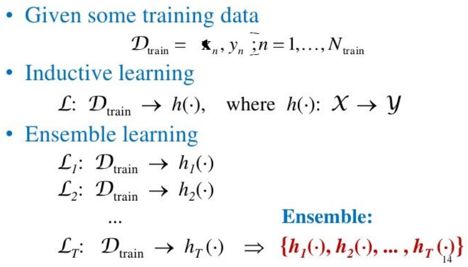
5.1 Vital Aspects of Implementation background of Total Health

5.1.1 UI and Intent: The design of the web pages are implementing by using interactive button, text link and progress bars throughout the application. The application navigates to another page by utilizing the intent activities to call/start a new page set-up.

5.1.2 BMI and Days to Goal: Another vital implementation is calculation of BMI and WHR. Post calculating BMI, the application provides an estimate to duration to reach the number of days to your Target Goal. We have implemented this by using Weighted Majority Algorithm.

5.2 Weighted Majority Algorithm

This Algorithm provides an insight to metadata learning algorithm to provide predictions from other basic algorithms put together that could be of the form of learnings or classifiers. This kind of algorithm also seeks input from human experts and basically assumes that there absolute zero knowledge of algorithms all put together in pool. This algorithm is also on the basis of considering that at least one of the algorithms would work well and the mistakes could be eliminated by the same.



- Given some training data

$$\mathcal{D}_{\text{train}} = x_n, y_n ; n = 1, \ldots, N_{\text{train}}$$

- Inductive learning

$$\mathcal{L}\colon \mathcal{D}_{\text{train}} \to h(\cdot), \quad \text{where } h(\cdot)\colon \mathcal{X} \to \mathcal{Y}$$

- Ensemble learning

$$\mathcal{L}_1\colon \mathcal{D}_{\text{train}} \to h_1(\cdot)$$
$$\mathcal{L}_2\colon \mathcal{D}_{\text{train}} \to h_2(\cdot)$$
$$\ldots$$
$$\mathcal{L}_T\colon \mathcal{D}_{\text{train}} \to h_T(\cdot) \Rightarrow \{h_1(\cdot), h_2(\cdot), \ldots, h_T(\cdot)\}$$

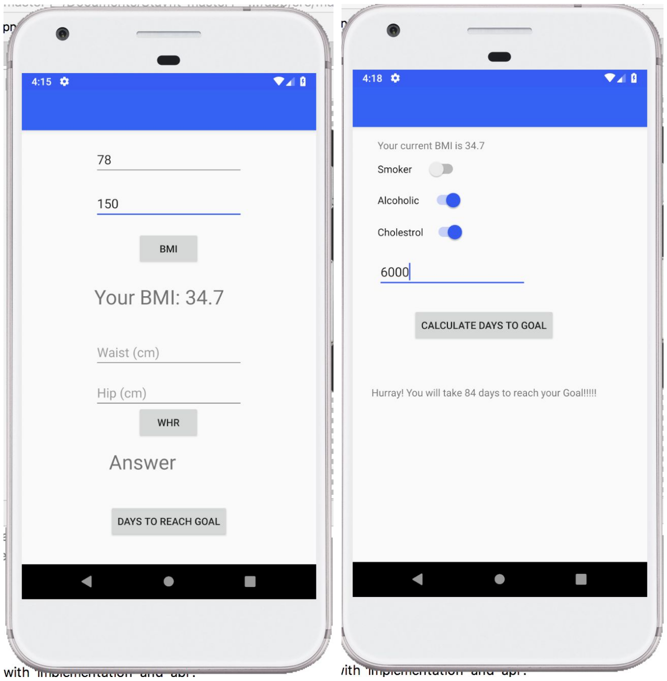
Ensemble:

14

Figure 9. Weighted Majority Algorithm

The above figure gives us insight of the Weighted Majority Algorithm that suggests learnings based on every algorithm present in the pool. The mistakes allowed by any algorithm $\mathbf{x}_i$ is given by **m**(maximum possible mistakes an algorithm can make), in a pool

A comprising of all algorithms. The upper bound for mistakes allowed in every algorithm is as follows: **O(log|A| + m)**



4:15

78

150

Your BMI: 34.7

Waist (cm)

Hip (cm)

Answer

4:18

Your current BMI is 34.7

Smoker ○

Alcoholic ●

Cholestrol ●

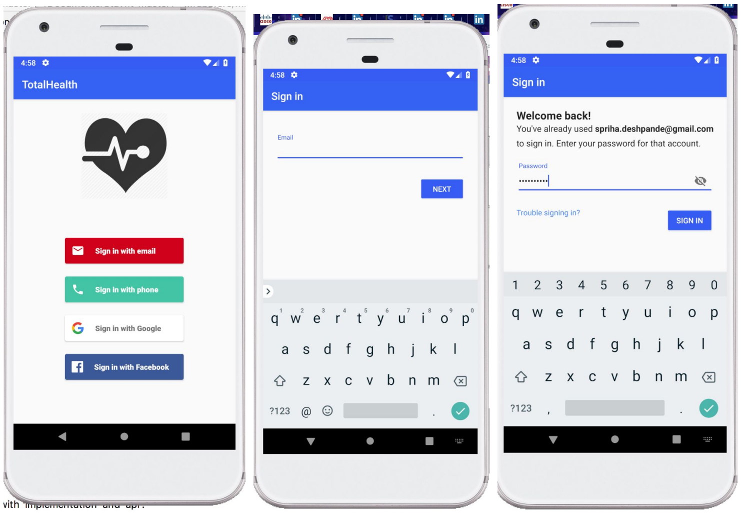
6000

Hurray! You will take 84 days to reach your Goal!!!!!

Figure 10. Screen Display of providing Estimation to Reach Goal

The above screen shows the calculation of BMI and WHR by the user input. The continuity of the next screen is upon clicking on the button "Days to Reach Goal". This screen takes input from the user habitual aspects, and maps to the target goal. Using ML Algorithm, we further are able to predict the number of days to reach the goal by the User.

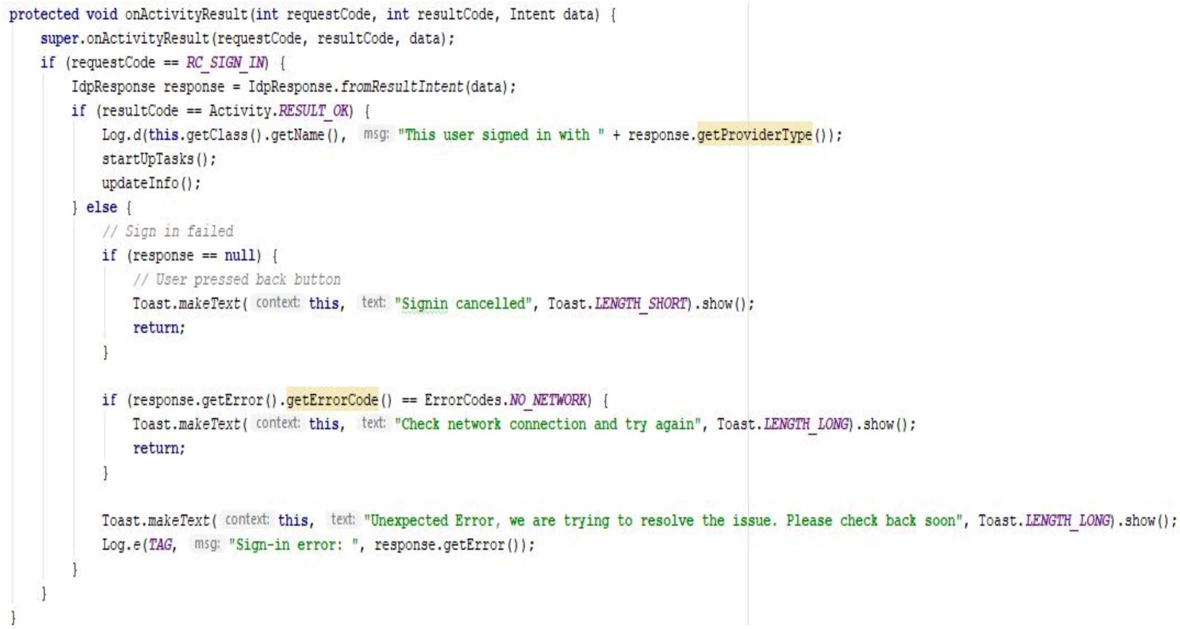
5.3 Login and Implementation



Welcome back!
You've already used **spriha.deshpande@gmail.com** to sign in. Enter your password for that account.

Figure 11. Login screen of the application.

Different features for user authentication is provided. A user can be authenticated by signing with google account or else can use a different email for signing in. These API function calls are made to google api which verifies the information provided by the user. Below is the code snippet for the login activity.



```java
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    if (requestCode == RC_SIGN_IN) {
        IdpResponse response = IdpResponse.fromResultIntent(data);
        if (resultCode == Activity.RESULT_OK) {
            Log.d(this.getClass().getName(),  msg: "This user signed in with " + response.getProviderType());
            startUpTasks();
            updateInfo();
        } else {
            // Sign in failed
            if (response == null) {
                // User pressed back button
                Toast.makeText( context: this,  text: "Signin cancelled", Toast.LENGTH_SHORT).show();
                return;
            }

            if (response.getError().getErrorCode() == ErrorCodes.NO_NETWORK) {
                Toast.makeText( context: this,  text: "Check network connection and try again", Toast.LENGTH_LONG).show();
                return;
            }

            Toast.makeText( context: this,  text: "Unexpected Error, we are trying to resolve the issue. Please check back soon", Toast.LENGTH_LONG).show();
            Log.e(TAG,  msg: "Sign-in error: ", response.getError());
        }
    }
}
```

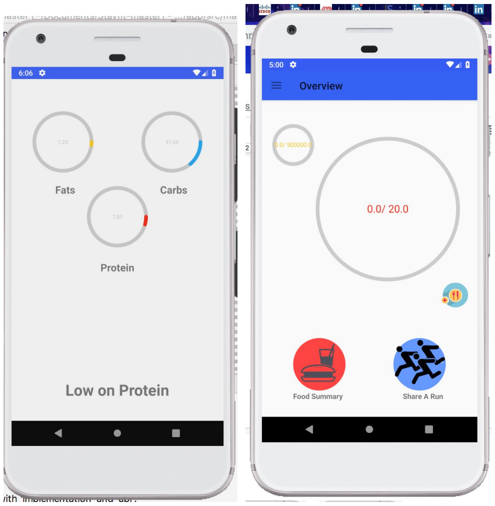
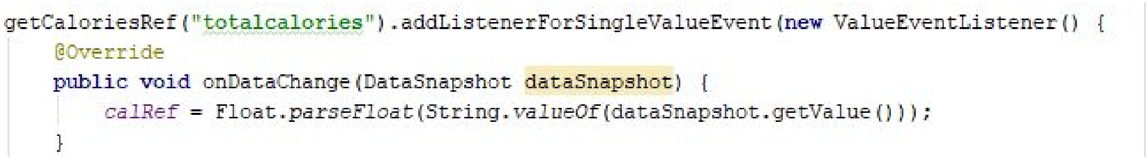
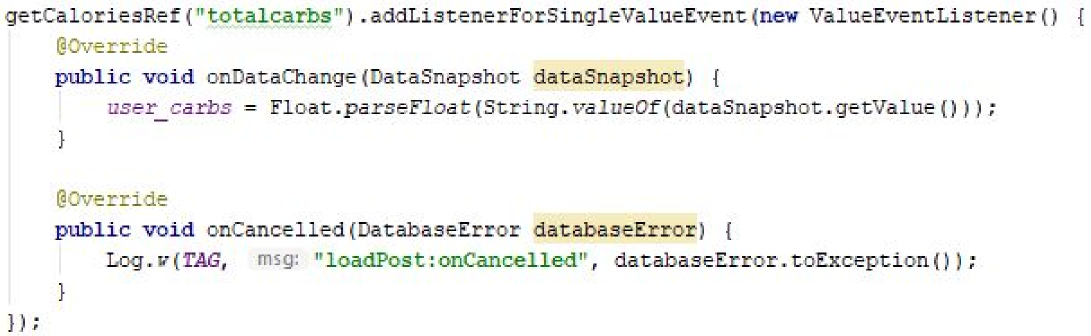


Figure 12. Calorie intake and workout details.

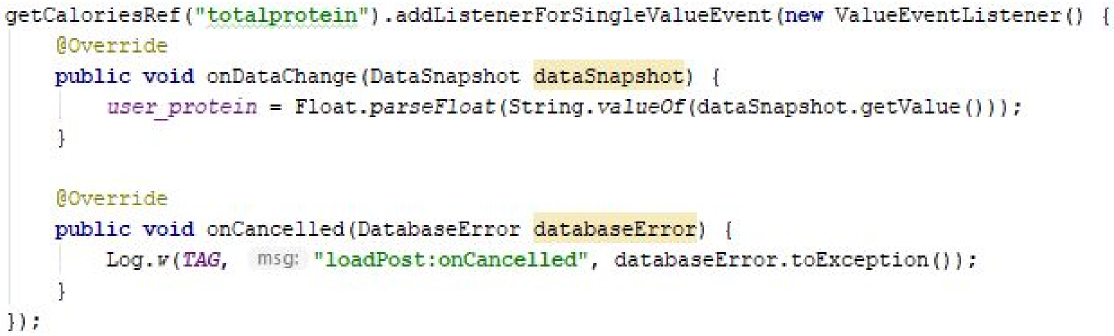
The above figure gives insights about the food intake of a person by calculating the amount of fats, carbohydrates and protein in the food. Codes and the API description are listed below



```
getCaloriesRef("totalcalories").addListenerForSingleValueEvent(new ValueEventListener() {
    @Override
    public void onDataChange(DataSnapshot dataSnapshot) {
        calRef = Float.parseFloat(String.valueOf(dataSnapshot.getValue()));
    }
```

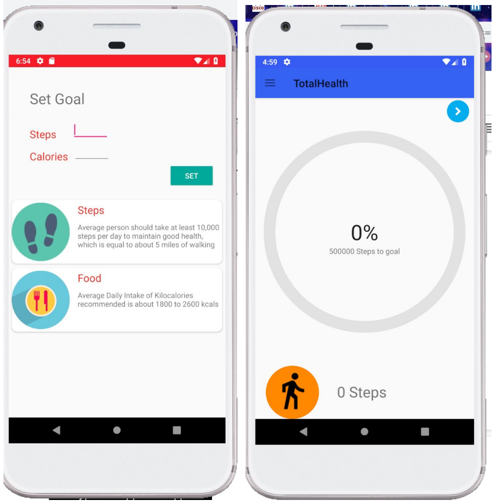


```
getCaloriesRef("totalcarbs").addListenerForSingleValueEvent(new ValueEventListener() {
    @Override
    public void onDataChange(DataSnapshot dataSnapshot) {
        user_carbs = Float.parseFloat(String.valueOf(dataSnapshot.getValue()));
    }

    @Override
    public void onCancelled(DatabaseError databaseError) {
        Log.w(TAG,  msg: "loadPost:onCancelled", databaseError.toException());
    }
});
```



```
getCaloriesRef("totalprotein").addListenerForSingleValueEvent(new ValueEventListener() {
    @Override
    public void onDataChange(DataSnapshot dataSnapshot) {
        user_protein = Float.parseFloat(String.valueOf(dataSnapshot.getValue()));
    }

    @Override
    public void onCancelled(DatabaseError databaseError) {
        Log.w(TAG,  msg: "loadPost:onCancelled", databaseError.toException());
    }
});
```

5.4 Target Goal and Database

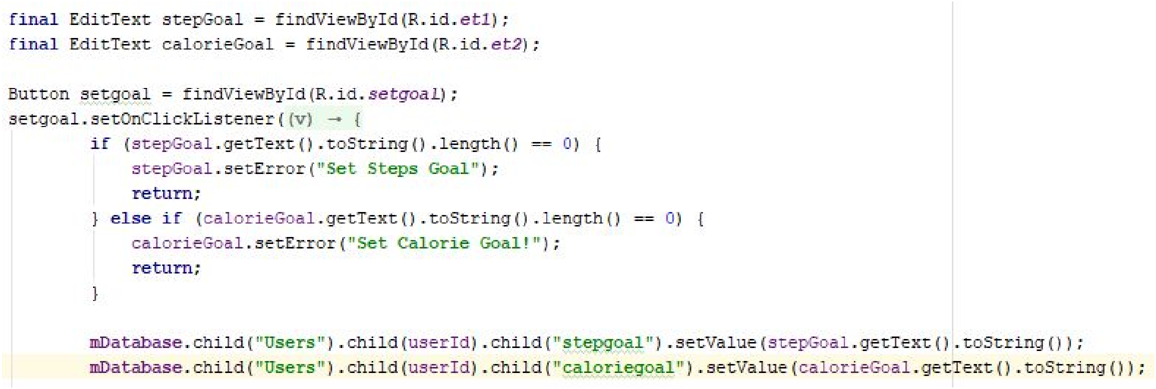


Average person should take at least 10,000 steps per day to maintain good health, which is equal to about 5 miles of walking

Average Daily Intake of Kilocalories recommended is about 1800 to 2600 kcals

Figure 13. Goal setting and monitoring.

The user can set goals personally and the calorie intake he wishes to intake for the completion of his goals. Application provides modularity to the user to change the goal according to his contentment.



```
final EditText stepGoal = findViewById(R.id.et1);
final EditText calorieGoal = findViewById(R.id.et2);

Button setgoal = findViewById(R.id.setgoal);
setgoal.setOnClickListener((v) → {
        if (stepGoal.getText().toString().length() == 0) {
            stepGoal.setError("Set Steps Goal");
            return;
        } else if (calorieGoal.getText().toString().length() == 0) {
            calorieGoal.setError("Set Calorie Goal!");
            return;
        }

        mDatabase.child("Users").child(userId).child("stepgoal").setValue(stepGoal.getText().toString());
        mDatabase.child("Users").child(userId).child("caloriegoal").setValue(calorieGoal.getText().toString());
```

The diagram below shows the steps involved in the connection of the database server which in this case is Google Firebase and the API function for its implementation.

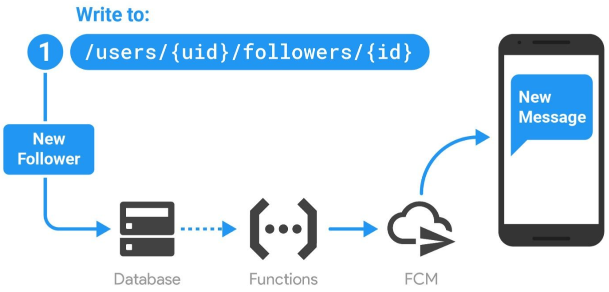
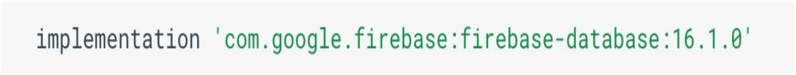


Figure 14. Connection to database server.

The API sends a request for fetching the user-data from the database. This API call includes the user-id and the id of the function call to verify the authenticity of the user. Once the request is received by the server a response message is forwarded to the user device consisting of information related to the user such as calorie intake and the goals set by the user.



```
implementation 'com.google.firebase:firebase-database:16.1.0'
```

The firebase API 16.1.0 is used for authenticating the application to the database server. It also provides real-time database updation for our application.

Chapter 6. Testing and Verification

Testing is an important phase of application development. The application under testing is made robust for security reasons, feasibility, quality, customer satisfaction. Typically, we follow the SDLC routine, and distributed over collective model of Agile in background. Advantage of Agile Model for testing and verification is that at every stage, we focus on working model, continually test through modules of structure of code and software development. Agile is useful in a phase, which handles continuous requirement changes and ad-hoc requests.

6.1 Process

The methodology will provide us a deep dive into the requirement analysis and for a developer who will be able to reproduce the work or extend the scope by the same. This section will provide information on the stages of design and analysis which is as follows:

6.1.1 Stage 1: Requirement Gathering - The research and documentation of Workbook part-1 and part-2 have helped by all means for functional requirement and observations. Here we have primarily decided to go with four basic functionalities i.e.

1. A User homepage for displaying information on goal, current activities, and profiling.

2. A UI based interactive tabs which involves different input grounds for calorie values and other body parameters.
3. A Machine Learning based regression model for predictive analysis.
4. A 3D Visualization of self-portrayed character and of variable size.

6.1.2 Stage 2: Preparing a UI - We believe the best of application is the user satisfaction and feedback. Preparing a UI using android framework is the most essential ones. We will implement the tags and input fields with the best possible combinations on every tab which will make it easier for the users to input their recordings. For instance, a dropdown, simple button, or a checkbox in preference to the traditional text field. The 3D visualization is also an important aspect as discussed in earlier section. Going through this implementation with different frameworks and tools in an environment that will be supported by all platform, will make the application a user-friendly one.

6.1.3 Stage 3: Coding, Testing and Modifications - The development of application is planned over Agile and its sprints that will cover implementing the four different functionalities in each one. Application is planned on developing with Java on Android SDK. The changes and the following regression test will be used with different test cases based on every Sprint.

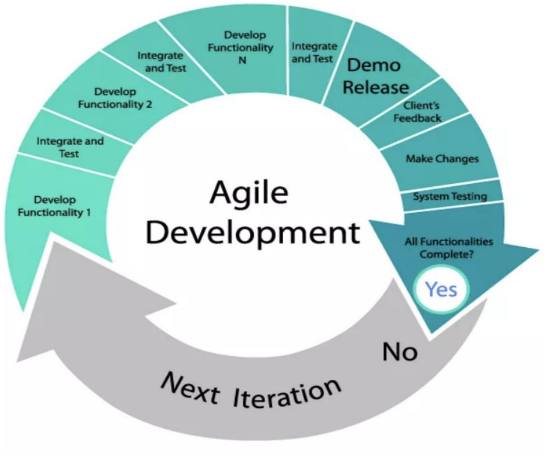


Figure 15. Software Development Life Cycle.

6.1.4 Stage 4: Launching - The application will then be launched that befit every platform of Android platforms. Also, that the application will support the feature of sharing the results or outcome of the challenges in the application. These challenges will provide user with motivation that is a factor in attracting their friends or other users to install the application via social media.

6.2 Results

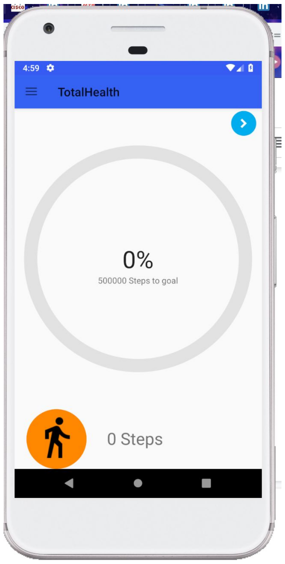


Figure 16. Achieve a Goal.

The goal is set with a particular number of Steps and Name of the goal. This page will display the goals being set and the progress bar is set in a circular shape.

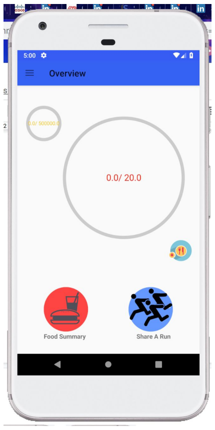


Figure 17. Food and Calories.

This page displays the total number of calories that a user has consumed according to USDA standards. This function is set automatically to add the calories to the diet and will have a burn out upon working out of the goal.

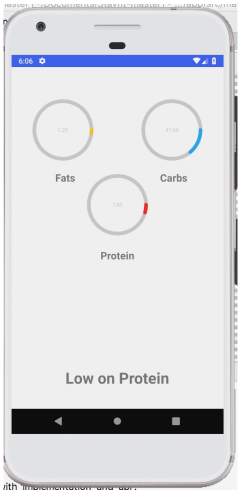


Figure 18. Nutritional details.

The food summary option helps the user realize their calorie intake and burnout on the basis of carbohydrates, protein and fat. This input is taken from user based on USDA calorific value on subsequent quantities that are user defined and customizable at any point of time during the day.

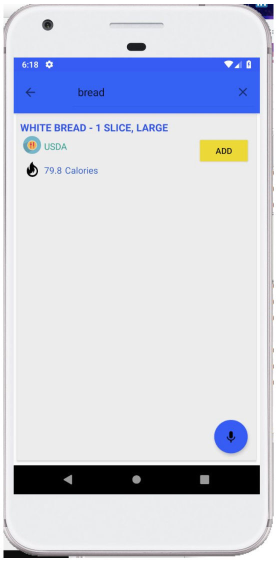
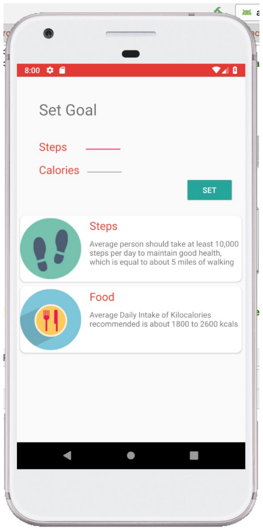


Figure 19. USDA Calorific Value

The following instance is an example of USDA calorific value along with the item name. The add button enable the user to add the item to their consumption automatically by using the Trigger function to have a total take on the food intake.



Steps

Average person should take at least 10,000 steps per day to maintain good health, which is equal to about 5 miles of walking

Food

Average Daily Intake of Kilocalories recommended is about 1800 to 2600 kcals

Figure 20. Setting a goal.

The screenshot above corresponds to the Goal set up along with requirements that are customizable. At the backend, these are free text fields so that it is user defined and can be set according to the user's convenience. The goal of this page is to have a personalized set up for goals and calories according to user's personal use.

6.3 Verification

The verification is along the requirement to be dealt with. We have byfar covered the requirements as follows:

6.3.1 Login Set Up: Using Single Sign On policy, with API's from Google for the benefit of login to the application.

6.3.2 Profile and Dashboard: We have successfully created a profile for self with target goals and burnout charts that will set the user on par with their ongoing workout routine.

6.3.3 Set Goals: The settings would deal with goals and target being set along with the type of workout/burnout is required from user point of view. The goals are completely customizable and are linked to the page 'Summary' that will be available to the user on one-click for routine monitoring.

6.3.4 Calorie Monitor: The calorie monitor page is linked to API from USDA, that gives a huge database to the item value and their calorific value based on the standards obtained. This standard is added along with a add button, that would reflect on the quantity of items the user have consumed.

Chapter 7. Performance and Benchmarks

7.1 Performance Analysis

Performance of the application is tested with modules in each iteration. The quality of the product will be improved with the help of Load testing, Stress Testing and Capacity testing. Performance is the measure of how the application handles itself under different circumstances without crashing and closing down. The optimal solution is to carry out testing all the factors of integration and the timeout time involved in the structure of the code. This would resolve bottleneck present. A set of definitive test scares and scenarios pertinent to the performance and pressure will result in smooth running of application.

To run a performance on the basis of load and concurrency typically refers to methods used for preserving a set of data. If users access a particular Goal, and change the values of the target goals, the result should be an outcome of changes. This is either implemented with the help of timestamps or concurrency techniques. In our project, we typically associate any set of operation to a certain timestamp(T_s) in server database system. The SQL Server also provides an interface to all concurrency techniques to resolve the integrity issue.

7.1.1 Concurrency Handling with Optimistic Read Write

In our project we use CONCUR_SS_OPTIMISTIC_CC to support the driver to monitor concurrency based on timestamps. This type of a concurrency check enables the database to compare the row values with the presence of a timestamp values in each row. We have a table that comprises of Timestamp and count values, using this further, we also compare with OPTIMISTIC VALUES i.e. the values that are set with greater timestamp, and the updated values is set to priority, and the integrity of row values is saved with a change in goal/task from the user's perspective.

7.1.2 Code Optimization and Reduce Redundancy

We have implemented the project with optimized memory management, by holding less count of each object on the basis of memory management. Less use of unnecessary objects and data clean up done in Phase III of the project, helped us structure the code with identifying objects without modularity and eliminated their use from the code. Another way to optimize the code is by making the use of Static variables as and when needed. The values of the the calorific values for instance is linked to USDA, and by adding these items in our routine are set as static values and be called, during the functional call made in the stack.

7.1.3 Frames and Performance Using Systrace

Systrace is a command that allows the developer to inspect the timing information on how the system interacts with their process with the application at the system level. It basically combines data from all the kernel such as scheduler, activity of disk, threads and operations and much through. All this together combines to form an HTML Report.

7.2 UI Performance

Systrace is basically used to analyse code and frame rate to identify the developer with the problems and suggest with possible solutions.

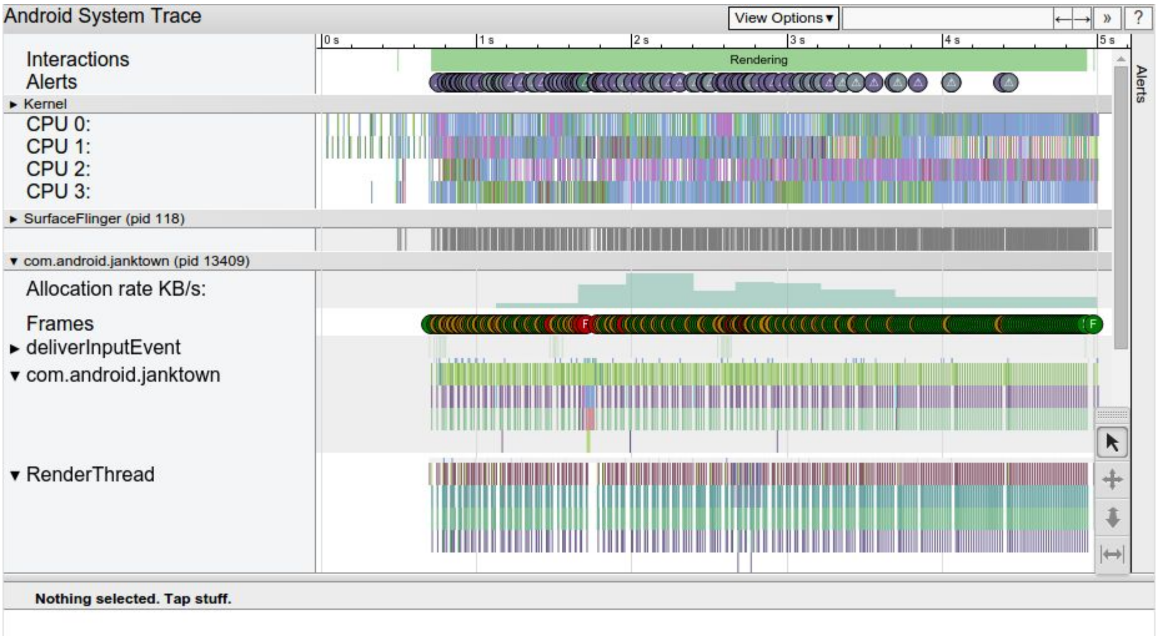
7.2.1 We ran out Application Total Health by means of Emulator on Android Studio.

7.2.2 Using the systrace command on the terminal

```
$ python systrace -t 10 [other-options] [categories]
```

7.2.3 We interact with the application which systrace is running continually and navigate through the pages of the application.

7.2.4 Systrace generated an HTML report once the defined time limit has elapsed.



Nothing selected. Tap stuff.

Figure 21. The HTML Report obtained from running Systrace in Terminal while interacting with frames/pages in application Total Health

7.3 Project Benchmark and Goals

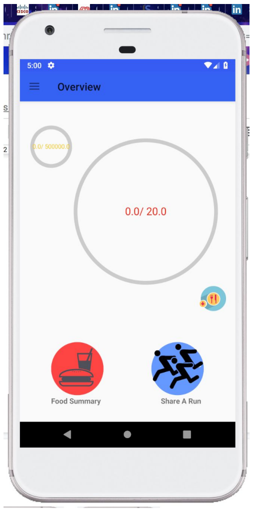
During the process of developing this Android based application, we have gone through ample amount of requirement changes and benchmarks that have evolved this project into a final bloom. The most

7.3.1 Database Normalization

The utmost benchmark of this project is normalization of all type of data. The input of an item in the calorie count, and their manipulation with integers with calorie count up and burn down, was a challenge in itself. Use of internal functions that inputs a take on string, pulling out the calorie with USDA calorific value was a critical benchmark experience. The benchmark of this kind of normalization helped us in analyzing a statistical model through which we could realize the requirement and the performance based requirement of the Android based application to achieve great customer satisfaction and success rates.

7.3.2 Database Confidentiality

Database confidentiality is critical/sensitive data stored in a shell that are accessible to users with only certain privilege. Here we have access to the Goals/Task/History of user data defined to certain user only provided, they are logged into the application and are available for making changes that are only visible to them in a private mode, and are visible to their peers upon sharing the goal.



Share A Run is a scenario which includes the data transmission that is secured through different layers of protection. This amply defines the degree of exposure to data to themselves as well as their peer. The person who receives the data, wouldn't be able to see the calorific value in exposure and it is certainly an example of abstract data.

Chapter 8. Deployment, Operations, Maintenance

8.1 Deployment and Activities

Deployment and activities are are usually a general process basically customizable for future changes or ad-hoc requirements to be accommodated as and when required. The activities usually are concerning to either producer side or the consumer side. On the basis of the idea of deployment, a producer is able to test the application in a proxy real time environment. This enables the testing to be more fierce and make the application robust and prone to security breaches, quality applications, with user-friendly behaviour. Overall, software deployment is one way to make the working application made available for all the functionality to be tested.

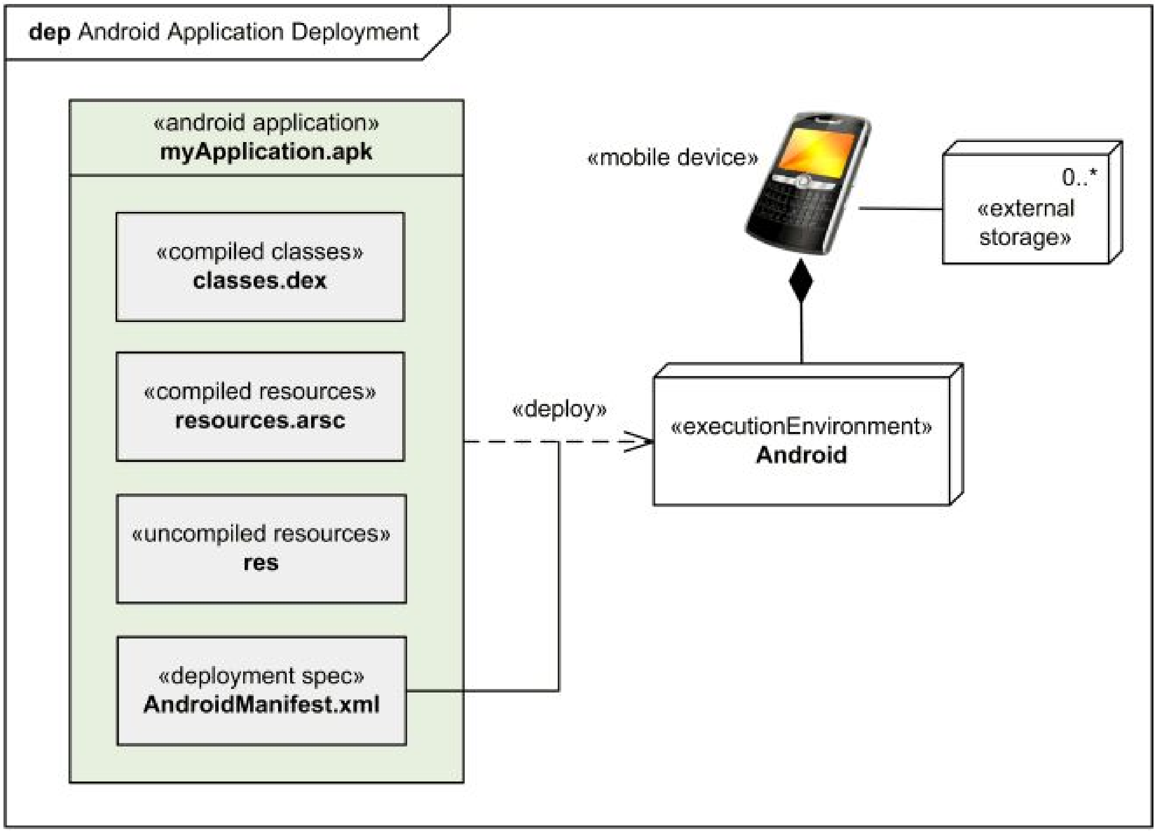


Figure 22. The above figure shows the interdependence of modules and their Deployment in Real Time Environment.

For the purpose of deployment, our application is deployed on Google Cloud with all documentation and relevant resources being shared on Google drive. Coming to publishing code, we will be publishing the code on Github web page. To have an internal control on application code monitoring and source code changes, all the team members modify the code on GIT for version control and branching. To avoid any disruption, we continually follow regular integration and regression based test cases are obtained from our shared library to execute the same.

8.2 Operational Needs

The modules should interact with one another post the design of the application is complete as follows:

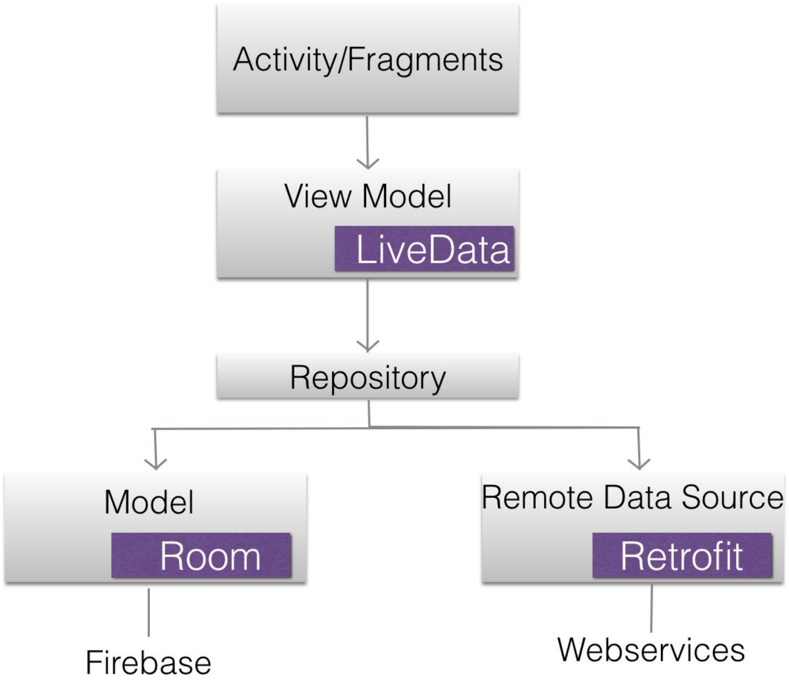


Figure 23. Component design of the application.

In the figure above, notice how each component is dependent on the components primarily below them. All the activities and fragments are completely dependent on the ViewModel for all the live data and dependencies under them. Likewise, View Model will be dependent on the Object Repository. The repository alone is dependent on multiple other classes, in this example repository depends on a very evident data model and a remote source at the backend.

The purpose if this design serves to consistent data distribution (Normalization) and pleasant user experience. Irrespective of the user's regularity with accessing the application Total Health, the design and operational model gives us a control on consistent data being available locally on the user phone for them to be able to access the application with valid credentials and instantly able to access the data of the user on the web page. If the data in the object repository is stale, the application starting updating the data in the background and clears cache with respect to new data being made available.

8.3 Maintenance

Maintaining an android application is important to revise the needs of requirement and updates on the basis of continuous integration along with the change in revisions. Consistent updates are the most important changes that should be accommodated, as the application requires performance analysis and avoid crashing at any point of time. User feedback, for better usability of functions/features are also essential to accommodate any change in requirement and scope. An application should be viable for OS updates, handle scalability, and interruptions. All of modules are designed with respect to handling changes and consistent performance.

Chapter 9. Summary, Conclusions, and Recommendations

9.1 Summary

Total Health has been developed with the view of tracking the record of personal fitness of the user. The application has been designed with a view of modularity, scalability and lightweight. The code designing has been done in Android Studio with Java as the programming language. Basic information has been collected from the user to calculate BMI and WHR. User is provided with a flexibility to set goals. The user also provides the input about the food consumed. The application compares this feed with the USDA data and informs the user about his daily calorie intake. Thus, by keeping a track of the activities and the calorie intake the application helps the user to achieve desired fitness.

9.2 Conclusions

The ultimate goal of the application Total Health is to be of suitable standard for users with different health and dietary platform across the globe. In pursuit of advocating to living a healthier lifestyle and workout routines, it is essential for a user track their intake and of how these dietary constraints affect their vitals.

9.3 Recommendations for Further Research

The application can be designed to collect the data from the sensors such as a fitbit. This recorded data consisting of the daily activities and the calorie intake can be send to the server where a machine learning model can be implemented. The model can provide the fitness desired by the user. It can provide recommendation about the calorie intake and the workout activities so that the user can achieve desired fitness in minimal time duration. More accurate information can be provided by analyzing the smoking and drinking habits of the user. Future improvements may also include the options among different food categories if a person is allergic to certain food types. An option for calorie intake according to different cuisines can also be included.

Glossary

Body-Mass-Index

Body Mass Index is is the ratio calculated by one’s weight in Kilogram by square of height in meters.

Waist to Hip Ratio

Dimensionless ratio of the circumference of the waist to that of the hips.

Machine Learning

“Machine learning is the science of getting computers to act without being explicitly programmed."

Weighted Majority Algorithm

This is a machine learning algorithm that is a consolidation of many algorithms put together in a pool. This provides a basis to make less mistakes, make faster learnings, and is more tolerant to user inputs and criterias.

References

[1] Rebolledo-Nandi, Z., et al. "Design of a versatile low cost mobile healthcare monitoring system using an android application." *Health Care Exchanges (PAHCE), 2015 Pan American*. IEEE, 2015.

[2] W. H. Organization. Available: http://www.who.int/topics/chronic_diseases/es/

[3] Imteaj, Ahmed, and Muhammad Kamrul Hossain. "A smartphone based application to improve the health care system of Bangladesh." Medical Engineering, Health Informatics and Technology (MediTec), 2016 International Conference on. IEEE, 2016.

[4] Bajaj, Deepali, et al. "Android based nutritional intake tracking application for handheld systems." 2017 8th International Conference on Computing, Communication and Networking Technologies (ICCCNT). IEEE, 2017.

[5] https://www.kaggle.com/yersever/500-person-gender-height-weight-bodymassindex

[6] Kong, Xiaomin, et al. "Design on mobile health service system based on Android platform." *Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC), 2016 IEEE*. IEEE, 2016.

[7] Prazina, Irfan, et al. "Usage of Android device in interaction with 3D virtual objects." *2017 40th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*. IEEE, 2017.

[8] Bhawar, Pooja, Nitin Ayer, and Sameer Sahasrabudhe. "Methodology to create optimized 3d models using blender for android devices." *Technology for Education (T4E), 2013 IEEE Fifth International Conference on*. IEEE, 2013.

[9] https://www.techsoft3d.com/products/hoops/visualize/

[10] Hales, Craig M., et al. *Prevalence of obesity among adults and youth: United States, 2015-2016*. US Department of Health and Human Services, Centers for Disease Control and Prevention, National Center for Health Statistics, 2017.

[11] Meier, Reto. *Professional Android 4 application development*. John Wiley & Sons, 2012.

[12] https://www.mulesoft.com/resources/api/what-is-rest-api-design

[13]https://towardsdatascience.com/selecting-the-best-machine-learning-algorithm-for-your-regression-problem-20c330bad4ef

[14] https://www.perfecto.io/

[15] https://www.spatial.com/products/3d-visualization

[16] Wettschereck, D. and Aha, D. (2019). *Weighting features*.

Appendices

Appendix A.

The development of code and repository generation is done in Android Studio. Using Git, we are able to achieve branching and consecutive integration on our application.

The implementation and development of application was around front-end development, database management using firebase, and backend development. In the front-end we have used web technologies with XML, Google API, and using Emulators on Android SDK. The backend of the program is supported by libraries, package and functions in Java. The firebase is the database in use on cloud platform, and resultant of APIs draw value through other server.

The use of Machine Learning Algorithm i.e. Weighted Majority Algorithm provides the user with more accuracy to predict the duration as this kind of algorithm provides very less pre-processing, and also have a better tolerance to interacting features and high learning rates[16].

During the course of development, we have adapted to Agile Methodology to better implement modules, and build the components and units integratively at every sprint to produce a working model.