**COSC​ ​6355​ ​–​ ​Introduction​ ​to Ubiquitous​ ​Computing**



**HealthAI**

Personal Health Monitor that moves with you.

Table of Contents

[1. Introduction: 4](#_Toc531861680)

[2. UML Diagrams 4](#_Toc531861681)

[2.1 Use Case diagram: 4](#_Toc531861682)

[2.2 Work Flow Diagrams 5](#_Toc531861683)

[Register Module 5](#_Toc531861684)

[Login Module 5](#_Toc531861685)

[Edit Profile 6](#_Toc531861686)

[Calendar Activity 6](#_Toc531861687)

[Diabetes Predictor Model 6](#_Toc531861688)

[BMI Calculator 6](#_Toc531861689)

[Calorie Counter 7](#_Toc531861690)

[Steps Tracker 7](#_Toc531861691)

[Workout 7](#_Toc531861692)

[2.3 Database Schema 7](#_Toc531861693)

[Record Structure 7](#_Toc531861694)

[Database Structure 8](#_Toc531861695)

[3. Use Cases 8](#_Toc531861696)

[3.1 Register Module 8](#_Toc531861697)

[3.2 Login Module - Dashboard 10](#_Toc531861698)

[Forgot Password: 10](#_Toc531861699)

[Weather Feature: 11](#_Toc531861700)

[Edit Background Picture: 11](#_Toc531861701)

[Tile Features: 12](#_Toc531861702)

[3.3 Navigation Drawer 12](#_Toc531861703)

[Edit Profile: 12](#_Toc531861704)

[Calendar View: 13](#_Toc531861705)

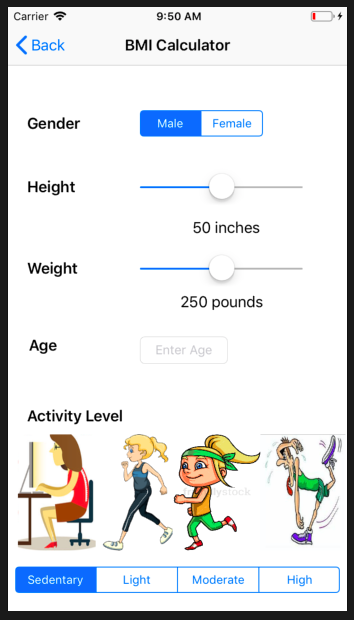
[Sign-out: 13](#_Toc531861706)

[3.4 Diabetes Predictive Model 14](#_Toc531861707)

[Predict Risk: 14](#_Toc531861708)

[Future Prediction: 15](#_Toc531861709)

[3.5 BMI Calculator 16](#_Toc531861710)

[ 16](#_Toc531861711)

[Go Button: 16](#_Toc531861712)

[Maintain: 17](#_Toc531861713)

[3.6 Calorie Counter 17](#_Toc531861714)

[Add: 18](#_Toc531861715)

[Calorie Data: 18](#_Toc531861716)

[Set Goal: 19](#_Toc531861717)

[3.7 Steps Tracker 19](#_Toc531861718)

[Steps History: 19](#_Toc531861719)

[Calories Burned: 20](#_Toc531861720)

[3.8 Workout Module 21](#_Toc531861721)

[Strength Workout: 21](#_Toc531861722)

[Cardio Workout: 22](#_Toc531861723)

[Video Tutorial: 22](#_Toc531861724)

[4.Protocols 23](#_Toc531861725)

[5. Testing Plan & Version Control 23](#_Toc531861726)

[5.1 Objectives 23](#_Toc531861727)

[5.2 Integration Testing 23](#_Toc531861728)

[5.3 User Acceptance Testing 23](#_Toc531861729)

[6.Sytem Requirements & Dependencies 23](#_Toc531861730)

[7. Challenges 24](#_Toc531861731)

[7.1 Team Communication 24](#_Toc531861732)

[7.2 Integrating Cocoa Pods 24](#_Toc531861733)

[8. Conclusion 24](#_Toc531861734)

# Introduction:

Organizing and managing your health in a fast-paced world has become the dream of the rich – who hire personal dieticians and doctors.

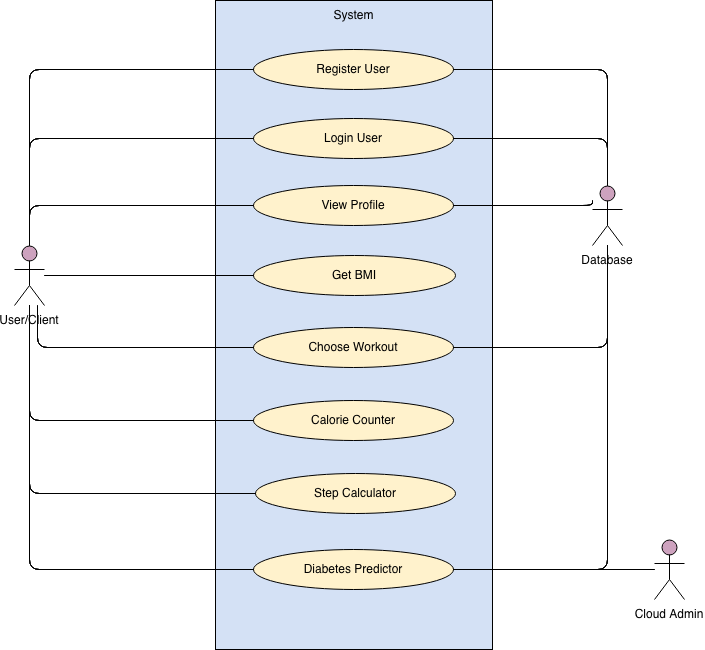
Keeping a track of your nutrient intake, planning your meals through the day, keeping a check on the exercise your body needs, trying to stay in shape – all these require a high price. With HealthAI everything is free of cost so that anybody can afford to lead a healthy and happy life.

Worried that you might have Diabetes ?

HealthAI comes with a Machine Learning predictive model that predicts your current health condition based on your daily activity.

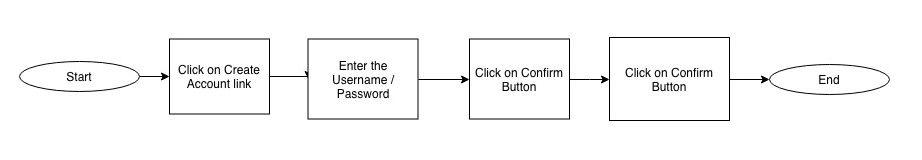
# UML Diagrams

## 2.1 Use Case diagram:

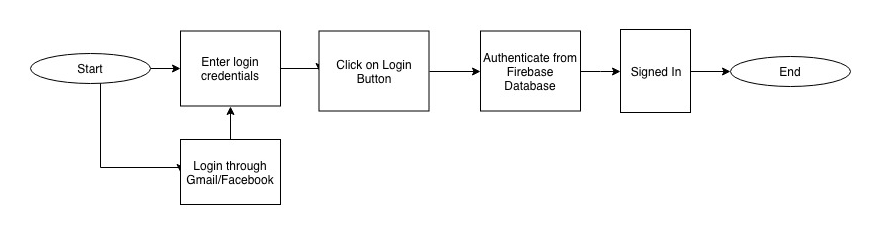


## Work Flow Diagrams

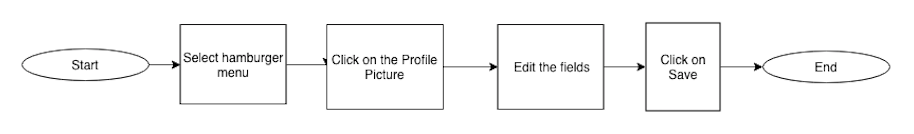
### Register Module



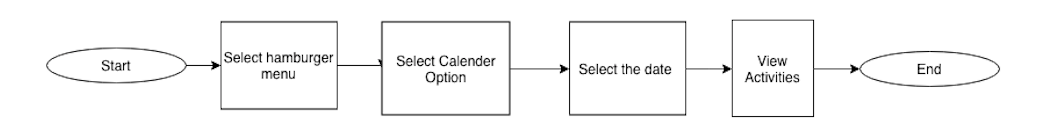
### Login Module



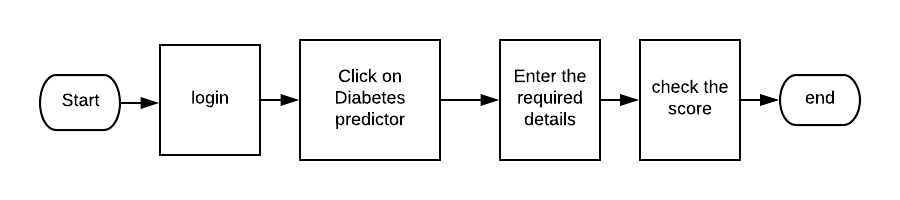
### Edit Profile



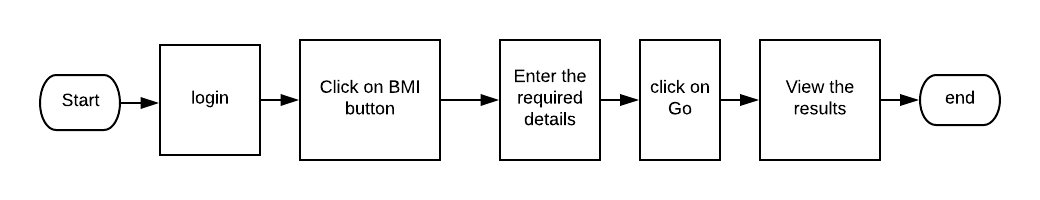
### Calendar Activity



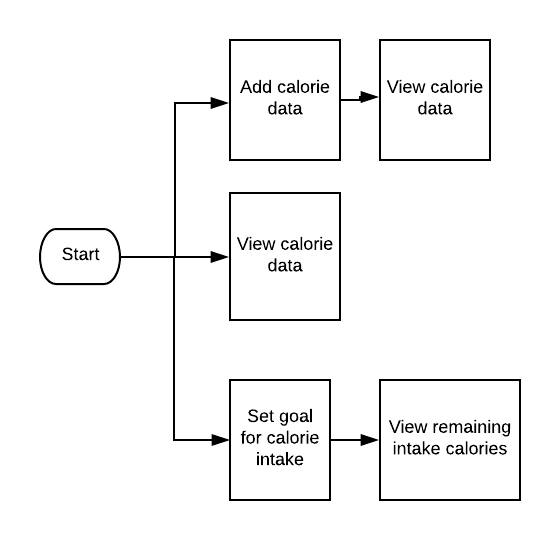
### Diabetes Predictor Model



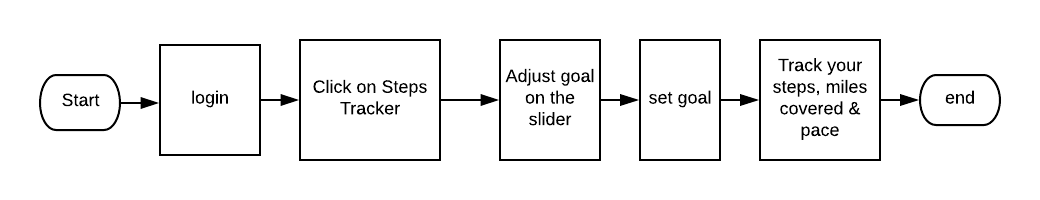
### BMI Calculator



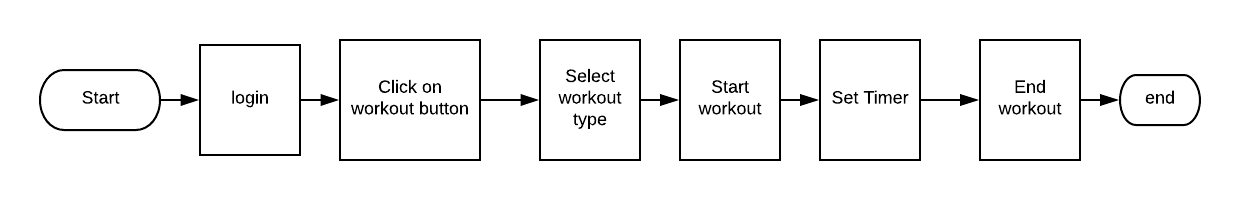
### Calorie Counter



### Steps Tracker

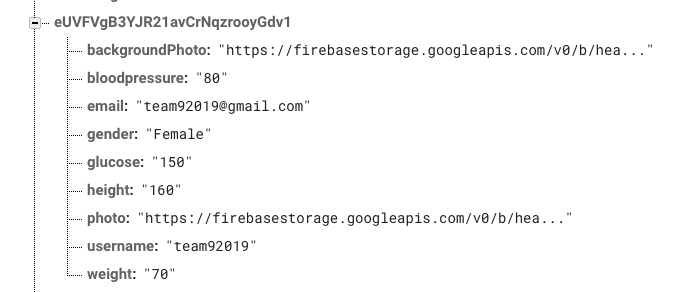


### Workout

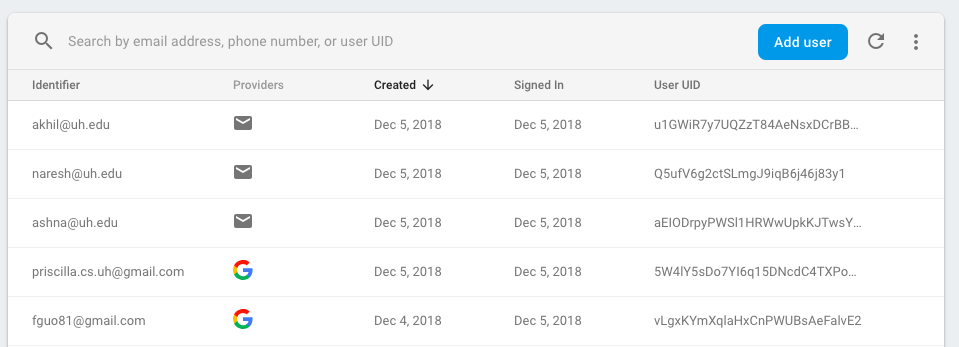


## Database Schema

### Record Structure



### Database Structure



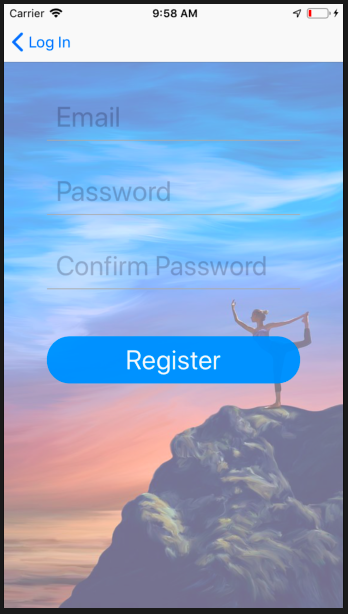
# Use Cases

## 3.1 Register Module

There are two ways to register to the app

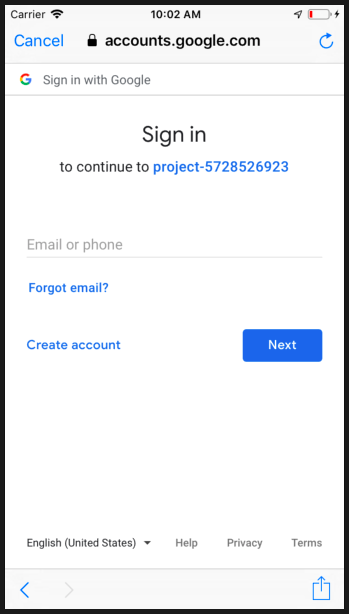
1. Creating a Normal Account

Click on “Create an Account” and enter the details. Now, Click on register Button.



1. Registering using Google/Facebook Account

Click on Google/Facebook symbols and enter your credentials. Now Login using these credentials.

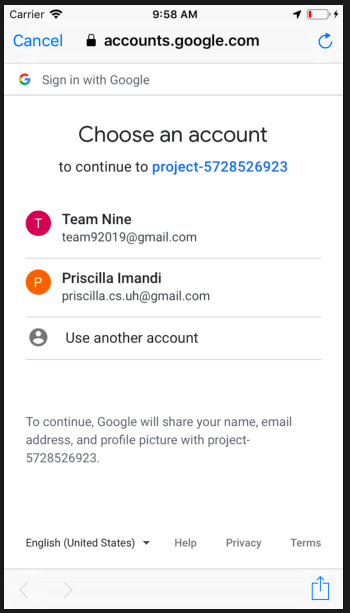
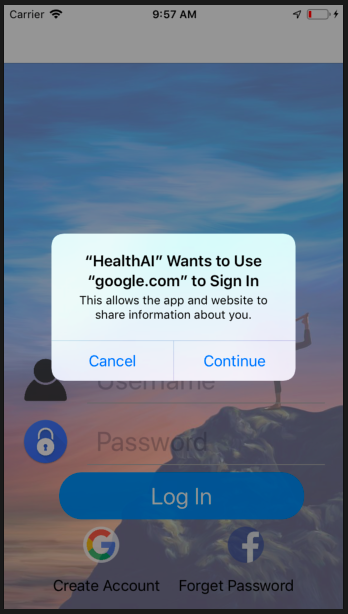


On registering, an account is created in the Firebase Database with the following fields. The email, username, background photo and photo are the default fields which are automatically populated in the database on creating the account.

1. Background Photo
2. Username
3. Email
4. Gender
5. Glucose
6. Blood Pressure
7. Height
8. Weight
9. Photo

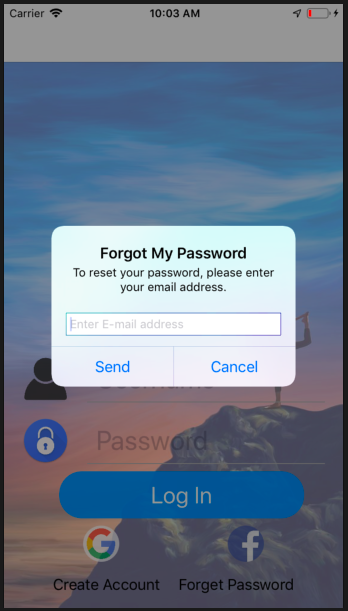
## 3.2 Login Module - Dashboard

This is the main login screen of the app, where you enter your credentials. These credentials are authenticated using the Firebase API.



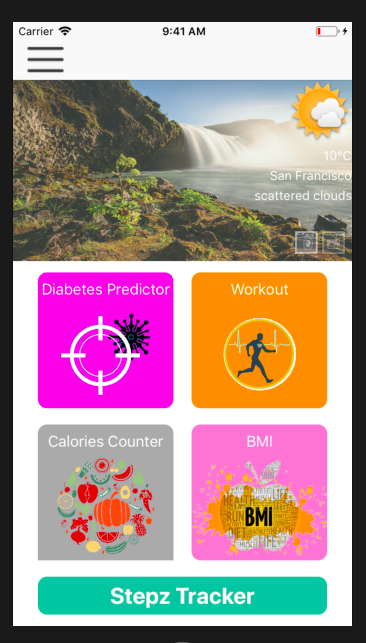
### Forgot Password:

Using this feature, we can click the link, an alert pops up to enter the email id and a reset password link can be sent to your account to reset the password. And using this new password, we can login again.



### Weather Feature:

This feature enables us to see the weather at the current location. This helps us to decide and do the workouts and other activities.



### Edit Background Picture:

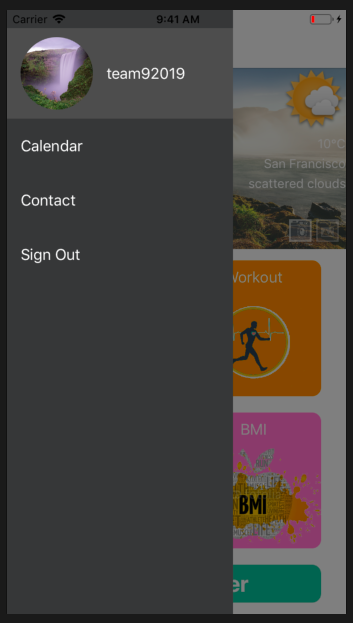
On clicking the Camera/Choose from Gallery icons located on the background image enables us to modify the background image.

### Tile Features:

All the different functionalities of the tiles are listed as tiles on the dashboard – Diabetes Predictor, Workout, Calories Counter, BMI Calculator and Steps Tracker.

## 3.3 Navigation Drawer

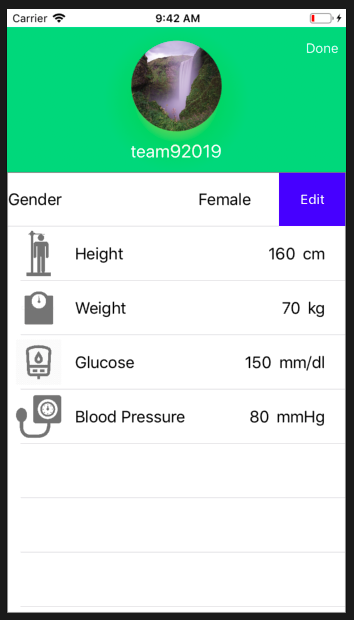
The navigation drawer is located on the top left corner. This has a profile picture which is used to edit the profile, a calendar button, a contact button and a sign-out button.



### Edit Profile:

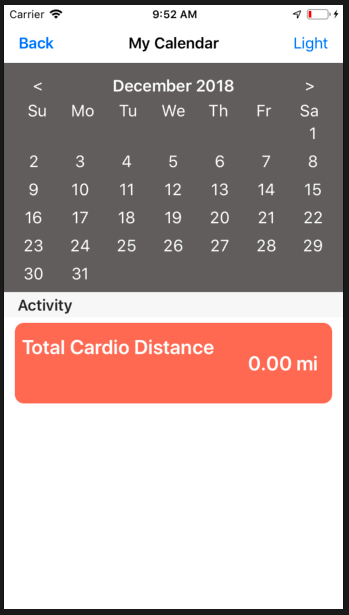
Click on the profile picture which is the Edit button. The user can edit the profile with the following fields by swiping to the left. An alert pops up to enter the new value and save it. This would update the data in Firebase Database.

1. Gender(female/male)
2. Height(cm/in)
3. Weight(kg/lb)
4. Glucose(mm/dl)
5. Blood Pressure(mm/Hg)



### Calendar View:

On selecting this button, we would reach the calendar page which lists all the activities of a particular day. Changing the theme of the calendar can also be done.



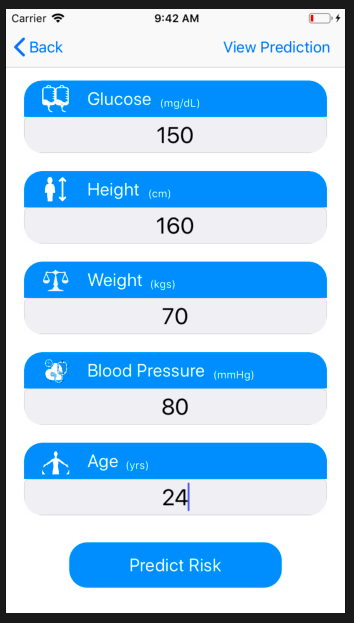
### Sign-out:

This button enables us to completely signout off the application. But does not exit the application.

## 3.4 Diabetes Predictive Model

On clicking the Diabetes Predictor tile, we are redirected to a form. This form is pre-populated with the data from the database. We can see the following fields.

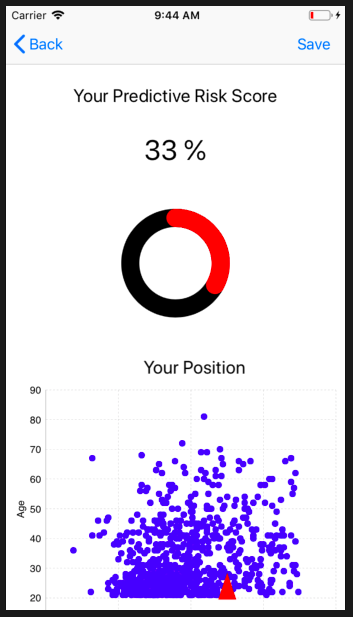
1. Glucose (mm/dL)
2. Height (cm)
3. Weight (kgs)
4. Blood Pressure (mm/Hg)
5. Age (yrs)



### Predict Risk:

On clicking the “Predict Risk” button, we can we the following

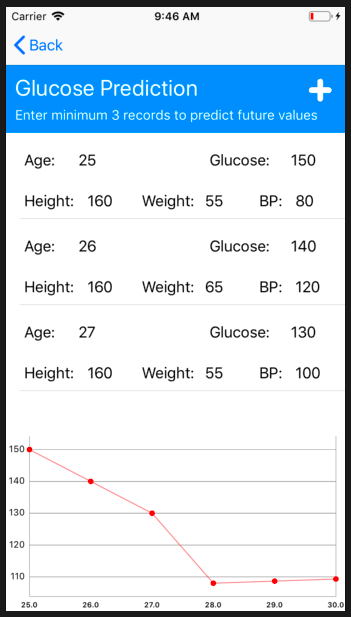
1. The score of Risk for Diabetes using our current data. This is done using a Machine learning – logistic regression model. The model has been trained using 3000 records and stored on Amazon AWS. This is called using a Rest API in the UI.
2. A graph which shows where we stand in a population of users.

The save button enables us to save the report as a .pdf file on the local device.

### Future Prediction:

On clicking the “Future Prediction” bar button, we can view the following

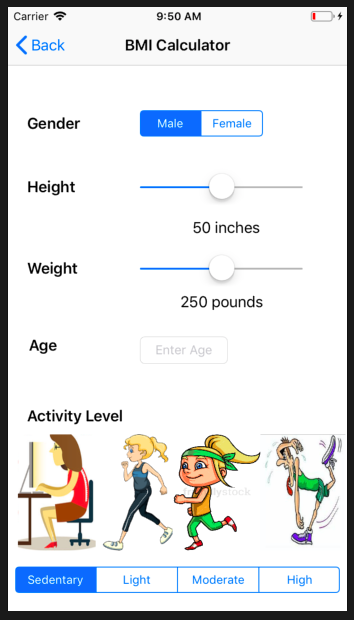
1. Data records of the user
2. Graph showing the prediction – Estimated using the glucose levels using the previous records.



## 3.5 BMI Calculator

On clicking the BMI Calculator tile, we are directed to a form. This form contains the following pre-populated fields.

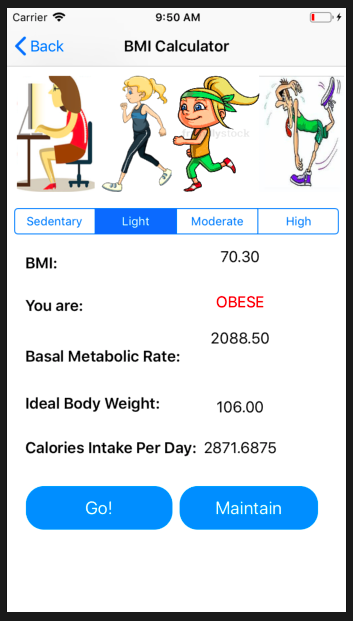
1. Gender
2. Height
3. Weight
4. Age
5. Activity Level



### Go Button:

On clicking the Go button, we can calculate the following values:

1. Body Mass Index – This is a measure of body fat in adults.
2. Basal Metabolic Rate – Rate at which the body uses energy to keep the vital functions running.
3. Current Body State – Whether a body is normal, under-weight or obese
4. Ideal Body Weight – The ideal weight a person should be for that age and height.
5. Calories Required per Day – The calories required to maintain the ideal body weight.



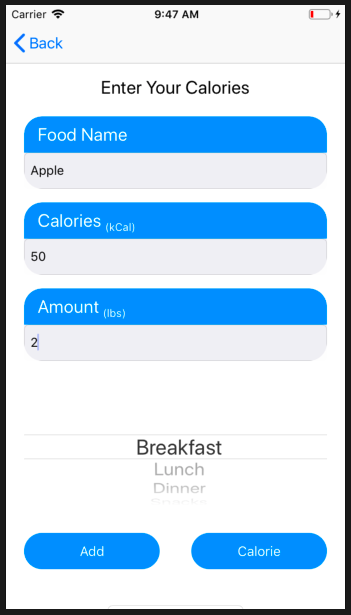
### Maintain:

On clicking the maintain button, we are directed to a screen which shows the foods that we have to eat based on our body type. These are the ideal foods that have to be eaten. They are changed based on the data that is given in the previous page.

## 3.6 Calorie Counter

This feature helps us to know how the calorie consumption of the users.We have the following fields:

1. Food Name
2. Calories
3. Amount

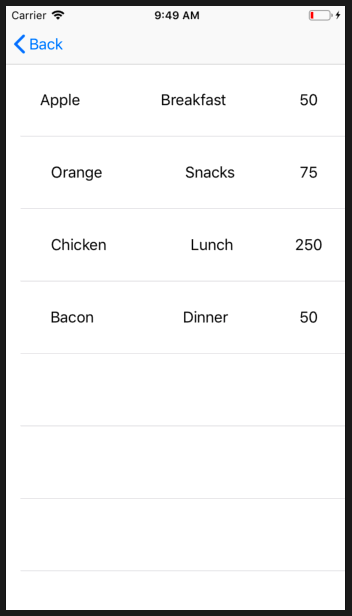


### Add:

The foods will be added to our list of foods in the Realm database.

### Calorie Data:

To view the list of foods already entered.



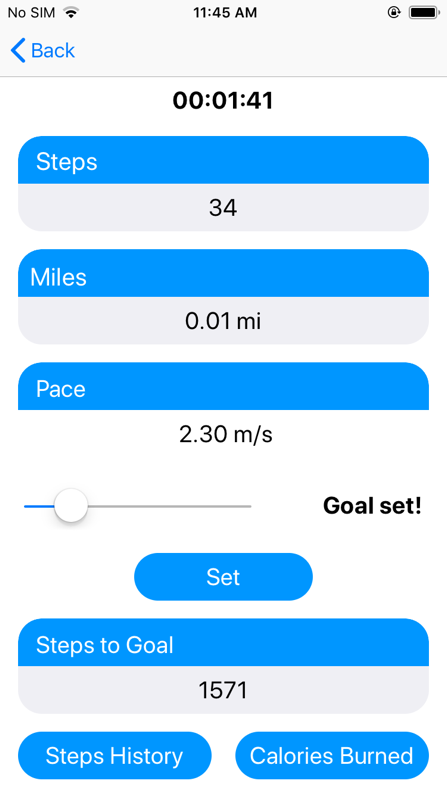
### Set Goal:

We can enter the calories that we intend to consume and set a goal. We are redirected to a graph that shows how many calories we are short of attaining our goal in the next view. A pie chart depicts the calories we need to meet our goal.

## 3.7 Steps Tracker

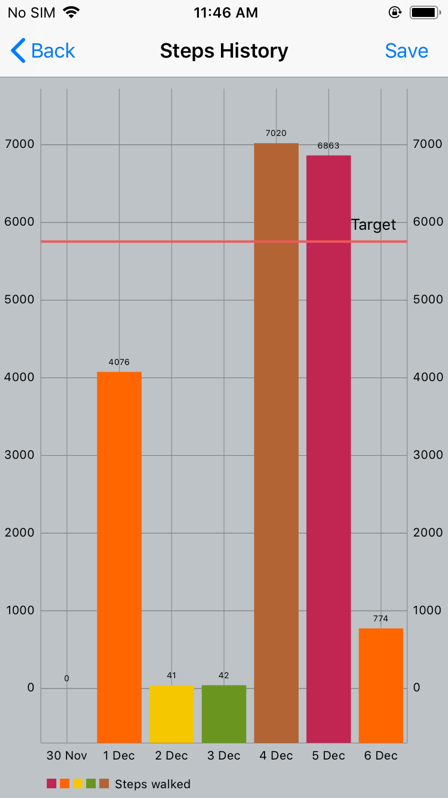
This feature is a pedometer which records the number of steps the user takes. We can find the following data

1. Steps taken
2. Distance Covered
3. Pace of the person
4. Goal of Steps – A number of steps that we need to take for the day.
5. Steps to Goal



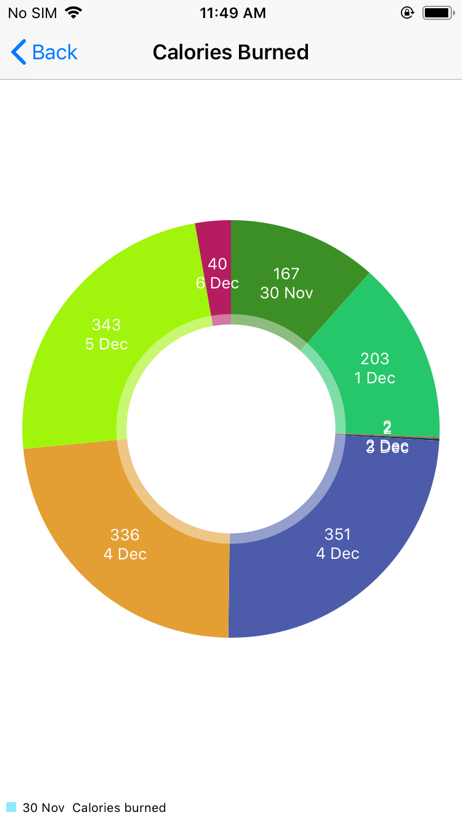
### Steps History:

Shows a graph of steps taken over the past week.



### Calories Burned:

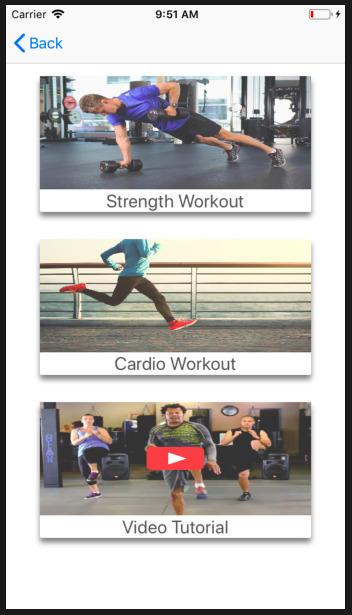
Calories burnt based on steps taken.



## 3.8 Workout Module

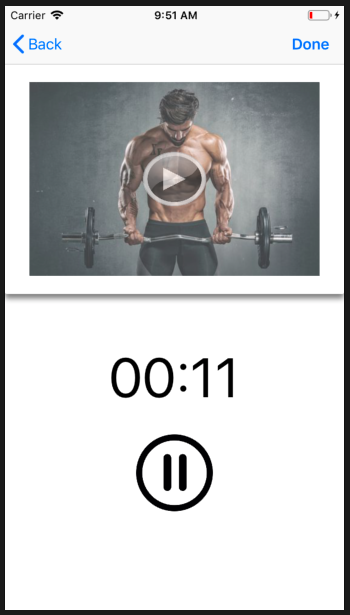
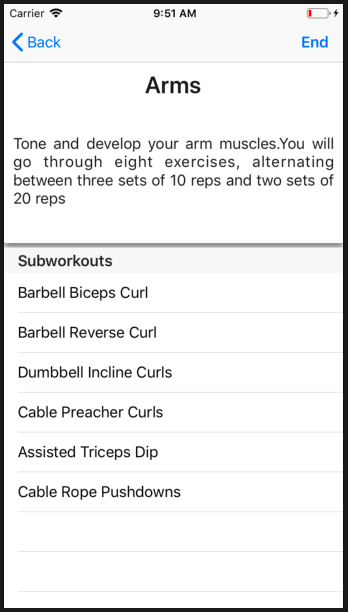
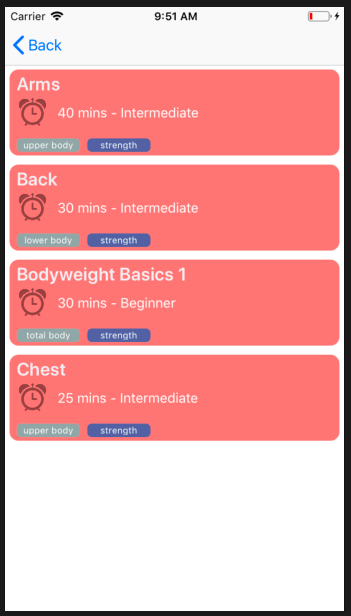
This module keeps a track of all the workouts that we can do

1. Strength Workout – Arms, Back, Bodyweight Basics, Chest
2. Cardio Workout – Running, Cycling, Walking
3. Video Tutorial



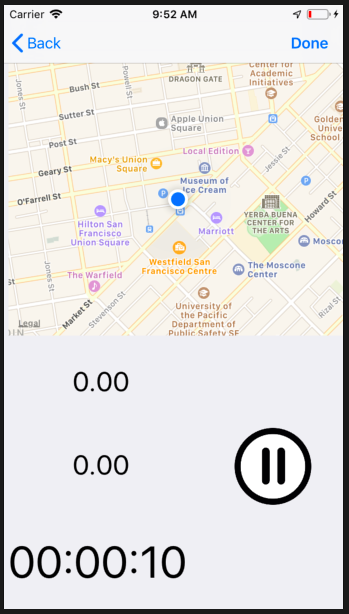
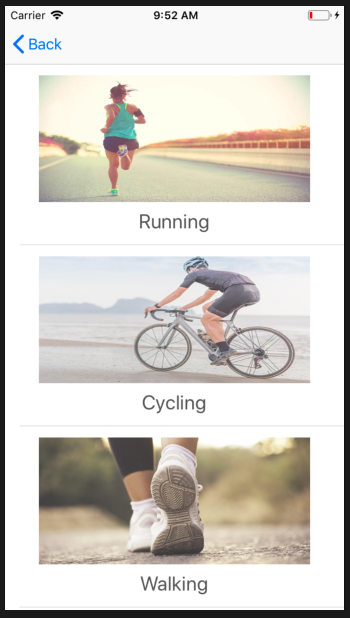
### Strength Workout:

On selecting either the strength or cardio workout we can record the time of the workouts by selecting any of sub-workouts. Once the workout is completed, you can save the workout by clicking Save on the Subworkout and End on the workout page. These activities can be saved and checked in the calender.



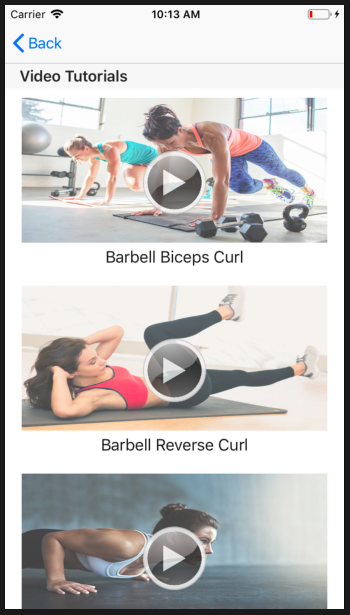
### Cardio Workout:

Select the workout you want to perform, then click the play button. This shows our location, distance and speed. On clicking the Done button, it is saved to the activities on the calendar.



### Video Tutorial:

Shows a list of videos of how the workouts can be performed. Using these, we can perform the workouts.



# 4.Protocols

**4.1 DataTransferDelegate**

This delegate has a function – UserDidFinishedSubworkout , which sends subworkout data to the previous workout view controller.

# 5. Testing Plan & Version Control

## 5.1 Objectives

Our objective is to produce a fully working bug-free application.

## 5.2 Integration Testing

As our app has multiple modules, we first have to ensure that all of the modules integrate and work seamlessly as a single unit. We have used github to ensure that the versions are consistent. But we had the issue that all the members did not have access to personal Macbooks – so a consistent and updated version should always be available in the lab Mac system.

This has been done after completion of every module on a weekly basis.

Each of the modules have been individually tested for robustness of individual modules. Also, we have tested it for edge cases and alerts will be sent to the users for any inconsistent data.

This has been done as and when the modules have been modified.

## 5.3 User Acceptance Testing

We have selected 7 users from friends and family who have been using android and ios. We have done this to eliminate bias from users who might be accustomed to ios. We have received suggestions on placement of buttons on the screens which have helped us make the app better.

This has been done once during mid-November and again at the end of November.

# 6.Sytem Requirements & Dependencies

Macbook(Simulator- except Steps Tracker), iPhone using Swift 4.

1. Patient Data from Kaggle Datasets
2. Food Data from Usda
3. Workout Routines from Google

# 7. Challenges

## 7.1 Team Communication

Because of different schedules – it was hard to have many group meetings. Also, because of multiple modules – integrating all the modules was very difficult.

## 7.2 Integrating Cocoa Pods

Because of multiple cocoa pods like charts, pdfgenerator, Realm pods – with very less documentation it was difficult to get information on implementing the data.

# 8. Conclusion

The main objective has been achieved – a robust, seamlessly working app that shows all the data that you need.