GitHub Username: amiraHag

Get Fit

Description

This app contain a BMI calculator and tracker to help you track your BMI level, it also provide step counter to help user make simple exercise which is walking. The app can also identify user current activity which varies from still, titling to running.

Intended User

Anyone who wants to get fit can use this app.

Features

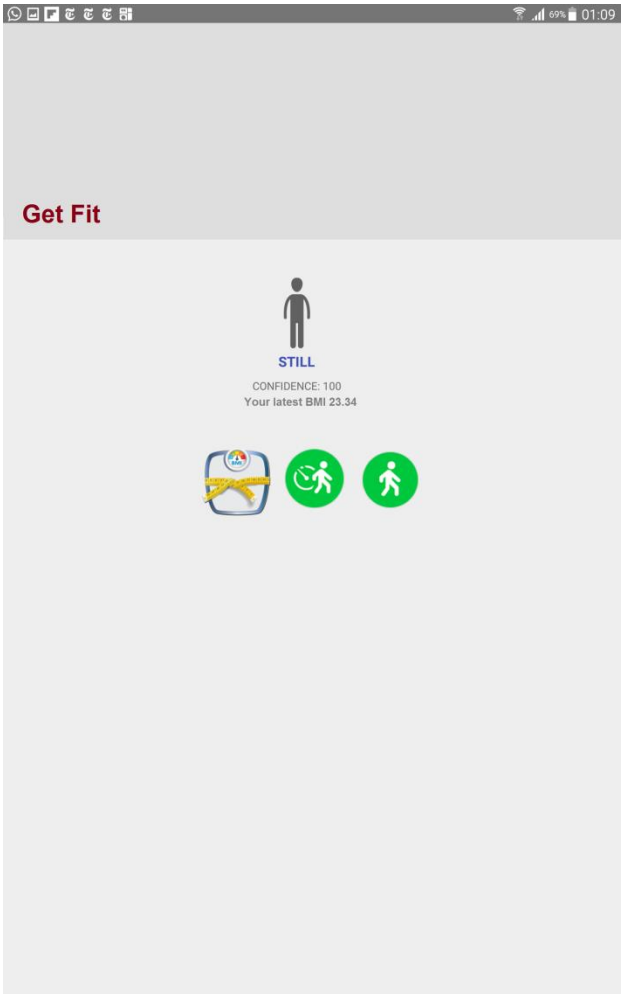
List the main features of your app. For example:

- Calculate and save your BMI
- Identify your current activity
- Count your steps and provide you history for the steps you walked within last week.

User Interface Mocks

These can be created by hand (take a photo of your drawings and insert them in this flow), or using a program like Google Drawings, www.ninjamock.com, Paper by 53, Photoshop or Balsamiq.

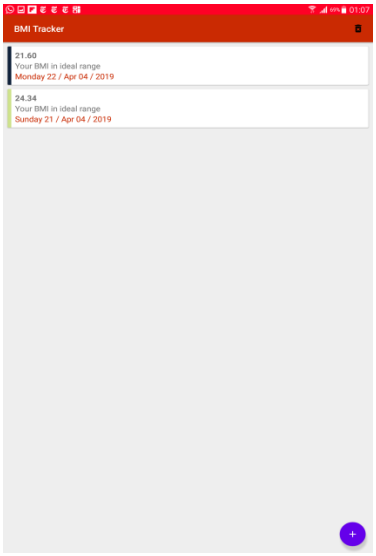
Screen 1



This is the main screen. It will display the user current activity and the user latest BMI value. The three icons will lead to different activity when clicked:

1. The first icon leads to BMI tracker activity.
2. The second icon leads to activity recognition activity.
3. The third icon leads to steps tracking activity

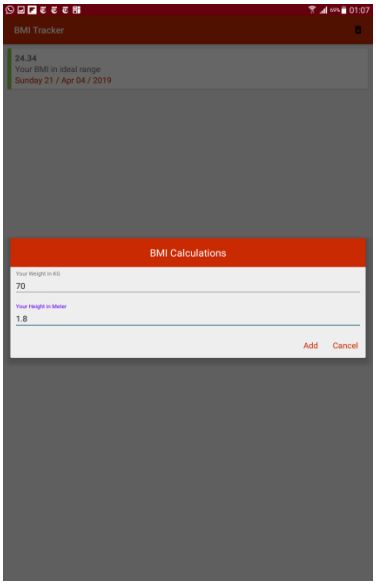
Screen 2



The second activity (BMI Tracker)
This activity will display history for the BMI values the user inputted.

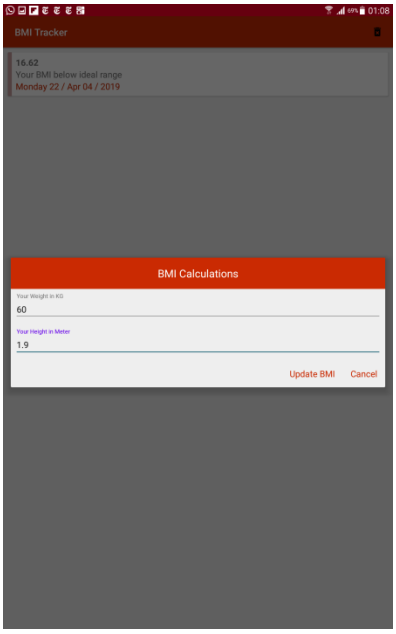
1. The user can add new value by click on the floating button (the plus button at the end of the screen).
2. The user can delete any item by swipe it right or left.
3. The user can update any value by click on the item.
4. The user can delete all items by click on the recycler ion on toolbar.

Screen 3



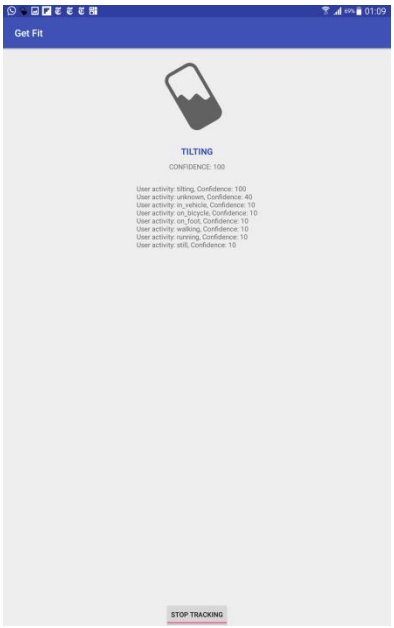
Add new BMI screen, the user will enter height and weight and the app will calculate BMI and identify its level.

Screen 4



Update BMI screen, the user will modify height and/or weight and the app will re-calculate BMI and identify its new level.

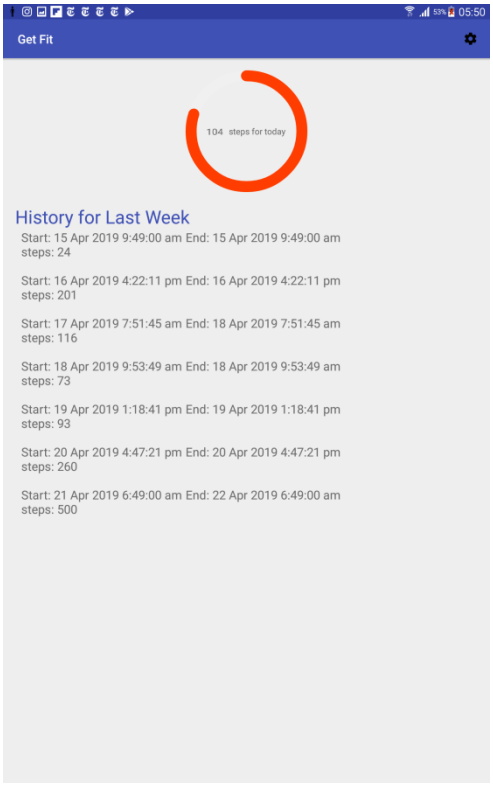
Screen 5



Activity Recognition Screen

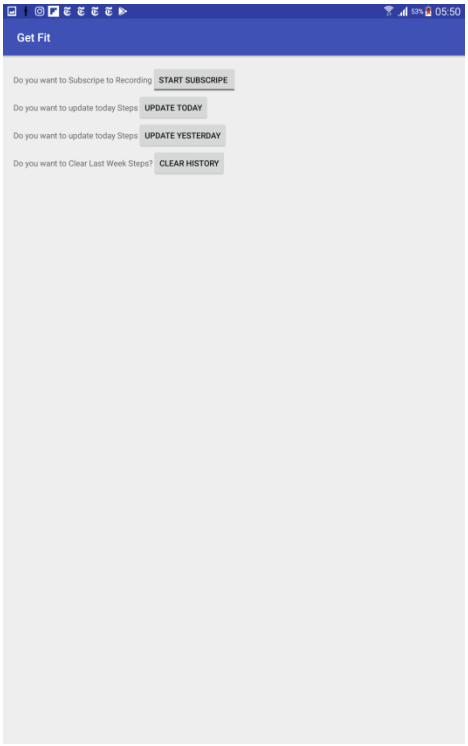
You can make you phone start identify your activity and it will send you notification when your activity status changed to walking or running (this feature can be extended to record all user activity and submit it as session directly to Google fit but for the capstone project we will identify the user activity only)

Screen 6



Steps Tracking Screen, this screen shows the user steps for today and history for the steps in the last week. The user also can manage the steps more by click on the settings icon on the toolbar which will leads to settings screen

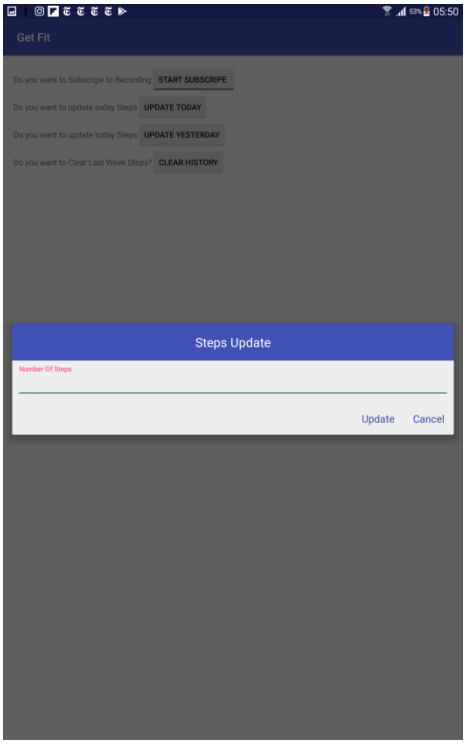
Screen 7



Settings Screen, this screen allow user to manage their step tracking history:

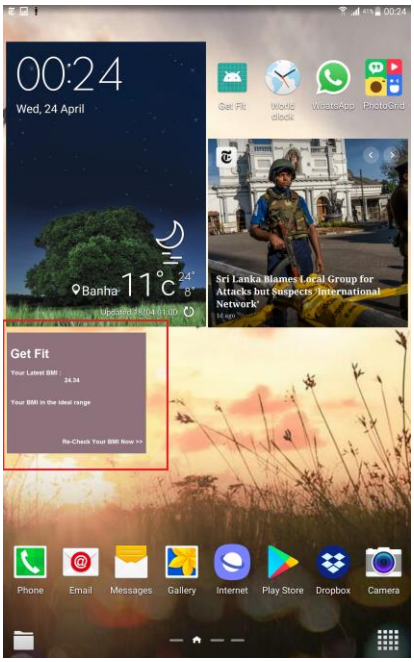
1. Remove all history by click on clear history.
2. Subscribe to recording to get all the recording data from all devices connected to your profile.
3. Update steps for today
4. Update steps for yesterday

Screen 8



Update steps screen which will appear when click on update today or update yesterday and it will allow you to update your steps manually.

Screen 9



Home Widget Screen that will display latest BMI level and the BMI status.

Key Considerations

How will your app handle data persistence?

The app will get the steps and history value from Google fit api and about BMI it will save it locally in room database.

Describe any edge or corner cases in the UX.

The app will re-identify the user activity in activity recognition screen and also re-check steps from Google Fit when the user returns to those screens.

Describe any libraries you'll be using and share your reasoning for including them.

1. Jake wharton butterknife (8.8.1) → to find and automatically cast the corresponding view in layout.
2. Txusballesteros fit-chart (1.0) -→ to draw user steps counter.
3. Room Database (1.1.1) → to store BMI values.
4. Android design Library (28.0.0) → to help in the app design features.
5. Android support Library (28.0.0) → to make backward computability for the app.
6. Google play services fitness (16.0.1) → use it to access Google Fit API.
7. Google play services auth (16.0.1) → use it to make authentication to guarantee permission for Google Fit API.
8. Google play-services-location: (11.6.0) → use to recognize user activity.
9. Google Mobile Ads (17.2.0) → to add an ad banner in the app.
10. Arch Lifecycle (1.1.1) → To help implement view model and livedata
11. Card View (28.0.0) → to show information inside card which is useful for material design.

Describe how you will implement Google Play Services or other external services.

1. Admob → use in banner ad
2. com.google.android.gms:play-services-fitness:16.6.0 → use it to access Google Fit API
3. com.google.android.gms:play-services-auth:16.6.0 → use it to make authentication to guarantee permission for Google Fit API
4. com.google.android.gms:play-services-location:11.6.0 → use to recognize user activity

General Considerations:

1. The app will build using android studio v 3.1.2 and gradle 3.1.2.
2. App is written solely in the Java Programming Language.
3. App keeps all strings in the strings.xml file.

4. App Keeps all color in color.xml
5. App Keeps all themes in style.xml
6. App enables RTL layout switching on all layouts.
7. The app includes support for accessibility. All images have content descriptions and The app have navigation using a D-pad specially on the BMI items in the BMI tracker.
8. App use AsyncTask to get history data for steps and also to update steps.
9. App utilizes stable release versions of all libraries, Gradle, and Android Studio.

Next Steps: Required Tasks

This is the section where you can take the main features of your app (declared above) and break them down into tangible technical tasks that you can complete one at a time until you have a finished app.

Task 1: Project Setup

- Create Google developer project and enable the Fitness API.
- Create an OAuth 2.0 client ID
- Connect to the Fitness API in the app.
- Setup gradle dependency.
- Setup libraries that required to the app.

Task 2: Implement UI for Each Activity and Fragment

- Build UI for each activity.
- Build two layouts for the dialog box one will be used in add and update BMI and the other will be used in add steps
- Build two menus, one for BMI tracker activity and the other for step tracking activity.

Task 3: Implement Room Database AND Live Data

- Creating a Repository Class/Presentation Layer.
- Implement ViewModel.
- Add Adapter and RecyclerView.
- Populate the Database.
- Connect UI and Data.
- Create Add BMI dialog.

Task 4: Implement User Activity Recognition

- Access the data from google service location.

- Implement activity identification function which identify the activity and update the UI.

Task 5: Implement Step Tracking

- Access sensor data from Sensor Google Fit API to get the number of steps live.
- Implement step tracking function to update UI with the sensor data.
- Implement Subscribe function to subscribe to Recording Google Fit API to get the number of steps from all connected devices.
- Access History data from History Google Fit API to get the number of steps for the last week.
- Implement History tracking function to update UI with the history data and also to allow user to modify this history.

Task 6: Add Widget

- Setup widget to display the user latest BMI value

Submission Instructions

- After you've completed all the sections, download this document as a PDF [File → Download as PDF]
    - Make sure the PDF is named "**Capstone_Stage1.pdf**"
- Submit the PDF as a zip or in a GitHub project repo using the project submission portal

If using GitHub:

- Create a new GitHub repo for the capstone. Name it "**Capstone Project**"
- Add this document to your repo. Make sure it's named "**Capstone_Stage1.pdf**"