GitHub Username: manvigupta1987

WalkMore

Description

WalkMore is a simple app to record and display: the number of steps, calories, distance, duration, and heart rate. A user can view the above information in a graph. WalkMore App supports English and Arabic languages. The app uses Google Fit API to record the number of steps. This App stores the information in a client side database.

Intended User

WalkMore automatically tracks your steps. Just keep your mobile phone in pocket when you walk.

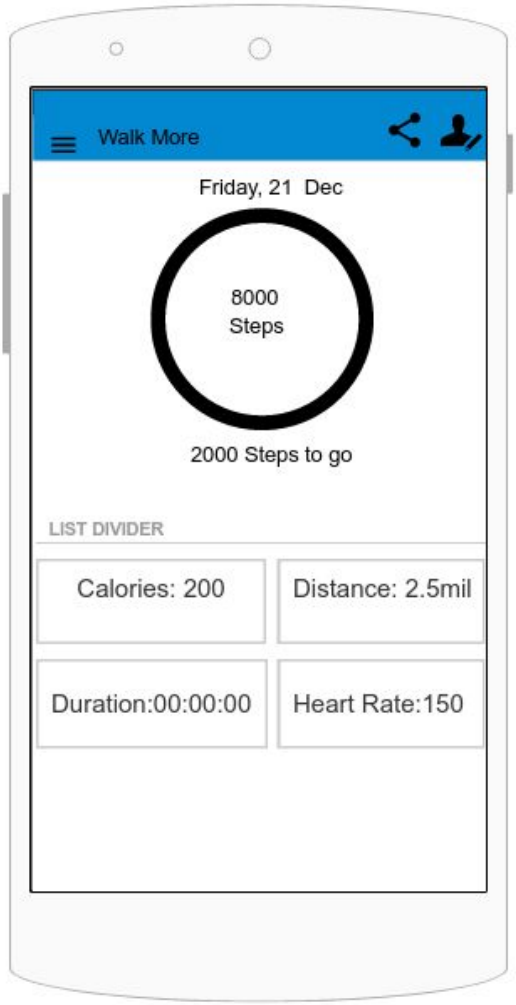
Features

- Displays steps, calories, distance, heart rate, and walking duration using Google-Fit API and stores to a local database
- User's current location is used to determine the steps walked.
- Displays a graph with step counts using MP Android library. The graph is created for weekly, monthly ,and yearly time resolutions. The app also shows the total calories burnt and the total distance covered.
- User can also set the daily goals.

User Interface Mocks

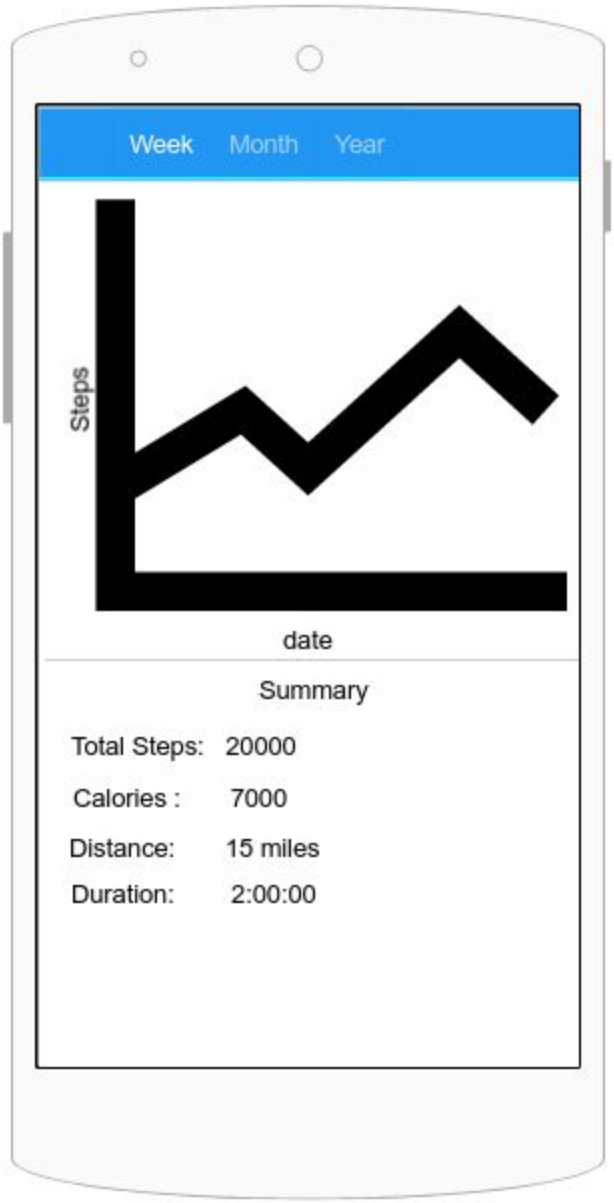
I used Proto.io to create the below mocks:

Screen 1- Main Screen



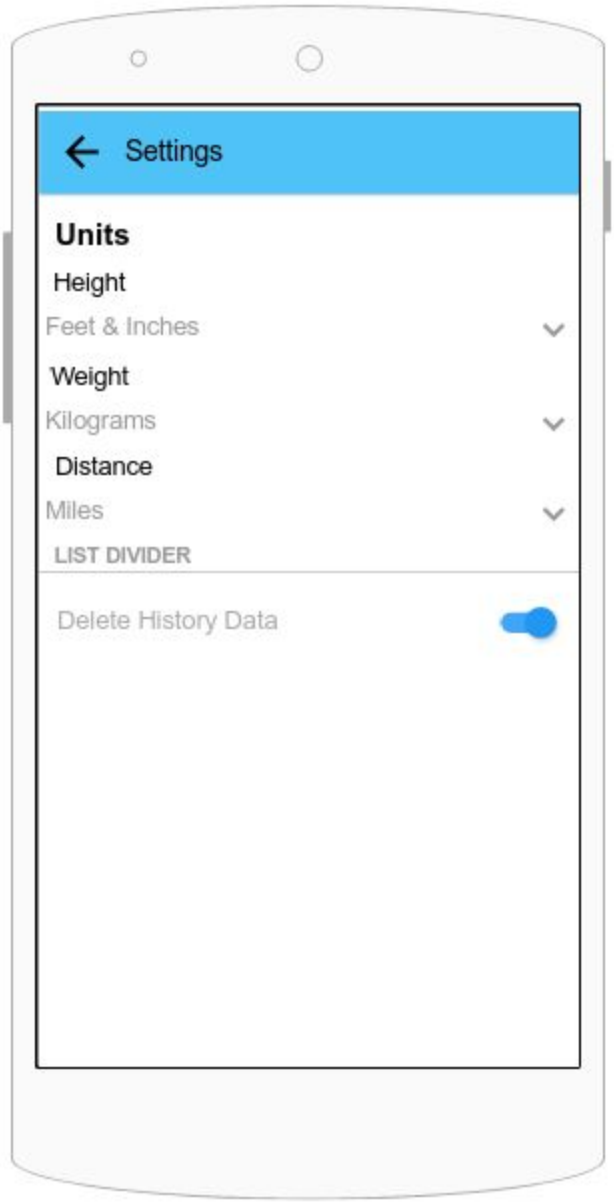
Displays number of steps completed and remaining steps in daily goals. It also shows the calories burnt, distance covered, duration, and heart rate in bpm. User can also share his/her achievement with his friends on facebook, twitter etc. Edit menu is available in the overflow menu to adjust weight, age and height of the user. The main screen also has a navigation drawer which helps in changing the app settings, goals, and displays the history data for a week, month, and year.

Screen 2- History Data



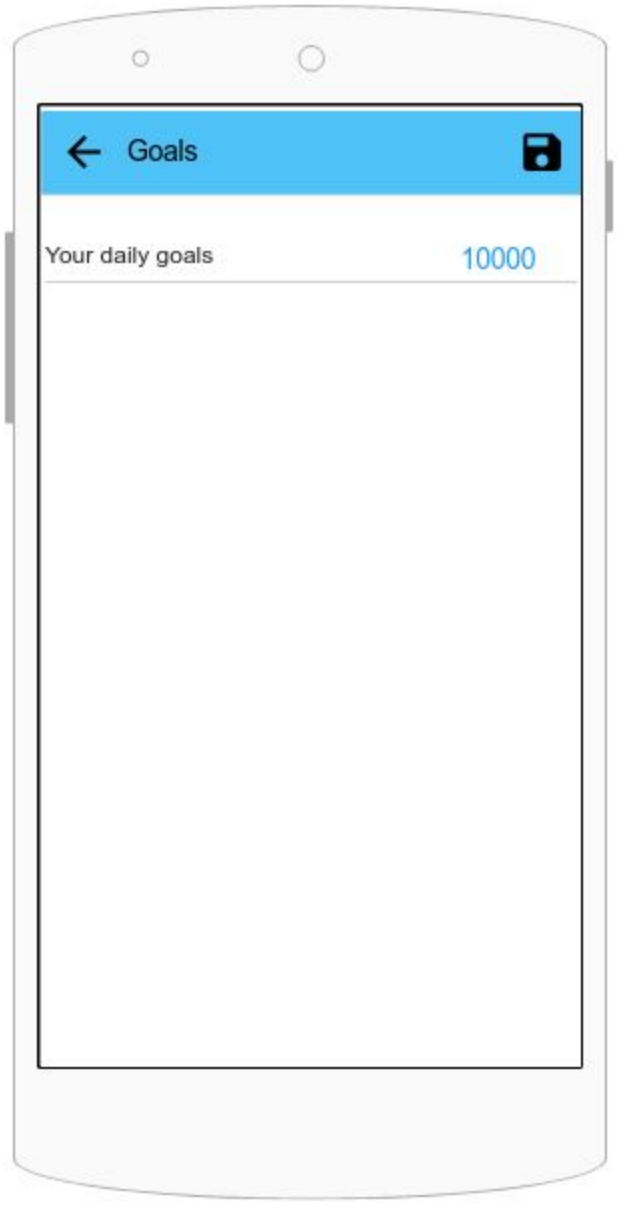
This screen has three separate tabs for week, month, and year. Each tab shows the summary of steps, calories burnt, duration and distance covered. It also shows step counts in a time axis graph.

Screen 3-Settings



This screen contains options to change the units of height, weight, distance. User can also delete history data.

Screen 4-Goals



This screen provides an option to change the daily goal limit. By default it is set to 10000 steps per day.

Key Considerations

How will your app handle data persistence?

App data will be fetched from Google-fit History API and stored in SQLite database. Sync will be handled by a sync adapter and content provider.

Describe any corner cases in the UX.

User needs to have google account. Location services must be enabled to use the app. The app cannot record steps otherwise.

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

- Google-fit API: for recording and displaying the steps, calories, distance, heart rate.
- Appcompat library: to ensure compatibility with multiple android versions
- Support design library: used for backward compatible material design
- Butterknife: for reducing view boilerplate code
- Timber: for logging.
- MP Android Library: to display the graph of step counts.
- Google play services: for location.
- Schematic: for content provider

Describe how you will implement Google Play Services.

This app uses Google-fit API for recording steps, calories, distance, and heart rate. The api is used as follows:

1. Install Google play services from SDK manager.
2. Create a project on Google API Console to enable the Fitness API and create new credentials to generate OAuth 2.0 Client ID.
3. Add the dependency in the build.gradle for google fit api; com.google.android.gms:play-services-fitness:9.8.0
4. Connect to the following fitness API of Google play services:
    a. Fitness.SENSORS_API
    b. Fitness.RECORDING_API
    c. Fitness.HISTORY_API
    d. Fitness.SESSIONS_API

Tasks

Task 1: Project Setup

1. Create a new Android project
2. Create the following activities:
    a. MainActivity
    b. HistoryDataActivity
    c. GoalsActivity
    d. SettingsActivity
3. Include the following libraries:
    a. compile 'com.google.android.gms:play-services-fitness:9.8.0'
    b. compile 'com.android.support:appcompat-v7:25.2.0'
    c. compile 'com.android.support:design:25.2.0'
    d. compile 'com.jakewharton:butterknife:8.5.1'
    e. compile ‘com.jakewharton.timber:timber:4.5.1’
    f. compile 'com.github.PhilJay:MPAndroidChart:v3.0.2'
    g. compile 'com.google.android.gms:play-services:10.0+'
    h. compile 'net.simonvt.schematic:schematic:0.7.1'

Task 2: Implement UI for Each Activity and Fragment

- Build UI for Main Activity
- Build UI for History Data Activity
- Build UI for Settings Activity
- Build UI for Goals Activity

Task 3: Create Main Activity

This activity shows the daily steps, calories burnt, distance, walking duration, and heart rate.

- Create a progress circle which shows the current step count.
- Create a navigation drawer with the menu items; history data, settings and goal settings.
- Create a overflow menu Edit; to change the user’s age, gender, weight and height.
- Create a share option which helps in sharing the user’s achievement on facebook, twitter, gmail etc.

Task 4: Implement Google Fit API

- Configure the project to use google play services using the steps mentioned above.
- Connect to the google-FIT API using GoogleApiClient. Implement the functions related to GoogleApiClient call backs.
- Once GoogleApiClient is connected to Google, access step data using the Google-Fit Sensor Api.

Task 5: Implement Content Provider and Sync Adapter

- Use Google fit Recording API and History API to fetch the steps, calories burnt, distance and heart rate on a regular basis.
- Implement a database to store above data retrieved from Google-Fit API. Create a database helper and a database contract to ensure data is written into database correctly. Create URIs for querying the database. Data can be fetched for week, month and a year.
- Implement a content provider and sync adapter to keep the database up to date. Content provider needs a stub only, since this app doesn't share data with other apps. Implement sync adapter to fetch data from web API and store in the database via content provider

Task 6: Create History Data activity

This activity shows the user's fitness data for week, month, and year. This activity is launched from the menu present in the navigation drawer.

- Use a loader to query the database created by the sync adapter to return fitness data based.
- Implement a chart using MP Android Library with step count and date.
- Display the total steps, calories, distance and duration for a week, month, and year using the textView.

Task 7: Create Settings activity

This activity allows user to choose the units for the height, weight, and distance. It also provides an option to delete the history data.

- Implement a settings activity with drop down menu to set the desired units for distance, weight, and height.
- Implement a switch to allow user to delete the history data.

Task 8: Create Goal Settings activity

This activity allows user to set the daily goal.

- Implement an edit text which allows user to edit the value of daily goal. By default this value is set to 10000 steps per day.