

GitHub Username: [aditya-bhawsar](#)

# Walk Tracker

## Description

This App will make it easier for user to track their daily walk time from one destination to another in form of storing daily walks virtually and displaying their details of walk such as Date, Distance, Calories burned and much more all at some taps distance

## Intended User

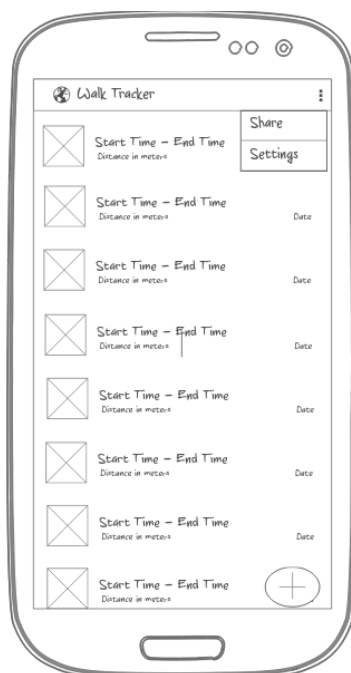
For people who would like to track their daily walk and calorie burn with just some taps

## Features

- Saves your daily walk history virtually at a simple cost of some taps
- Allows inviting friends using dynamic links
- Tracks App crashes using Crashlytics
- Receives a notification from firebase console
- Home Screen widget that shows last walk by the user

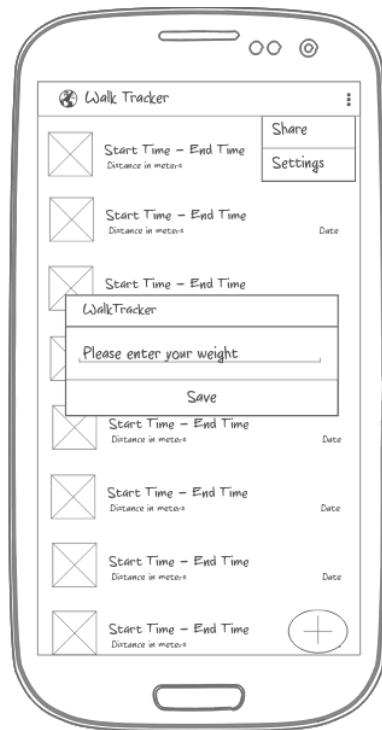
## User Interface Mocks

### Screen 1



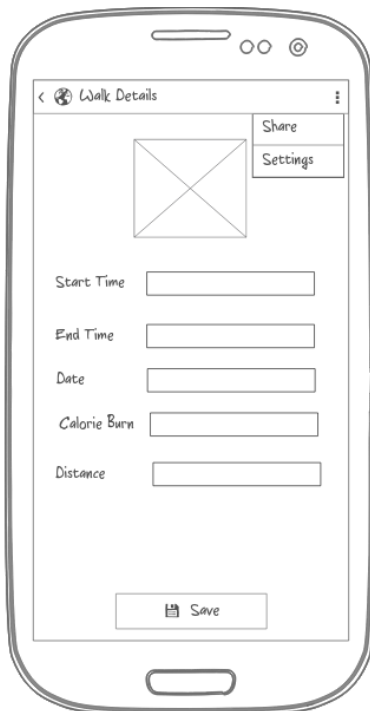
Main Activity Screen Which Displays list of all previous walks and a Add button along with settings and share menu

## Screen 2



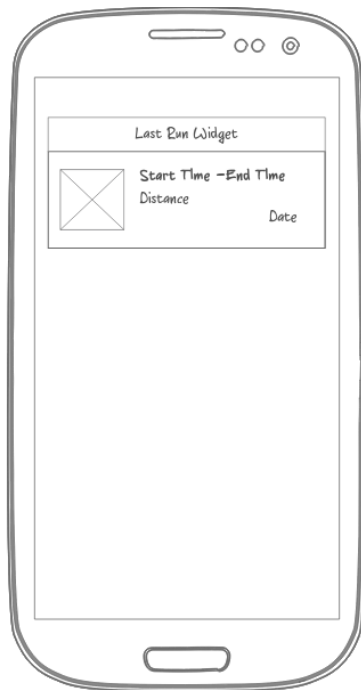
A dialog box will be shown which is required to get user weight later used to calculate burned calories

## Screen 3



This is detail screen that can be seen for each walk and take input of distance at the end of walk

## Screen 4



This widget represent the widget on the home screen which shows the last recorded walk

## Key Considerations

**How will your app handle data persistence?**

This App will user Room to store the data of all the walks in the app locally and will use a Weight input from user which will be stored in the apps Shared Preferences

**Describe any edge or corner cases in the UX.**

Pressing back in any activity returns you to the last activity and if no other activity is in the stack then to the home screen of mobile device

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

Glide to handle the image loading from resources

Jetpack Libraries that are required to implement Room, ViewModel and LiveData

Firebase Libraries to implement crash analysis and cloud messaging

Design libraries from support

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

Will use firebase to Get Crash reports using Crash Analytics and Dynamic Links to help user to invite friends

## 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

Adding libraries to the project and configuring them and firebase project

- Adding Libraries
- Creating Firebase Project and adding the json file

### Task 2: Implement UI for Each Activity and Fragment

Create Ui for all the items in the app such as Widget, Settings and Other Screens

- Build UI for MainActivity, Details Activity and Settings Activity
- Build UI for RecyclerView
- Adding View Model and LiveData Along With Room
- Create UI for widget of last run

### Task 3: Create App Backend by adding Database Using Room

Will Setup Database in project

- Create Dao and Classes to Make Database
- Create Repository in project
- Integrate the UI and Database

### Task 4: Setting Up Sharing and Settings

Add Dynamic links to make sure the app user can invite their friends

- Add Calculation and UI show logics to the app and calculation logics such as “Calorie Burn”
- Add a share message for the messaging app
- Add cloud messaging to show notification