How to Use this Template

1. Make a copy [File → Make a copy...]
2. Rename this file: "**Capstone_Stage1**"
3. Replace the text in green

Submission Instructions

1. After you've completed all the sections, download this document as a PDF [File → Download as PDF]
2. Create a new GitHub repo for the capstone. Name it "**Capstone Project**"
3. Add this document to your repo. Make sure it's named "**Capstone_Stage1.pdf**"

GitHub Username: kishorandroid

DaiET - Daily Expense Tracker

Description

DiaET is a daily expense tracker where users can track their everyday expenses and generate reports to analyse their spending habits/patterns to better manage their monthly income.

Intended User

This app is intended for salaried/self-employed users in India.

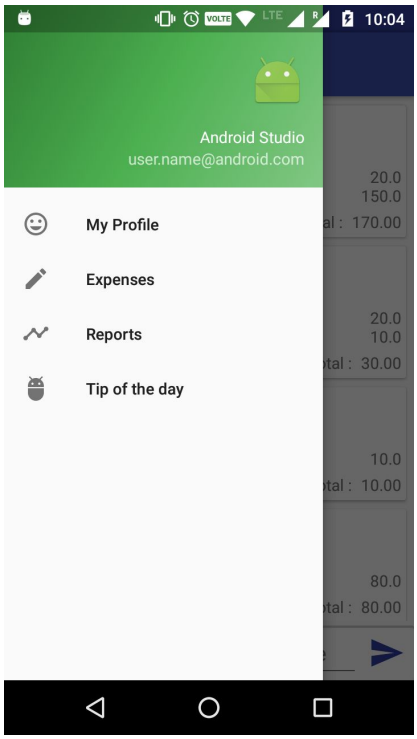
Features

- Save user's daily expenses
- User can set monthly savings target based on monthly income (e.g. 35% of monthly income)
- Show how much can be spent in a day based on savings target
- Keep user informed whether savings target is achievable.
- Generate reports based on specified criteria (e.g Weekly/Monthly report, Categorised report)
- Tip of the day (Personal Finance tips)

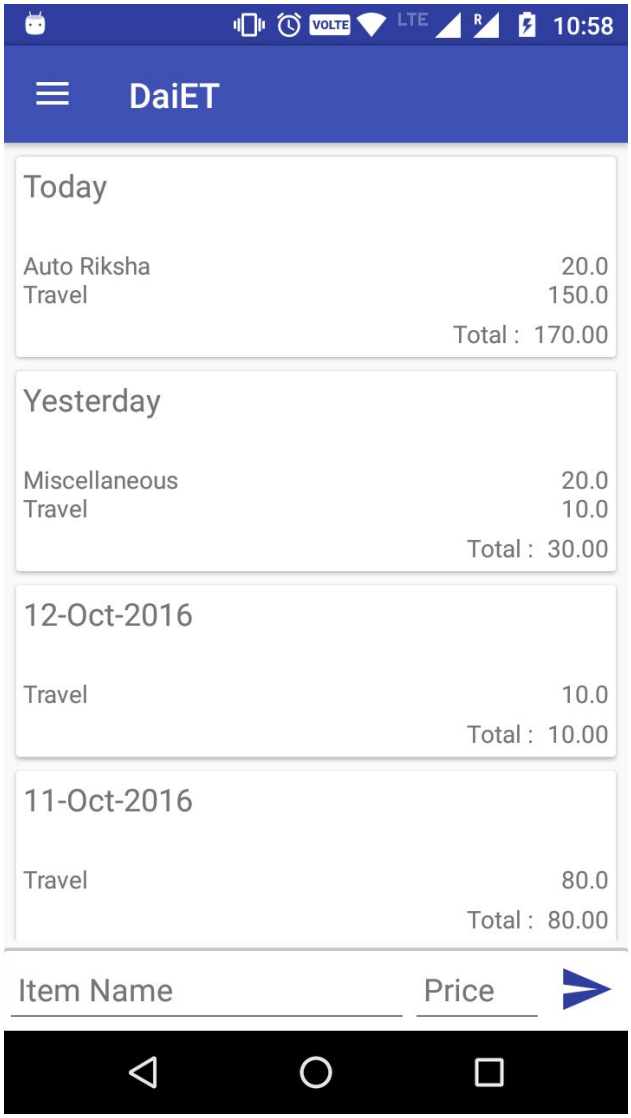
User Interface Mocks

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

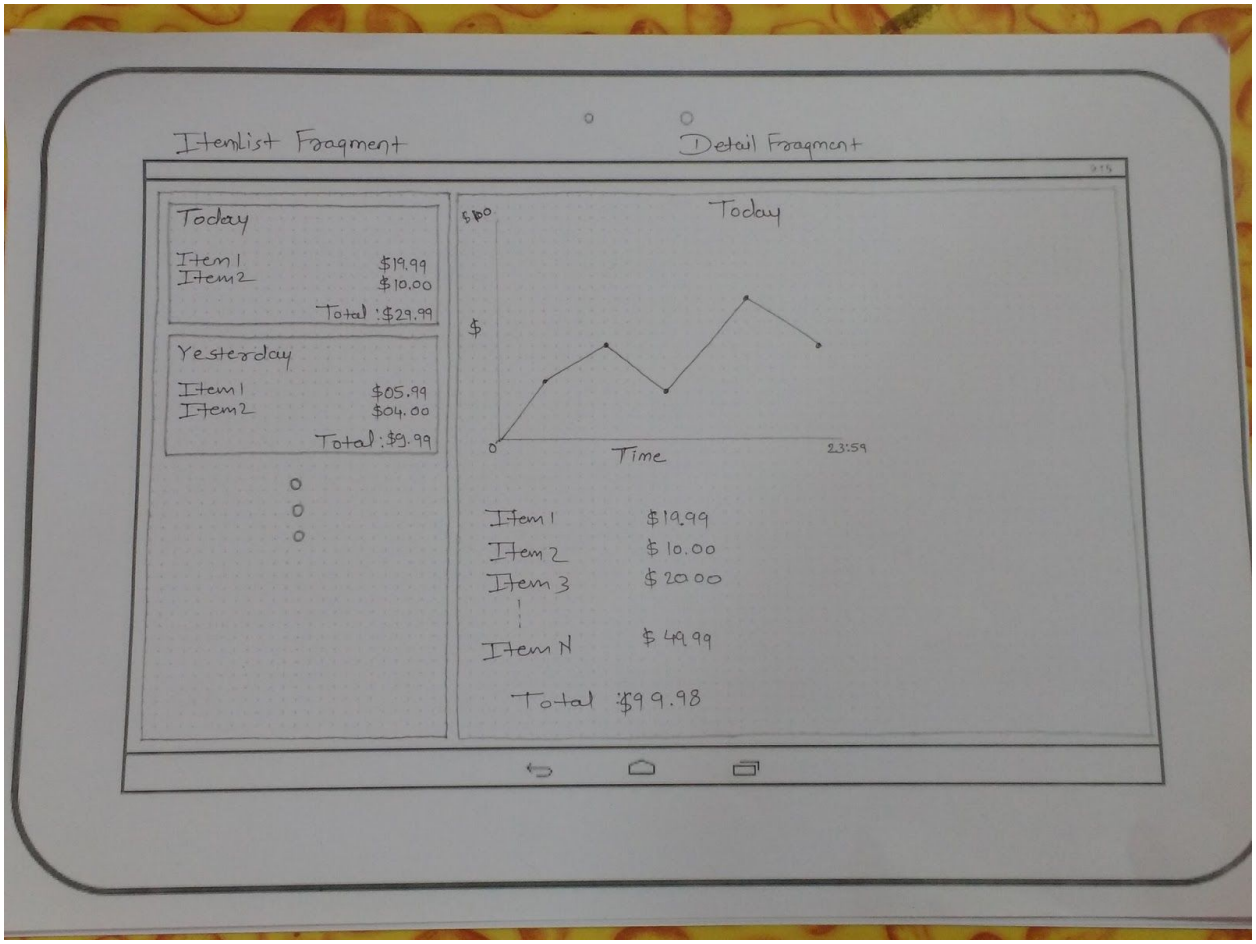
Screen 1 - Drawer Menu



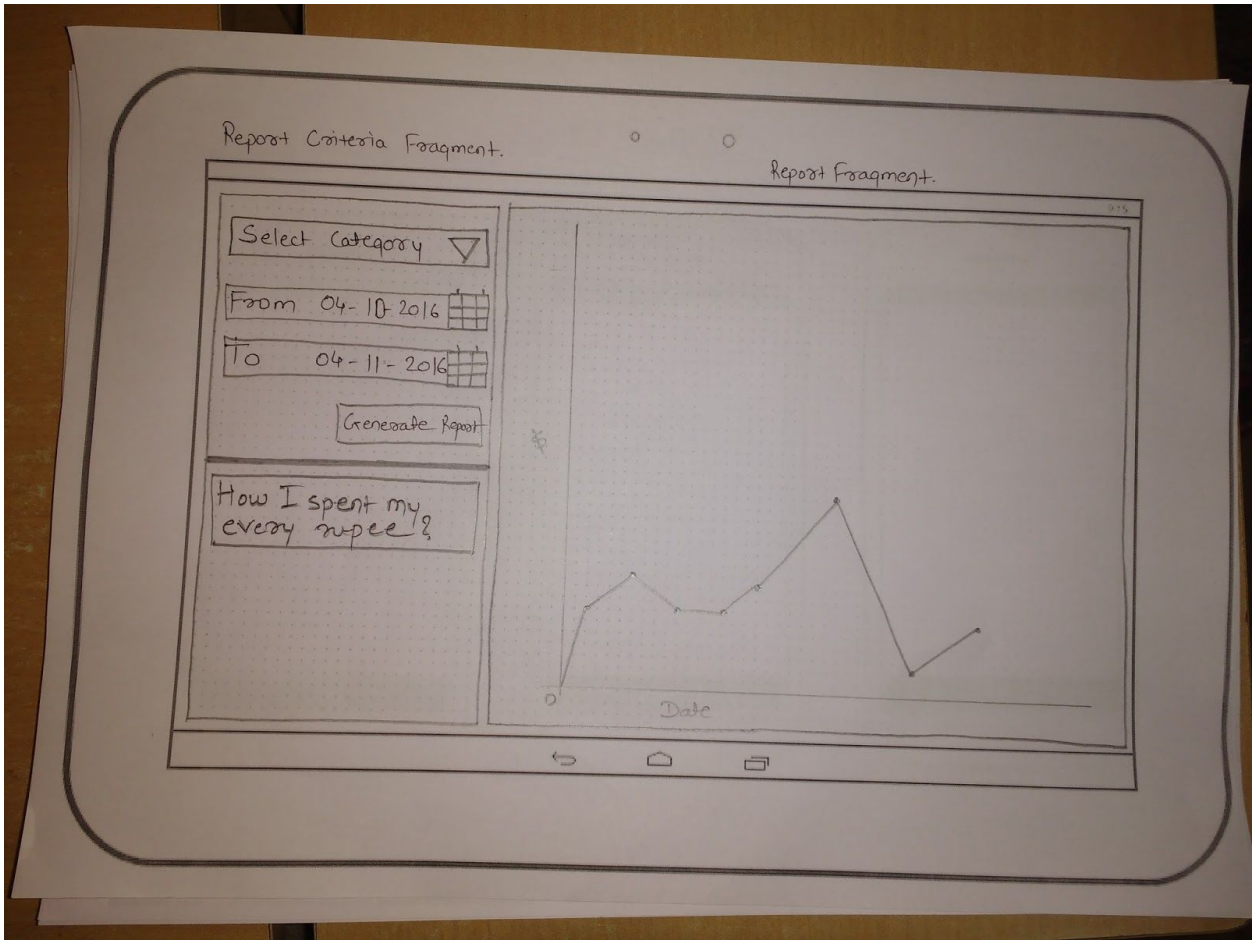
Screen 2 - Expenses List



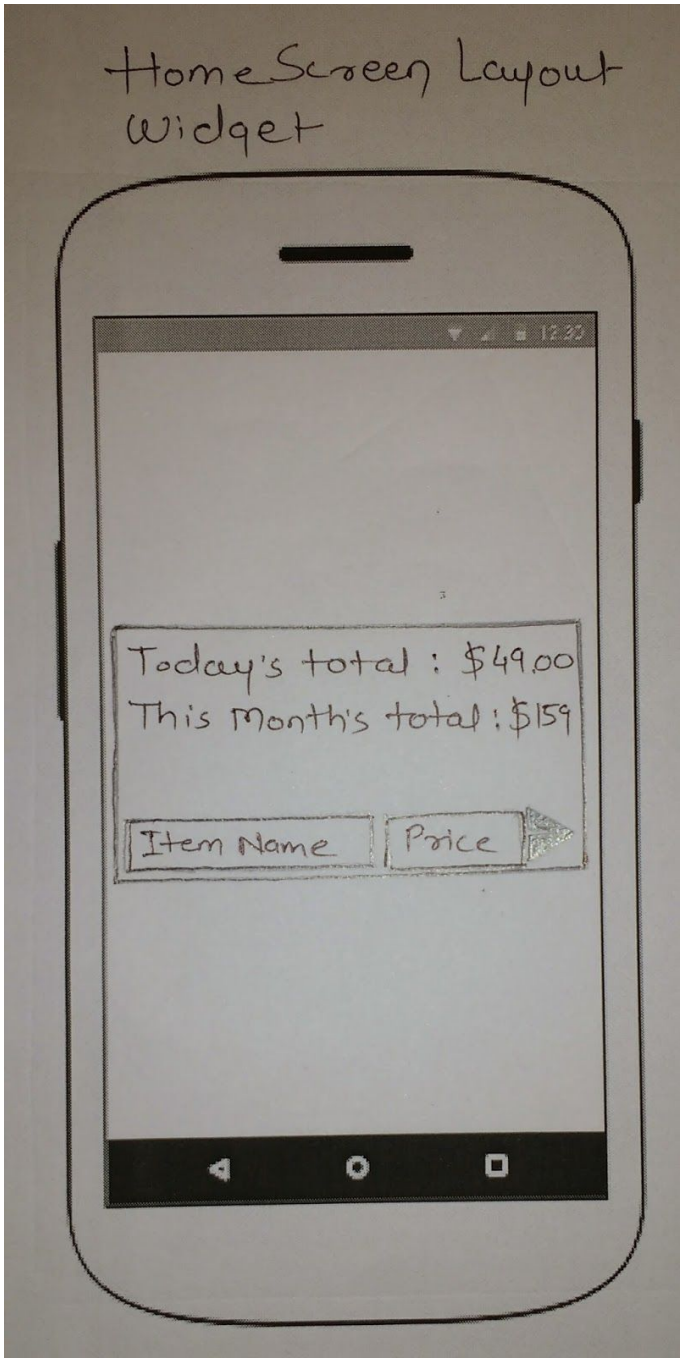
Screen 3 - Tablet Layout for Expenses List



Screen 4 - Tablet Layout for Reports Screen



Screen 5 - Layout for Home Screen Widget



Key Considerations

How will your app handle data persistence?

App will use SQLite database and content provider for data persistence.

Describe any corner cases in the UX.

For example, how does the user return to a Now Playing screen in a media player if they hit the back button?

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

App will use MPAndroidChart library for displaying charts in reports

Describe how you will implement Google Play Services.

- Google play services "com.google.android.gms:play-services-analytics:9.6.1" for analytics
- Google play services for user login

Next Steps: Required Tasks

Task 1: Project Setup

- Setup development environment
- Brainstorm on App navigation
- Design app structure
- Create a Project with empty activities

Task 2: Implement UI for Each Activity and Fragment

- Build UI for Expenses view activity with master detail UI
    - On user data submit validate, store and update "Daily Expenses" screen
    - Write test for testing validations and database operations

- Build UI for My Profile activity
    - My Profile - Monthly Income, Total Fix Monthly Expenses (Rents + Insurances + Utility bills + EMIs),
    - Savings Target (Recommended 40% of Monthly income)

- Build UI for Reports activity with master detail UI
    - Report criterion are Month selection, Category selection, Every $ spent
    - Update detail UI (display chart) on criteria selection

- Build UI for Tip-Of-The-Day activity

Task 3: Design and Create Database structure

- Design database
- Create database structure and add related classes into project

Task 4: Develop “Tip Of The Day” provider cloud connected library module

- Develop "Tip Of The Day" provider library
- This library fetches "Tip Of The Day" related to personal finance from cloud backend
- Write test for testing library features

Task 5: Add Google account login feature

- Add "Google Sign In" feature for backing up data, for users who wants to change device and restore content

Task 6: Implement Home Screen widget

- Design and implement home screen widget for adding expenses
- This widget will be used by user to quickly insert expense record into database from home screen directly.
- This widget will also show “Tip-of-the-day” text message that will be pulled from web service at regular intervals (every one hour), app will use IntentService to pull “Tip-of-the-day” and Broadcasts to update widget.

Submission Instructions

1. After you’ve completed all the sections, download this document as a PDF [File → Download as PDF]

2. Create a new GitHub repo for the capstone. Name it “**Capstone Project**”
3. Add this document to your repo. Make sure it’s named “**Capstone_Stage1.pdf**”