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: splxsg

Money Saver

Description

This is an App for personal financial management including daily maintenance, utility, insurance, loan, income and so on. For daily maintenance, this app will be an easy way to record each penny cost by manually type in over the widget. For utility, insurance, app will be able to set a pay out monthly with a fixed cost, also it would be able to modify exact cost when the bill is

generated later. (In UK, we pay the electric bill by a fixed price every month, it based on an estimate reading from the meter. Staff may check the exact reading seasonally, then generate an exact bill for last three month). A fixed salary with modified pay in day can be set as well.

Intended User

University student, housewife, employer, retire, everyone who wants to monitor where his money goes to.

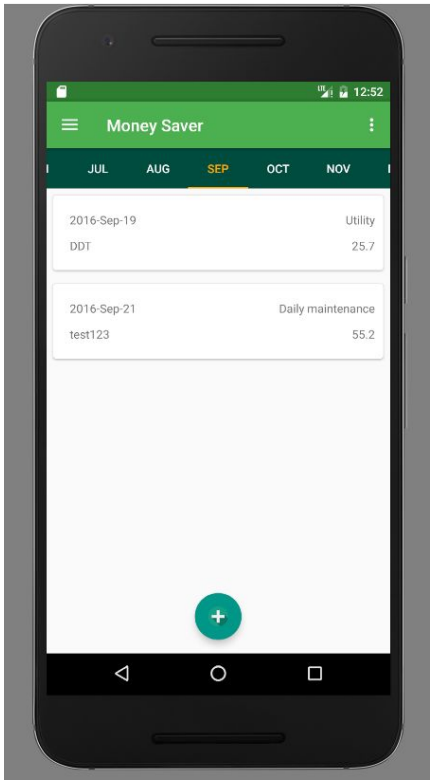
Features

Store information
Takes pictures(for receipt)
*OCR(in the future)

User Interface Mocks

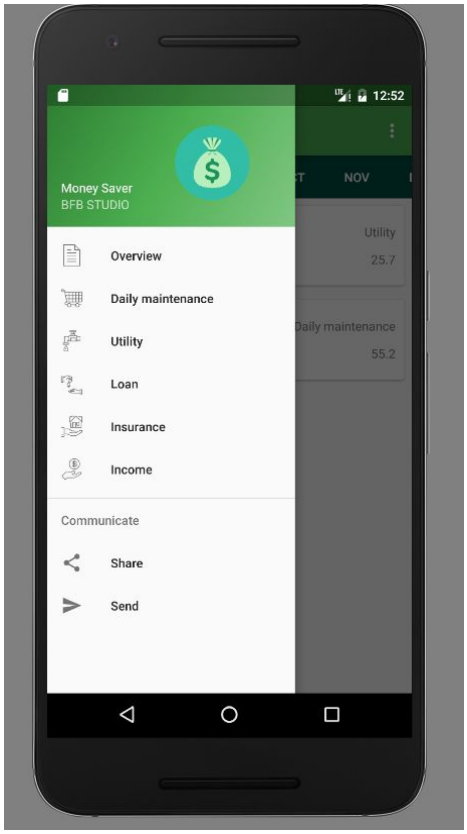
These can be created by hand (take a photo of your drawings and insert them in this flow), or using a program like Photoshop or Balsamiq.

Screen 1



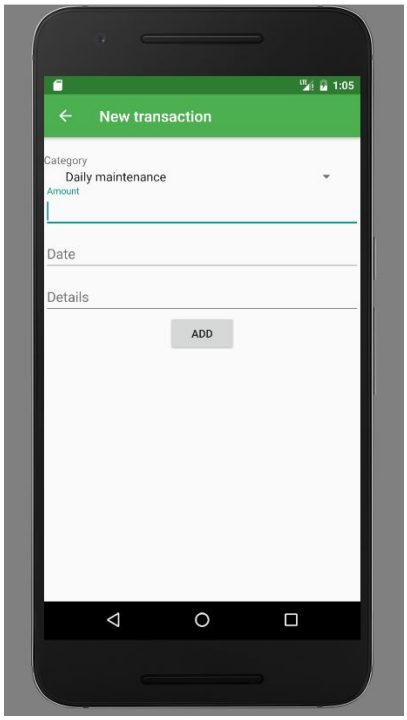
This is the screenshot for display two transactions on Sep. One is for utility payout one is for daily maintenance.

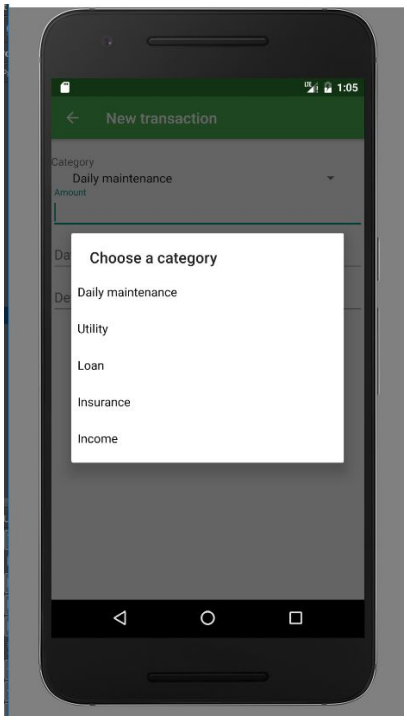
Screen 2



This is the screenshot for drawer menu which can choose different category to view the transactions.

Screen 3





Page to add new transaction, user can choose from 5 different category, and choose date from a date picker.

Key Considerations

How will your app handle data persistence?

All the transactions will be stored in content provider. An update flag will be set to true when there is a new transaction added. The current month report will be updated if the flag is true. The monthly summary report for past month will be generated and stored in content provider as well, in order to make the app faster.

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.

SQLite, maybe library related to OCR in the future.

Describe how you will implement Google Play Services.

Describe which Google Play Services you will use and how.

Next Steps: Required Tasks

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

Task 1: Project Setup

1. Setup drawer menu
2. Tap view and pageview
3. Generate fragment under different tab automatically
4. Obtain tab index in order to unique the fragment content based on different tab
5. Create a new activity to add new transaction into sql
6. Try to display sqlite content on the fragment
7. Display content on recycleview in fragment
8. Display the sqlite content on recycleview
9. Display the content based on different tab(month)

Task 2: Implement UI for Each Activity and Fragment

List the subtasks. For example:

- Build UI for MainActivity
- Build UI for something else

Task 3: Your Next Task

As the app has basically run well: the next task is to

1. Able to click each transaction, to view the details or modify amount or something
2. User defined category (dynamically add item on drawer menu and category spin in ADD page)
3. View different category when choose it on the menu
4. The overview will be able to see total balance now
5. Not sure if i can , but will try to generate a diagram (like pie type) to analysis payout by different category
6. Able to save the photo of receipt, it will be able to see on the detail page.

Add as many tasks as you need to complete your app.

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**"