

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**”
-

[Description](#)

[Intended User](#)

[Features](#)

[User Interface Mocks](#)

[Screen 1](#)

[Screen 2](#)

[Key Considerations](#)

[How will your app handle data persistence?](#)

[Describe any corner cases in the UX.](#)

[Describe any libraries you’ll be using and share your reasoning for including them.](#)

[Describe how you will implement Google Play Services.](#)

[Next Steps: Required Tasks](#)

[Task 1: Project Setup](#)

[Task 2: Implement UI for Each Activity and Fragment](#)

[Task 3: Your Next Task](#)

[Task 4: Your Next Task](#)

[Task 5: Your Next Task](#)

GitHub Username: **gj100596**

Hisab

Description

Forgot how much money your friends owe to you?? Don’t worry, Hisab will let you manage that? Hisab is an android app which allows user to maintain their personal account with their friends. It will help you to manage how much money you lend to you friend and also give the option to remind them about it [In case if they are avoiding it ;)]

Intended User

For people who travel or live in group and have to take/give money from/to their friends.

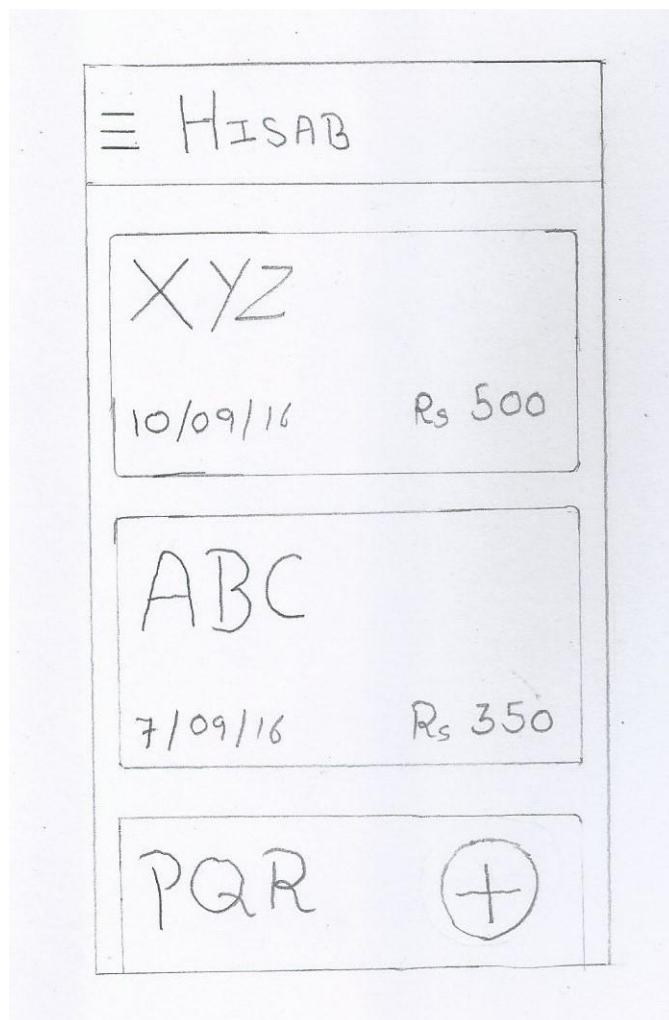
Features

- Record Transactions
- Send Reminder to friends
- Save previous transaction

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



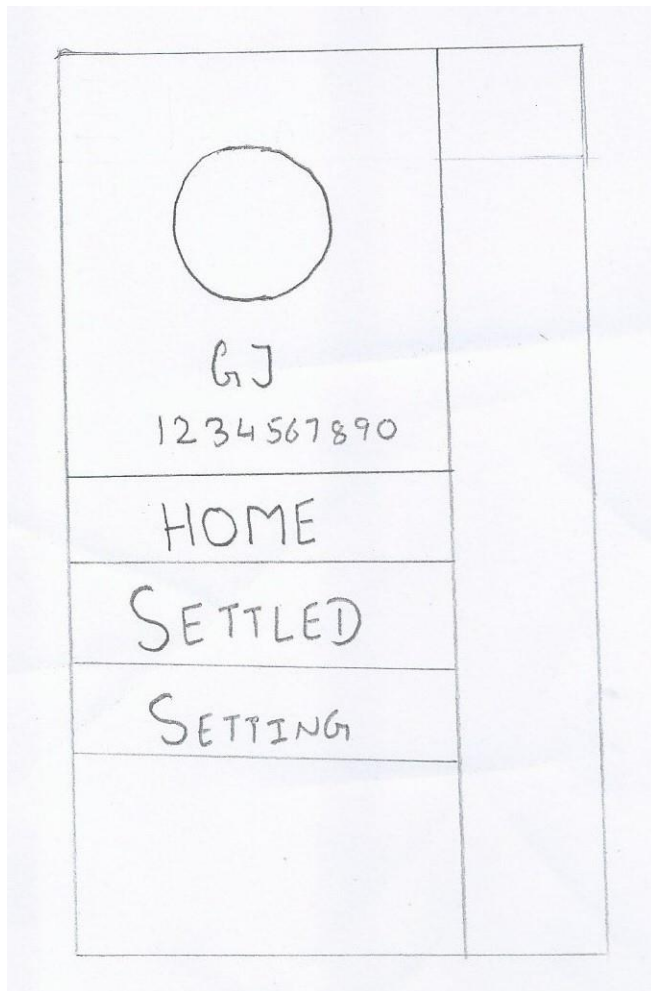
The Main Screen (Home). It will have Cards inflated in a recycler view. The Cards will be showing name of each person to whom I have any transaction. Each card will show the name of friend, the last date of its transaction update and the total amount of money borrowed and not yet returned.

Screen 2

| | |
|-----------------------|--------|
| ≡ HISAB | |
| XYZ | Rs 500 |
| Reason | Amount |
| Breakfast 20/08/16 | 150 |
| Pizza 14/08/16 | 300 |
| Misc 10/08/16 | 50 |
| | |

Detail View for each friend. This layout will be use for 2 fragments:

- 1) When user click the friends card in main screen, this layout will be used by current unsettled transaction fragment to display all the pending transaction with that friend.
- 2) For the Settled fragment, this layout will be inflated with previously settled transactions.



Navigation Drawer. It shows one pic of the user(Probably thinking to take it from contact but not final, will depend on review from test users). After DP there will be the name and mobile no. After that there will be options for going to different activities.

Key Considerations

How will your app handle data persistence?

Will Store the transaction into SQLite DB and use Content Provider for fetching it. Loader will be used to provide data to the view.

Describe any corner cases in the UX.

Most of all movements within the apps will be straight forward only.

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

Volley: To interact with server for registering the user no, sending reminder to other person etc.

Picasso: To load the DP in navigation drawer.

Describe how you will implement Google Play Services.

Cloud: Will Use GCM to send reminder to people about their debits.

Ads: Ads will be used but in limited amount that does not hinder user experience.

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

- Update Android Studio and SDK to latest versions.
- Set up the Git repo to work with.
- Integrate volley module into project.

Task 2: Implement UI for Each Activity and Fragment

List the subtasks. For example:

- Build Layout for Card View which will be used in ListView
- Build UI for MainScreen
- Build UI for Detail Screen
- Build Layout for Navigation Drawer.

Task 3: Create Server Endpoints

- Develop the endpoints

- Deploy them on EC2
- Test Them.

Task 4: Integrate GCM using endpoints.

- Use Volley to connect to the endpoints.
- And test the GCM functionality.

Task 5: Test all the components.

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**"