
[Description](#)

[Intended User](#)

[Features](#)

[App Flow](#)

[App Module](#)

[Add Categories](#)

[Recurring expense](#)

[History/Report](#)

[History Tab](#)

[Reports Tab](#)

[Account](#)

[Recurring Account](#)

[Non-Recurring Account](#)

[Notification](#)

[Settings](#)

[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.](#)

[Required Tasks](#)

[Task 1: Project Setup](#)

[Task 2: Make Database](#)

[Task 3: Implement Dashboard Feature](#)

[Task 4: Account](#)

[Add Account](#)

[Display Account](#)

[Task 5: Expenses](#)

[Add Expense](#)

[Display Recurring Expense](#)

[Task 6: History/Report](#)

[Task 7: Settings](#)

[Task 8: Show Notification](#)

[Task 9: Expense Categories](#)

[Task 10: Orientation Change Handle](#)

GitHub Username: [rajesh-khadka](#)

Capstone Project : [github](#)

Expense Tracker

Description

People generally forget where they have spent their salary. Even they don't know to effectively spend money to fulfill their desire.

Expense tracker tracks expenses made by user. It shows the history and report based on criteria they have spent money. From this app user will be aware of proper utilization of money.

Intended User

Employed people, should be common people as common people tends to spend money more and they need to keep track of money spent.

Features

List the main features of your app. For example:

- Keeps expense record
- Shows daily expenses
- Shows history and report
- Can change recurring accounts/expense
- Notifies while adding recurring amount and expenses

App Flow

[Url](#)

If there is no account this screen will appear. After pressing add account it will navigate to add account screen

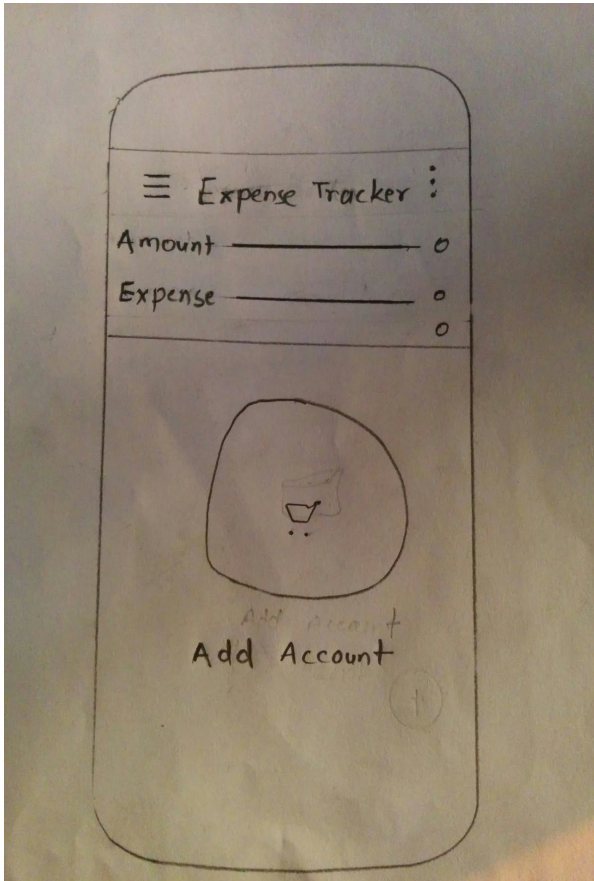


Fig 1. Dashboard

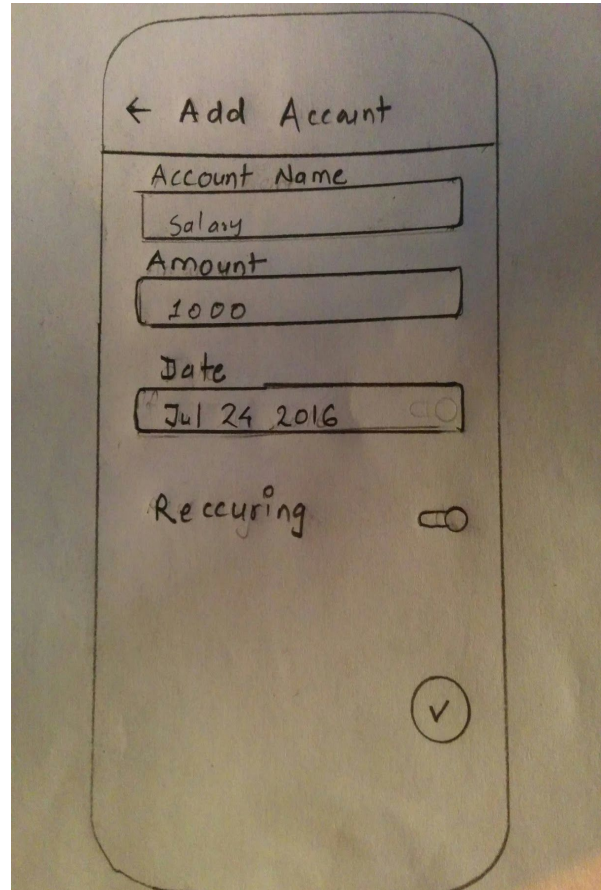


Fig 2 showing add account

In fig 2 we can specify the description of account. If the account is recurring it will be added to total amount every month at same date specified in date field. if it is not recurring it will be considered for one month only.

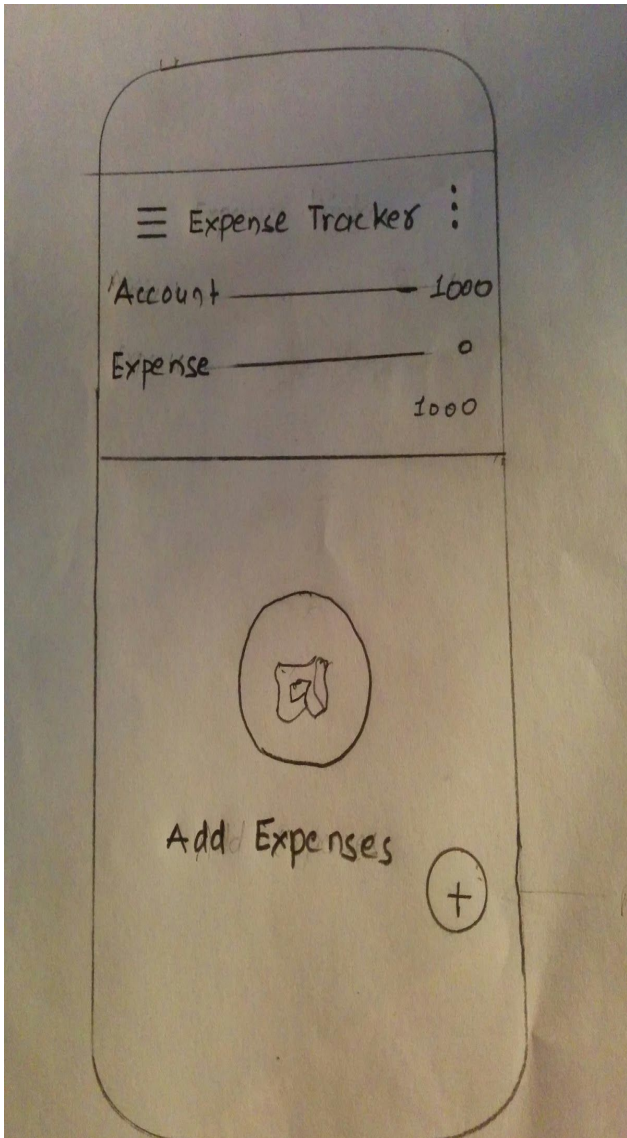


Fig dashboard with No expenses

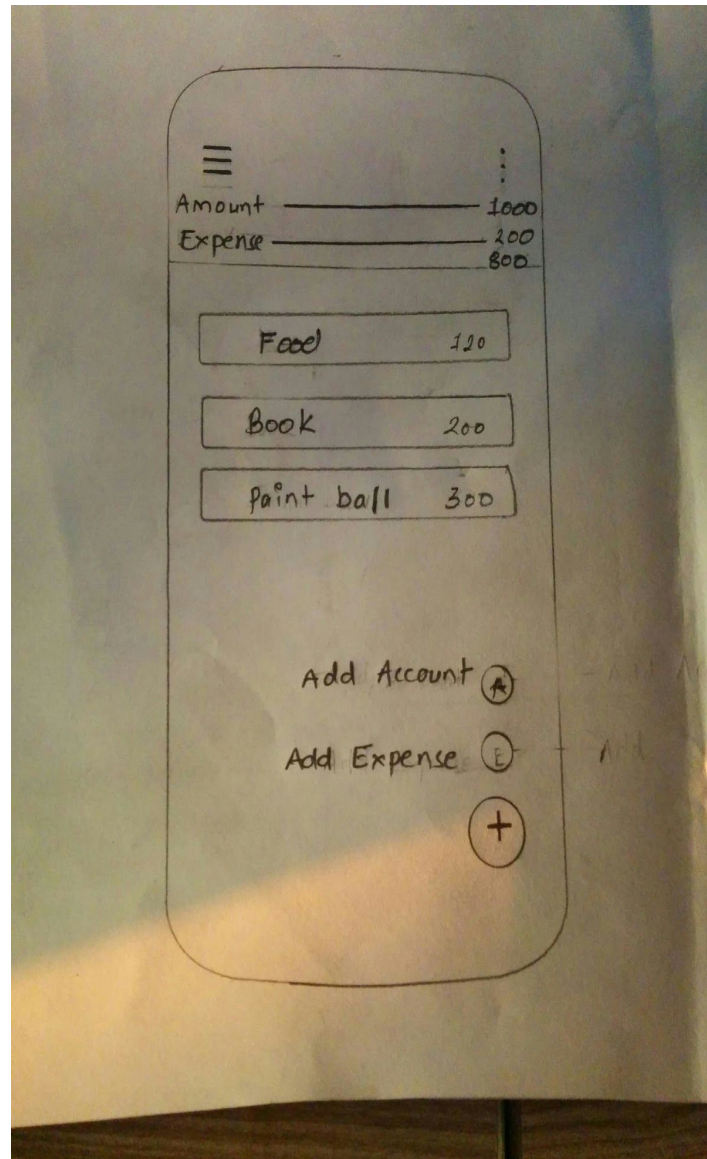


Fig Expanded FAB

After pressing floating action button we can get two sub floating action button for adding account and expense.

← Add Expenses

Expense Date
1st Aug 2016

Expense Title
Gym at ABC club

Description
Gym for fitness

Amount
1000

Category
☒ Food

Recurring ☐

✓

Fig Screen For adding expense

We can add detail description of expenses on this screen. If expense is the recurring it will be added at same date in coming month.

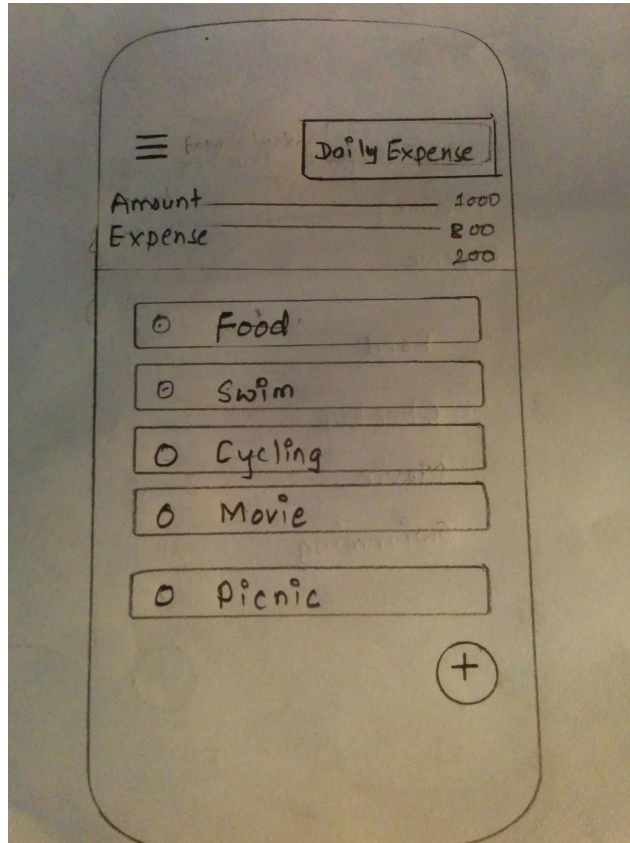


Fig Dashboard with overflow menu shown

User will be able to see daily expense by pressing on overflow menu as shown above. User can delete expense by swiping left to right and edit by long press on particular expense in dashboard.

App Module

App consists following feature

Navigation drawer consist these options:

- Add Categories
- Recurring expense
- History/Report
- Account
- Settings

Add Categories

It consist different categories such as :

- Food
- Entertainment
- Health
- House
- Pets
- Sports
- Taxi
- Toiletry
- Transport
- Communications
- Cloths
- Electricity
- Internet

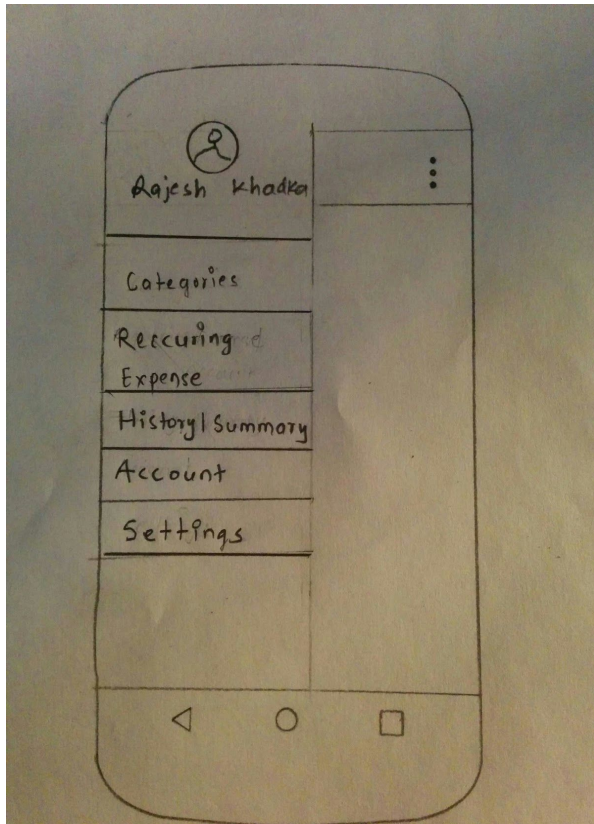


Fig Navigation Drawer Opened

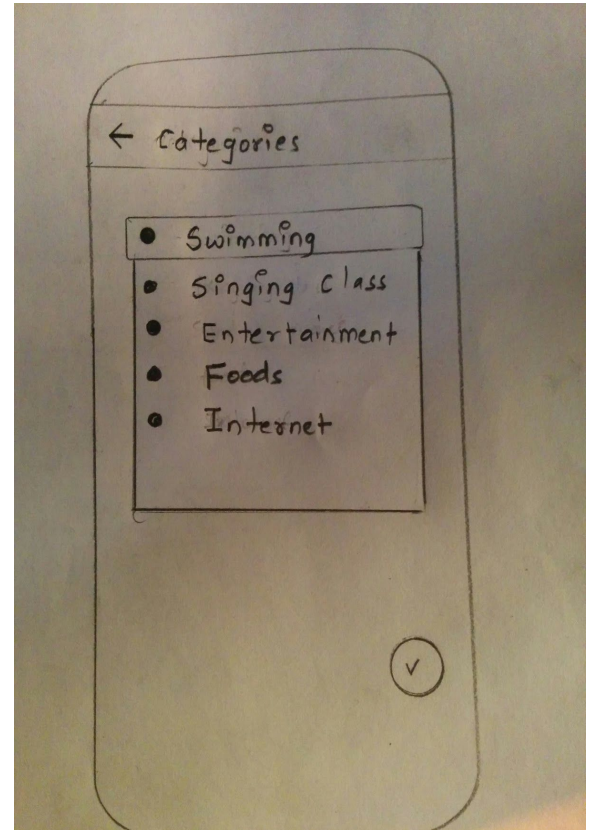


Fig Displaying Categories

Our app will consist default fundamental categories at the time of application installation. User can also customize according to their needs.

Black circle color will be different color according to categories. User can change color of black circle according to their need as:

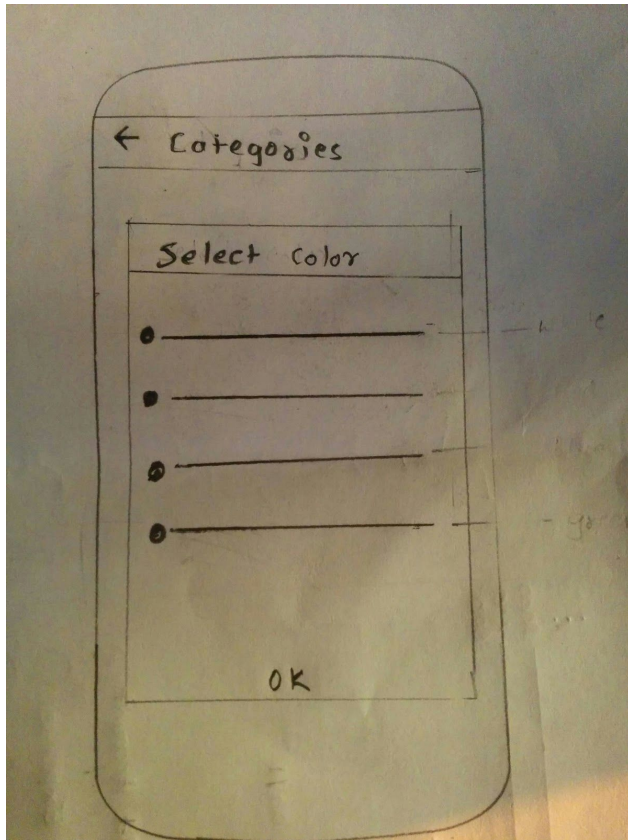
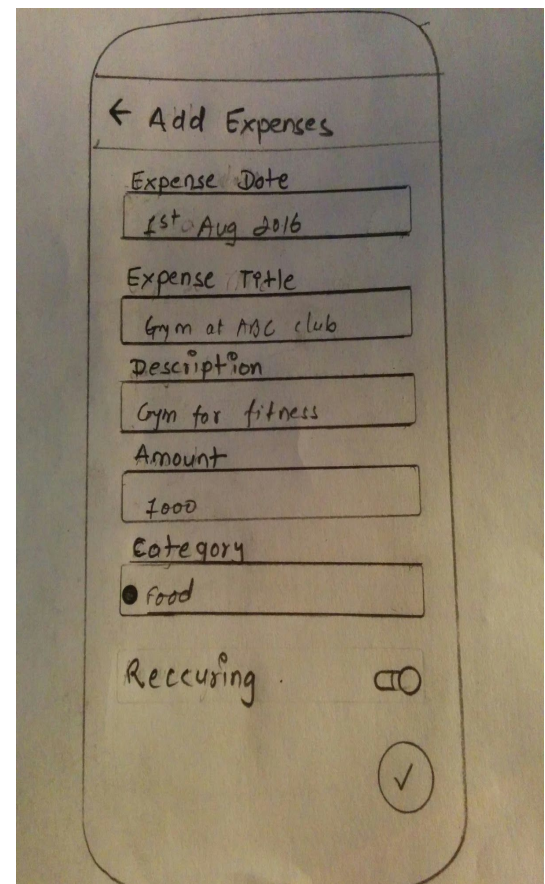
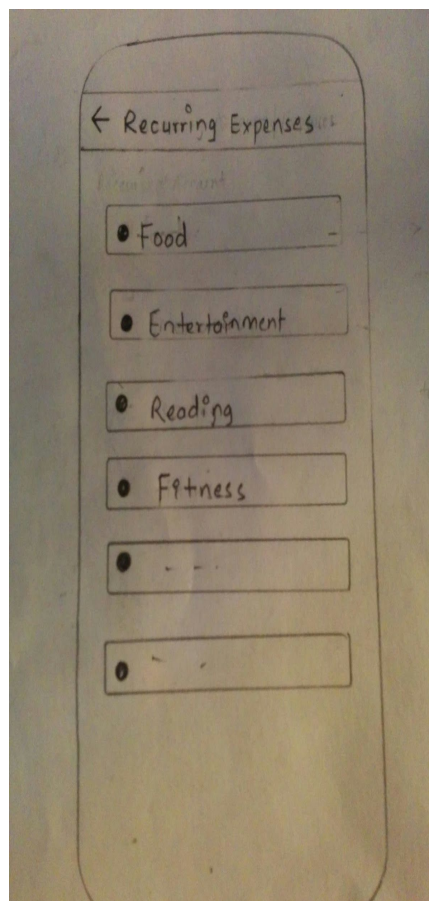
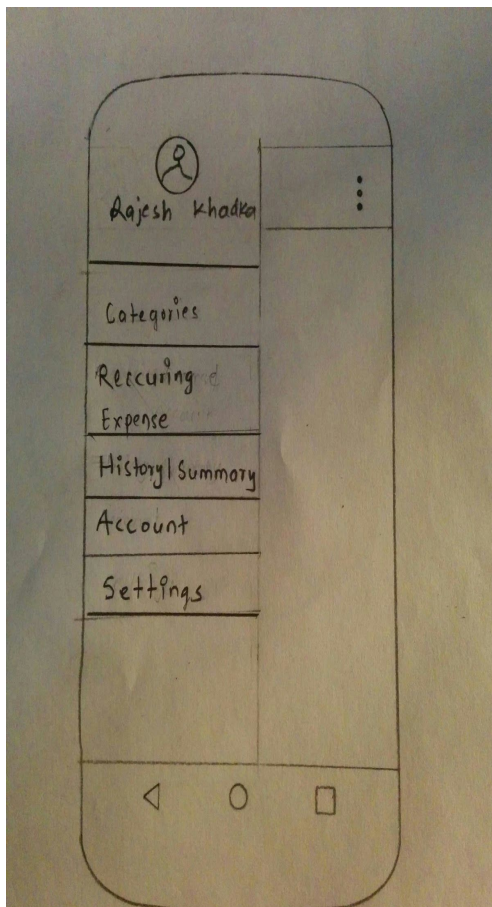


Fig Color picker Fragment Dialog for Categories

Recurring expense

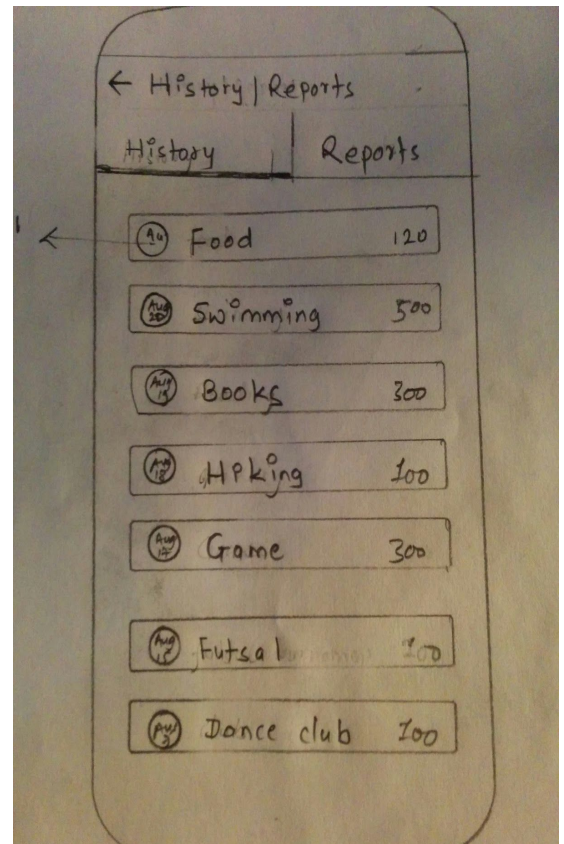
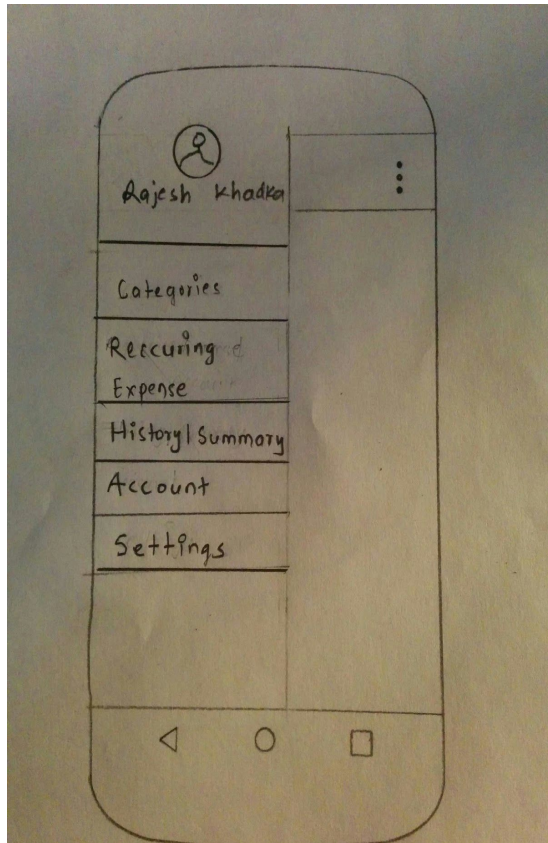
Most of expenses recur in periodic interval such as food, rent, gym electricity bill, telephone bill etc. These recurring expenses can added by user according to their need. These expense can be added or removed. **Long press** to edit and **swipe left to right** to remove recurring expense. Edit feature will be helpful eg: when schedule of paying electricity changes



History/Report

It consists two sliding tab of **history** and **report**. History tab shows all expenses in sorted order by date.

History Tab



Reports Tab

It shows expense in pie chart. User can see the expenses by categories in percentage. User can see report by following options ,which we can see on spinner.

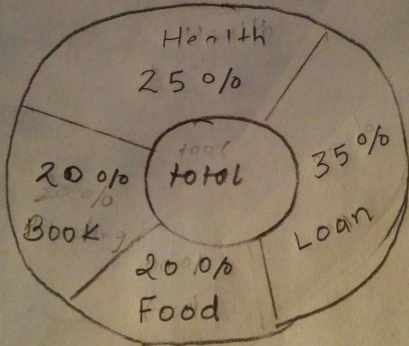
- Week
- Month
- Year

← History/Reports

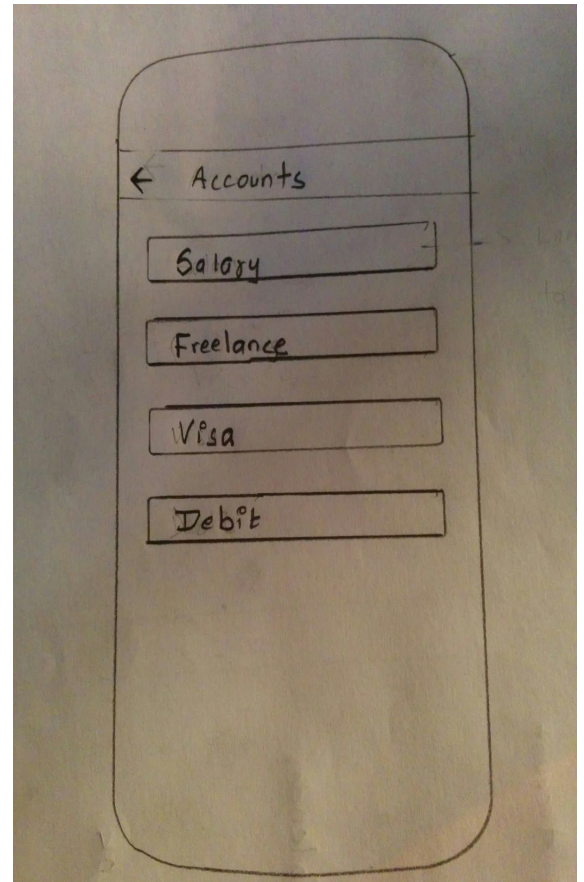
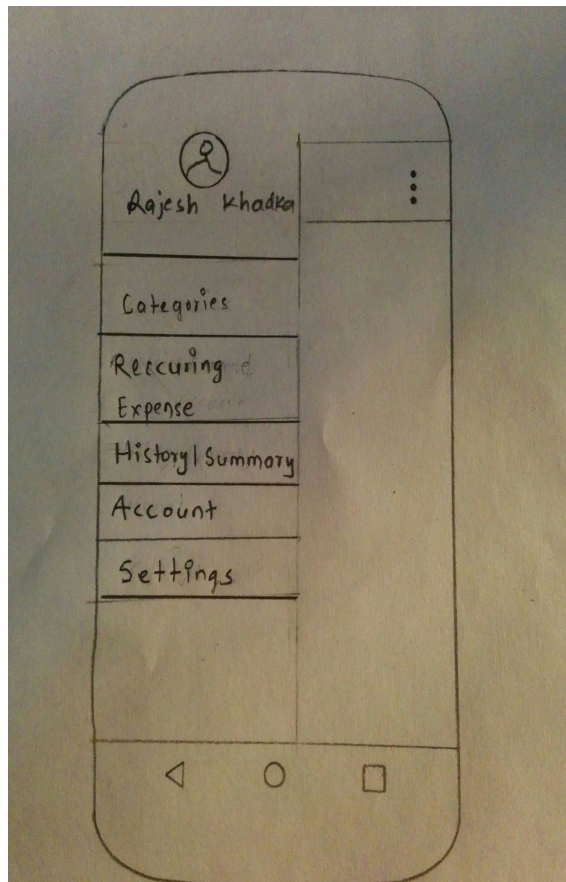
History

Reports

Week ▾



Account



Account will be of two type.

- Recurring Account
- Non-Recurring Account

Recurring Account

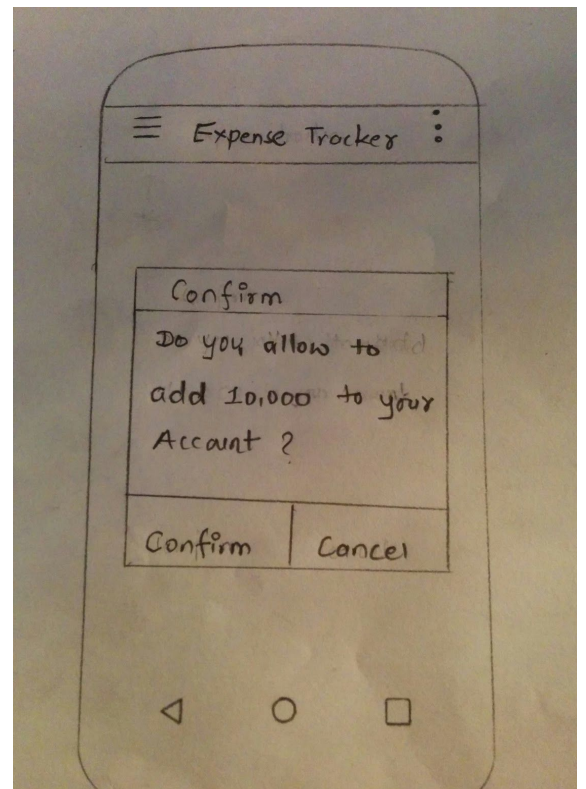
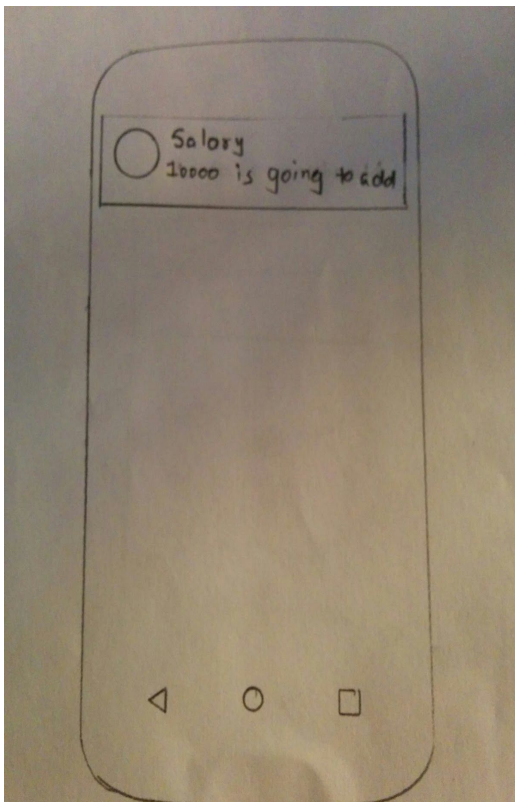
In our context employee are paid based on per month. So salary will be our recurring account type ,which will be added automatically to our account if it is specified as recurring account. Application will ask to add this amount in account by providing notification.

Non-Recurring Account

It may be amount sent by family member , bonus , earned from overtime bank balance etc.

Notification

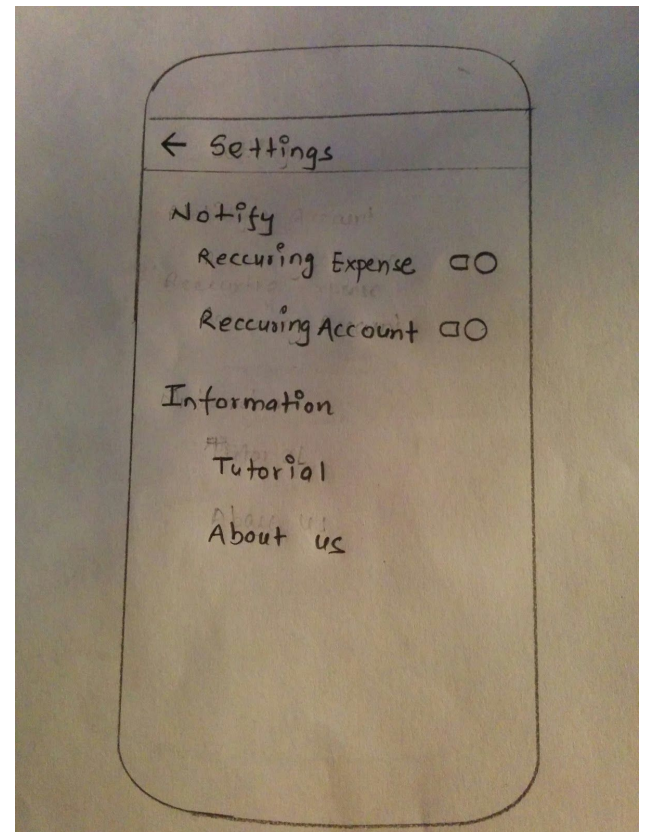
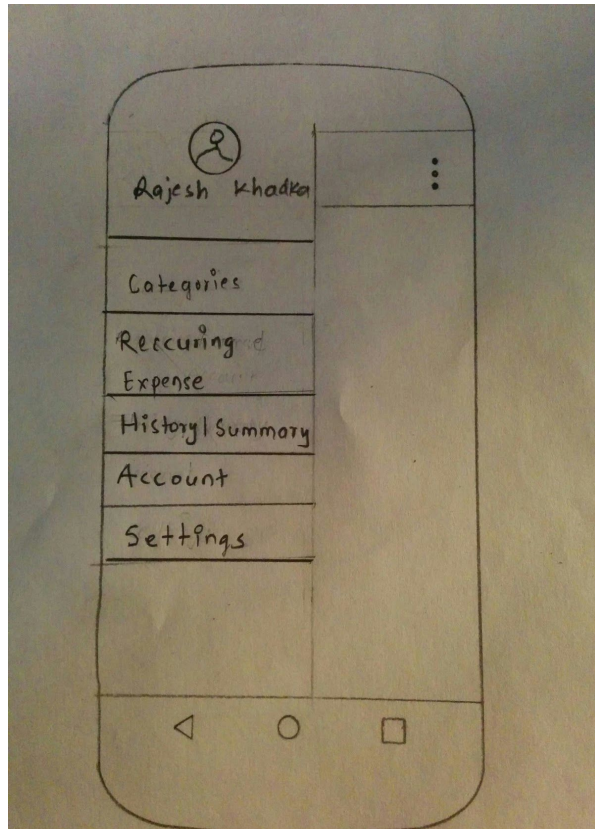
App will ask ask to user before adding recurring amount in the form of notification as:



User can edit account by long pressing and delete by swiping left to right. Edit will be helpful if the date of getting salary changes.

Settings

User can enable or disable the notification of recurring account and expense from settings. User can also see app guide on tutorial.



Key Considerations

How will your app handle data persistence?

Data will be stored by the help of Realm.

Describe any corner cases in the UX.

Fragment will be popped from stack on back navigation

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

Realm

Realm will be used for storing data related to expenses. It will be faster to use realm than through content provider. [Link](#)

Timber

For logging purpose. [Link](#)

MPAndroidChart

This will be used for showing pie chart. [Link](#)

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

- Create Project name Expense Tracker in IDE android studio
- Create theme for project
- Create folder structure
- Configure libraries [AndroidMpChart](#), [Realm](#), [Timber](#)
- Make Github repository and submit

Task 2: Make Database

- Create database table for Account
- Create database table for Expense
- Create database table for Categories

Task 3: Implement Dashboard Feature

- Create layout
- Make DashBoard Fragment
- Make Navigation drawer
- Make recyclerview adapter for listing expenses
- Fetch total amount of current month from database
- Fetch total expense of current month from database
- Set data to view
- Implement Edit and Delete expense feature
- Add Action on Floating action button
- Implement feature to show data of current date

Task 4: Account

Add Account

Describe the next task. List the subtasks. For example:

- Create layout
- Insert created account into database

Display Account

- Create layout
- Get all account from database
- Make Recyclerview adapter
- Set data to view
- Make Editable to Recurring account
- Sort recurring account

Task 5: Expenses

Add Expense

- Create Layout
- Keep action on UI element
- Insert expense into database

Display Recurring Expense

- Create Layout
- Create RecyclerView adapter
- Get data from database
- Set data to view
- Make Recyclerview Item delete able on swiping gesture left to right
- Make Recyclerview item editable on swiping gesture Long Press
- Sort recurring expense so that it will be easier to edit

Task 6: History/Report

- Create Layout
- Make ViewPagerAdapter
- Make Fragment History and Report
- Make Recyclerview Adapter for listing all expenses
- Make Spinner to show options : week,month,year
- Fetch data from database
- Display data fetched into pie chart
- Change pie chart content on spinner content change

Task 7: Settings

- Create layout
- Create Settings Activity
- Store data to default shared preferences
- Update data of sharedpreferences on settings item change

Task 8: Show Notification

- Create Notification layout
- Create broadcastreceiver to show notification
- Get expenses and accounts of recurring type form database
- Implement business logic to show notification

- Make Action able to notification
- Create confirmation dialog

Task 9: Expense Categories

- Create layout
- Make recyclerview adapter to display existing categories
- Create fragment dialog to choose color of categories
- Insert created categories into database

Task 10: Orientation Change Handle

- Handle Orientation change in DashBoard activity