

[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: Setup and implement Firebase Auth](#)

[Task 4: Implement Main Activity](#)

[Task 5: Implement Adding Activity](#)

[Task 6: Implement Widget](#)

[Task 7: Polish UI](#)

**Developer:** Dongmyung Ahn

## Expense Diary

### Description

Expense Diary is expense tracking application which you can track your expenses daily, weekly, monthly, and yearly.

### Intended User

Someone who want to manage expenses wisely, or who want to know which expense was high during some periods.

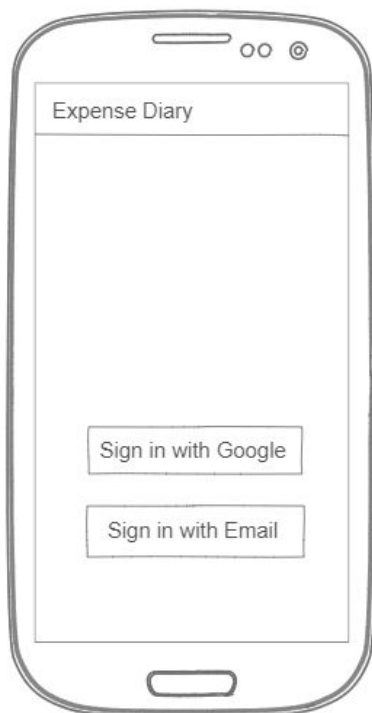
## Features

Here is the list of the main features of the app :

- Add / Modify / Remove expense.
- Save data and show the history of spending.
- Home Screen Widget to add expense easily.
- Firebase Authentication for login and registration.
- Firebase Realtime Database for data persistence in all devices.

## User Interface Mocks

### Login Screen



Login Screen using Firebase Auth

## Sign in with email Screen



Sign in with email Screen using Firebase Auth

## Create Account Screen



Create Account Screen using Firebase Auth

## Home Screen



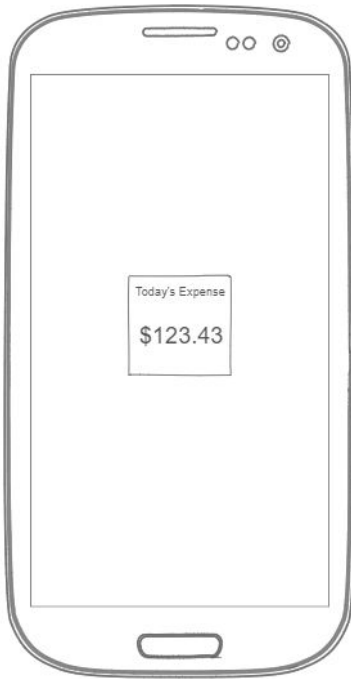
After login, this is home screen. The user can see some history of expenses and can add new expense by clicking FAB.

## Add new expense Screen



The user can add new expense after fill data.

## Widget Screen



Simple widget to show today's expense. If the user click the widget, app will launch.

## Key Considerations

### Programming Language

Application will be written solely in the Java Programming Language.

### Development IDE

Application will be developed using Android Studio Version 3.4.1.

### Resources

Application will keep all strings in a strings.xml file instead of being hard coded.

### Accessibility

Application will include support for accessibility.

### How will your app handle data persistence?

User expense data will be persisted using Firebase Realtime Database.

### IntentService

IntentService will be used for developing widget for the app.

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

If there is no expense data, it will show empty view.

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

- Butterknife : For easier field and method binding (Ver. 10.1.0)
- Material Component : For better and clean design (Ver. 1.0.0)
- Firebase Auth : For easier simpler authentication (Ver. 17.0.0)
- Firebase Realtime Database : For data persistence (Ver. 17.0.0)
- Google adMobs : For using ads. (Ver. 17.2.0)
- Gradle (Ver. 5.1.1)

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

- Firebase Auth will be used for user authentication.
- Firebase Realtime Database will be used for data persistence.
- Google adMobs will be used for ads.

## Next Steps: Required Tasks

### Task 1: Project Setup

- Create the project in Android Studio.
- Setup gradle dependencies.
- Setup project in Firebase Console

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

- Build UI for MainActivity
- Build UI for Adding Expense Activity

## **Task 3: Setup and implement Firebase Auth**

- Setup Firebase Auth
- Implement Firebase Auth in the project

## **Task 4: Implement Main Activity**

- Setup Firebase Realtime Database
- Implement Main Activity

## **Task 5: Implement Adding Activity**

- Implement Adding Activity

## **Task 6: Implement Widget**

- Build UI for Widget
- Implement Widget

## **Task 7: Polish UI**

- Polish UI using Material Design component and guide.