High-Level Design Document

Group D

32141631 Juhwan Moon

32154019 Hansol Jang

32170854 Yujin Kim

32174035 Soyeon Jung

32179181 Davi Chalita Meneguelli

Contents

Document Revision History

Date	Version	Author	Description
11/26/2019	V0.1	Soyeon Jung	• Initial Writing
11/26/2019	V0.2	Soyeon Jung	• Added Introduction Definitions, Acronyms and Abbreviations
11/26/2019	V0.3	Yujin Kim	• Added Design Considerations
11/27/2019	V0.4	Hansol Jang	• Added System Architecture
11/30/2019	V0.5	Juhwan Moon , Davi Chalita Meneguelli	• Added Detailed System Design Class Descriptions Dynamic Model

1. Introduction

1.1. Purpose

The purpose of this document is to present a detailed description of the Money Manager Application. It will explain the purpose and features of the application, the interfaces of the application, what the application will do, the constraints under which it must operate and how the constraints will react to external input.

1.2. Scope

Our goal is to people be alert about money usage. Using money manager, we will help those who are not sensitive about money flow and want to know about it. The application will separate the spendings on the following categories: food, transport, health, leisure, etc. It will make the user aware of his spending pattern and help on how to plan it in a more intelligent way.

1.3. Definitions, Acronyms and Abbreviations

Those following terms, acronyms, and abbreviations are throughout this document.

- Firebase: It is a web application owned by Google that has 3 main features: Authentication process (where the user can register and login with his registered account), Non-sql database (it stores and retrieve data from a non relational database) and Storage (where you can store videos, images, sounds, etc).
- Authentication: It is the act of proving an assertion, such as the identity of a computer system user.

2. Design Considerations

2.1. Assumptions and Dependencies

- Assumption

  Used for programs assuming good security. As it is a money planner application, security is a really important part. Therefore, the security is designed well so that there is no risk of hacking.

- Dependencies

  We need to set up a basic password entry to enable security.

2.2. General Constraints

- Hardware or software environment

  The application is developed with Android Studio and so on is made for phones and tablets with Android operational system installed.

- End-user environment

  Users must have an email account to use this application.

- Security requirements (or other such regulations)

  It is not possible to use this application without an email account, because it requests email and password inputs for the user to have access to the money planner.

- Memory and other capacity limitations

  Since all data is stored in firebase, the space used by memory is only the size of the application itself.

- Performance requirements

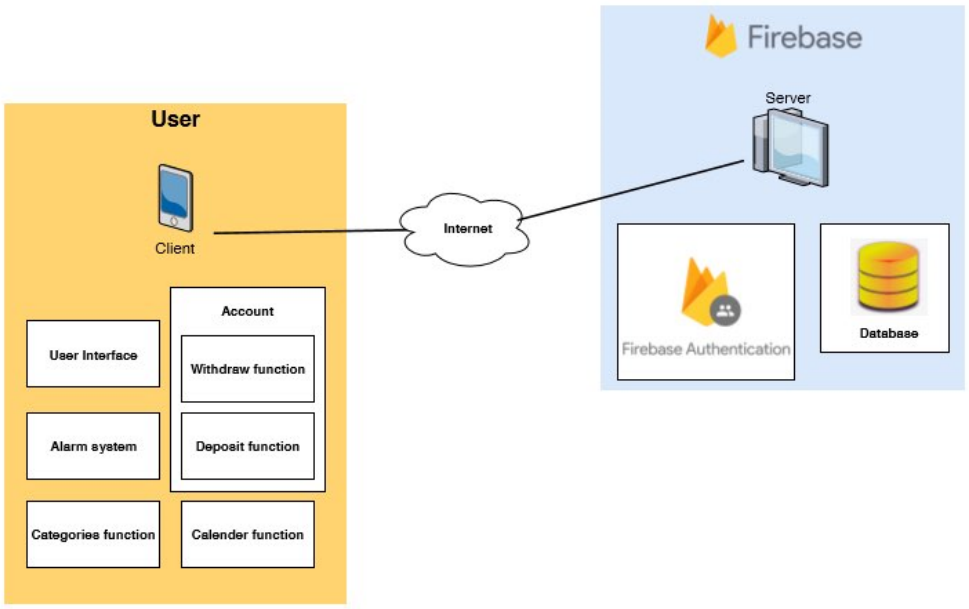
  It is in a format which the user enters deposits and withdrawals himself.

- Network communications

  This application can only be used in a networkable environment because the application requires communication with the firebase server.

3. System Architecture

3.1. Overview of system architecture



3.2. Components of system

3.2.1. Application(User)

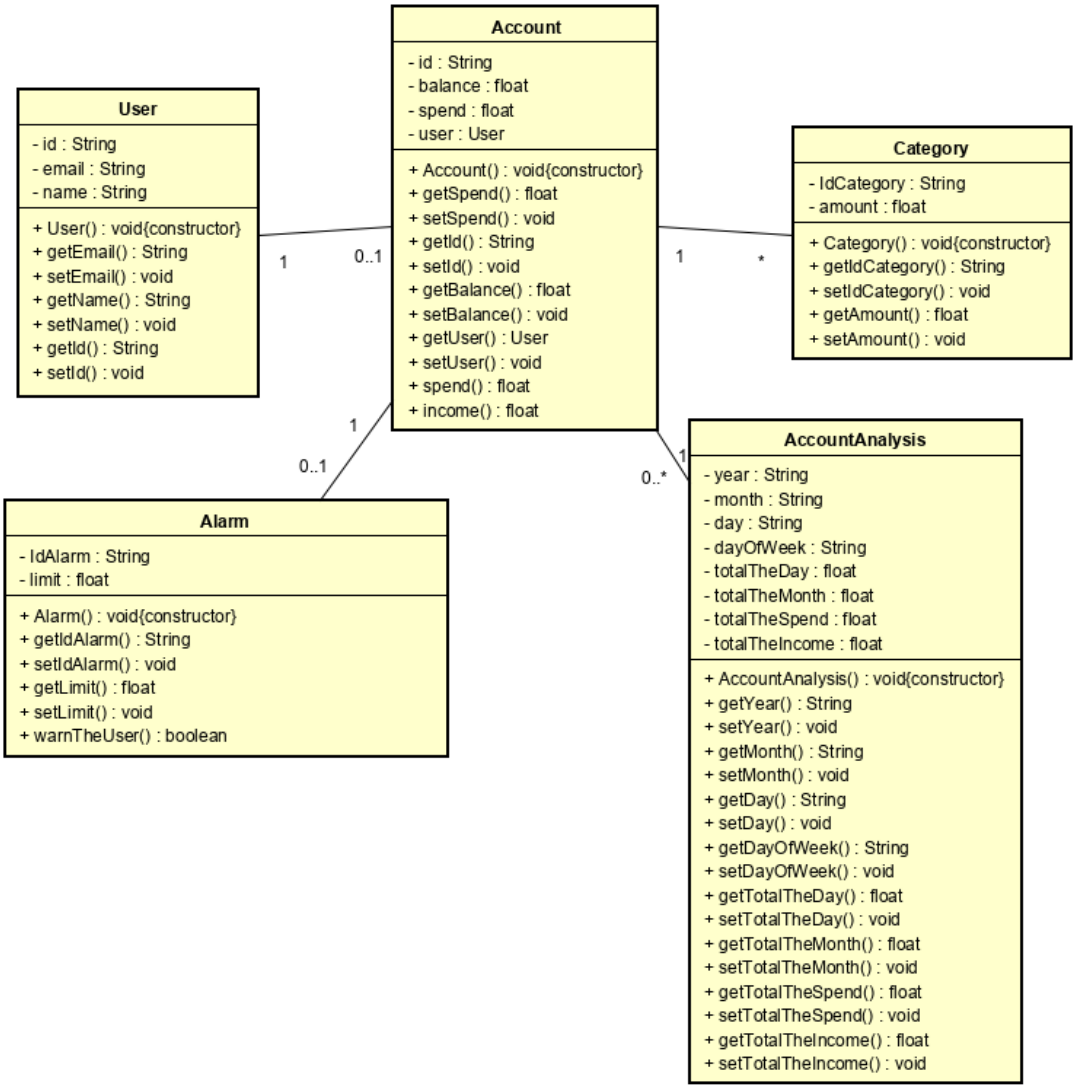
Our application is divided into five parts. The first part is the User Interface. It allows the user to enter an ID and password to login. The second is the Account section, it shows the balance and allows you to enter deposits and withdrawals.The third, the Alarm function, sets a limit and the alarm is triggered when the spending reaches the limit set beforehanded. The fourth is the Category function that shows the amount withdrawn divided by each category. And the last, the Calendar function shows deposits and withdrawals by date and month.

3.2.2. Firebase(Server and Authentication)

Firebase stores the email and password that are used to login. Then save and update the balance and others variables in it, connecting the user with his own information.

4. Detailed System Design

4.1. Class Diagram



4.2. Description of Class Diagram

Class name <User>

-> Attribute :

id (String)
email (String)
name (String)

->Descriptions :

This class has e-mail and name because each user must be managed with their ID and balance value in Firebase.

-> Methods :

get Method - Method that allows value to be returned.
set Method - Method that allows value to be saved.

Class name <Account>

-> Attribute :

id(String)
balance(float)
spend(float)
user(User)

->Descriptions :

User object is required because different balance values are managed for each user. This class manages balance value by receiving 'spend' and 'income' from each users.

-> Methods :

get Method - Method that allows value to be returned.
set Method - Method that allows value to be saved.
spend Method - Method to subtract spend values from current balance.
income Method - Method to adds income values to current balance.

Class name <Alarm>

-> Attribute :

IdAlarm(String)
limit(float)

->Descriptions :

User needs to specify a limit for the balance, so the limit variable is required.
This class compared limit and balance to trigger an alarm.

-> Methods :

get Method - Method that allows value to be returned.
set Method - Method that allows value to be saved.
warnTheUser Method - Method triggers a push notification

Class name <Category>

-> Attribute :

IdCategory(String)
amount(float)

->Descriptions :

This class receives input from users which categories are associated with their spent.

-> Methods :

get Method - Method that allows value to be returned.
set Method - Method that allows value to be saved.

Class name <Account Analysis>

-> Attribute :

year(String)
month(String)
day(String)
dayOfWeek(String)
totalTheDay(float)
totalTheMonth(float)
totalTheSpend(float)
totalTheIncome(float)

->Descriptions :

This class represents the amount used every day and the amount spent during the month

-> Methods :

get Method - Method that allows value to be returned.
set Method - Method that allows value to be saved.

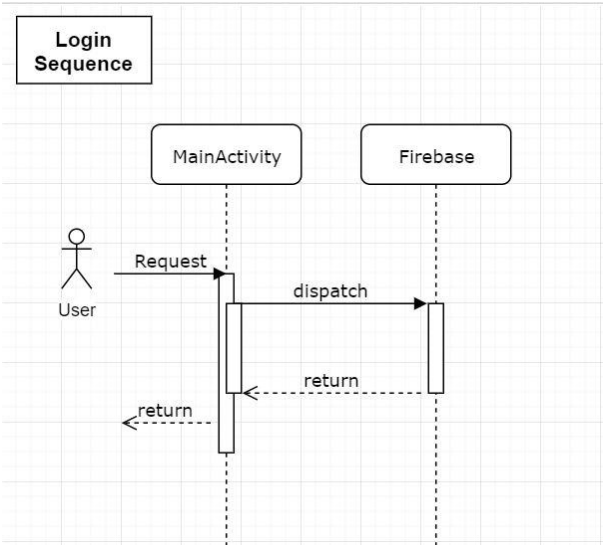
4.3. Dynamic Model

4.3.1. Scenario: <Log in>

4.3.1.1. Scenario Description

After starting the application, app will ask to Register or Log-in. As the user who uses at the first time do not have any ID/PW the sequence will begin by register. When user register, all the important data will be stored at the firebase authentication.

4.3.1.2. Sequence Diagram

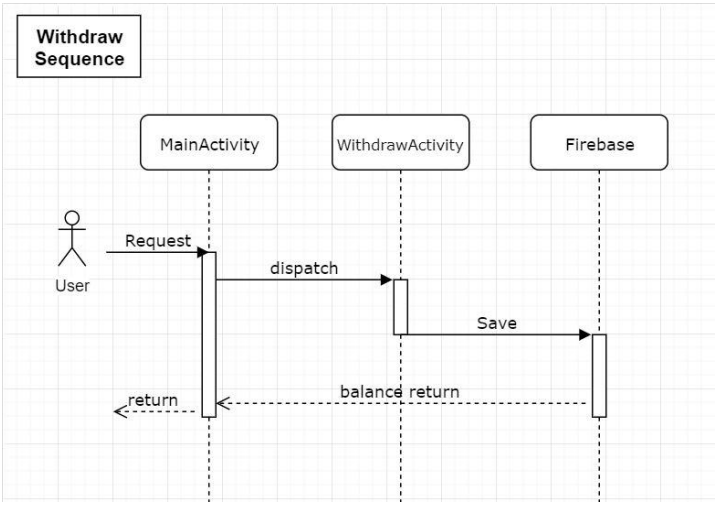


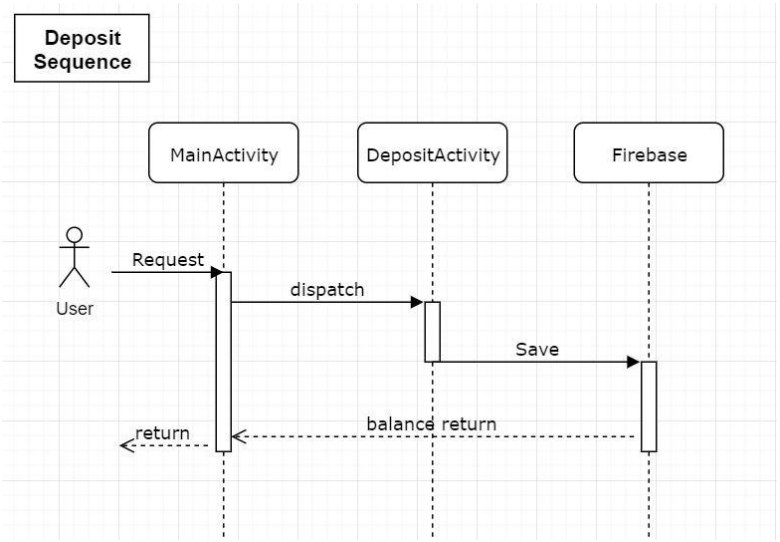
4.3.2. Scenario: <Balance input deposit/withdrawal>

4.3.2.1. Scenario Description

The users enters the balance. The balance entered is stored in firebase. When a users makes a deposit or withdrawal, the balance value is updated accordingly. Balance value changed by deposit/withdrawal is updated in firebase.

4.3.2.2. Sequence Diagram



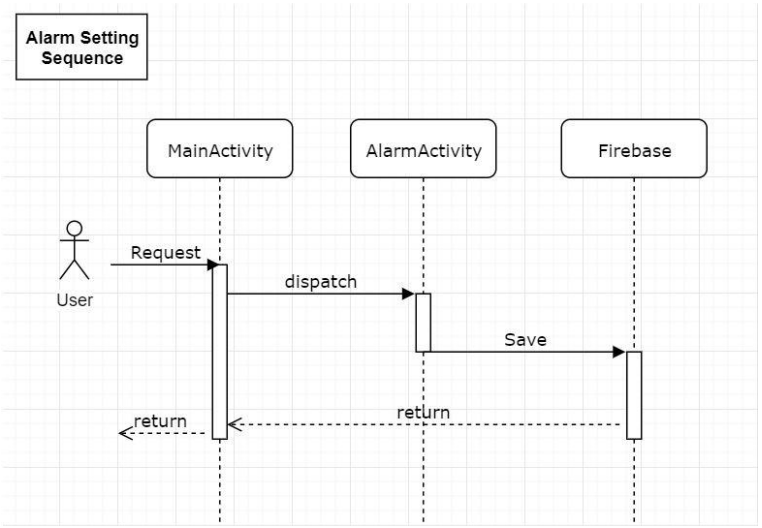


4.3.3. Scenario: <Alarm setting>

4.3.3.1. Scenario Description

The user can set an alarm that rings when the limit inserted is reached. Push notification occurs when the total spend value of the current user reaches the limit. The limit value is stored in firebase so that once the limit is specified, it is maintained and can be replaced.

4.3.3.2. Sequence Diagram

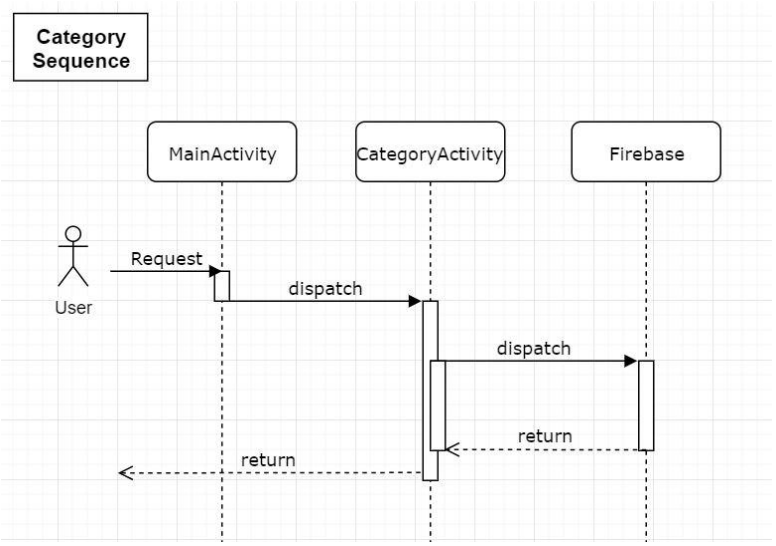


4.3.4. Scenario: <Category>

4.3.4.1. Scenario Description

Divided the amount withdrawn into food,leisure, traffic, shopping, etc. The pie chart shows the percentage by category.

4.3.4.2. Sequence Diagram

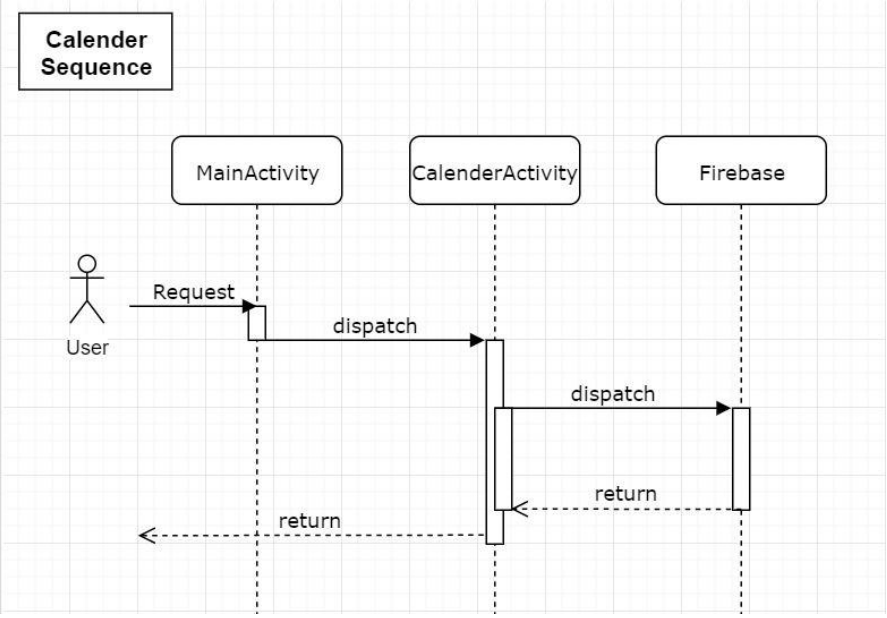


4.3.5. Scenario: <Calendar>

4.3.5.1. Scenario Description

Clicking on a date in calendar will show user how much was spent in the day, how much was earned, and the total amount.

4.3.5.2. Sequence Diagram



4.4. User Interfaces

This app needs to have an easy interface so the users will be able to understand the appearance and layout of the contents. We can increase the ease of use with intuitive icons that helps to identify features and uses. After that, the display which shows the balance will appear. When the users click the menu, they can confirm their monthly expenditure analysis, in addition, they can set up the alarm which rings if it is over the set limit. All activities will be designed to be opened, closed or swiped.

5. Roles and Responsibilities

Name	Role	Email	Etc.
Juhwan Moon	Calendar	juhwan0718@gmail.com	
Hansol Jang	Alarm	j0218j300@gmail.com	
Yujin Kim	Deposit/Withdraw -main	best5537@gmail.com	
Soyeon Jung	Category	jjm4414@gmail.com	
Davi Chalita Meneguelli	Firebase	chalitameneguelli14@gmail.com	leader