Software Design Document

1. Introduction

Today, with cloud becoming more and more used, collaboration becomes easier and more efficient. The purpose of this application is to provide a mobile-friendly collaboration planning application to help make collaborations easier to handle and to make communication of ideas flow smoothly. The application includes a user's own personal account to protect personal data and prevent data being leaked. From here each user will be able to create groups or be added to groups which helps them begin their collaboration. Each group will display a set of updates on the landing screen to keep users informed of changes. Each group will also have a section dedicated to meeting plans, task assignments and a cloud file storage system which helps plan meetings and organise the structure of the group. A final additional feature which may be of convince will be a vanilla chat embedded into the application. It allows users to communicate without having to switch to another chatting application.

1.1 Application vision

Most collaboration applications today are useful; however, the number of mobile-friendly collaboration applications is relatively low. The innovative feature of this application is that it allows users to collaborate and share data in real-time whilst also planning meetings and task assignment.

1.2 Scope

For this design release, I will be adding a set of items which should be implemented by the end of this release. Some features may be added extended with more functionality; however, this will only occur if time permits.

The first constraint identified is the funding of this project. As Firebase is currently the primary driver of this application, the free tier of firebase limits the amount of data that can be stored in both the Realtime Database and Cloud Firestore, so it is assumed that this application will only be capable of handling a medium sized number of groups.

1.3 Document Version History

Version	Release Date (DD/MM/YYYY)	Comments
V1.0	07/05/2018	Added in introduction, application vision and functionality overview

2. Functionality Overview

1. Account Creation and Management

Users will be able to create an account using email and password. Their account may contain details such as phone number, email and a profile picture which can be edited by the account owner. After each account is created, users will receive an email to confirm their email to activate their account. Users may also reset their password through an email link, if they happen to forget their password.

2. Private Groups

This function of the application allows users to view groups that are in. This feature allows users to view groups they've created or groups in which they are a member of. This helps prevents other users from wondering into restricted groups and stealing ideas or sabotaging any groups. As of now groups will only be allowed to have a maximum of 5 members plus 1 admin, for a total of 6 members.

3. Posting Updates

These updates will be displayed on the "home screen" of a group. This will allow members of the group to read these updates when they access the group. Users will be allowed to add, edit and remove updates. It is important to note that users may only edit and remove their own updates. Members will also be allowed to add any comments they like to the update

4. Meeting Planning

This aspect of the application will help compile meetings in one section, which prevents any miscommunication. Users will be able to enter a meeting title, select a location by entering a location (using the google places API), selecting a date and a time. They may also add optional information if needed. Additionally, creators of the meeting may edit or remove the meeting. In addition to planning meetings, users will also be allowed to notify their group about their distance from the group expressed in minutes.

5. Cloud File Sharing

As editing text files is much easier on a computer, as of now users will not able to edit files on the cloud but may view it. Users will be allowed to share the images either by directly capturing their camera or by selecting an image in the gallery. Files can also be shared by selecting a file from the file manager. As per of now, only administrators will be allowed to remove files from the cloud, but this may change depending on feedback.

6. Task assigning

For this part of the application, only the administrator will be allowed to assign tasks to members. Tasks will display an image representing the status of the task (complete or incomplete). Only the assigned member will be allowed to mark the status of the task, this prevents any conflicting statuses of task.

7. Group Messaging

The final aspect of the application allows members to communicate ideas with each other. This just makes communication easier and more convenient instead of having to constantly switch apps just to chat.

2.1 Feature Summary

2.2 Application Flow Diagram

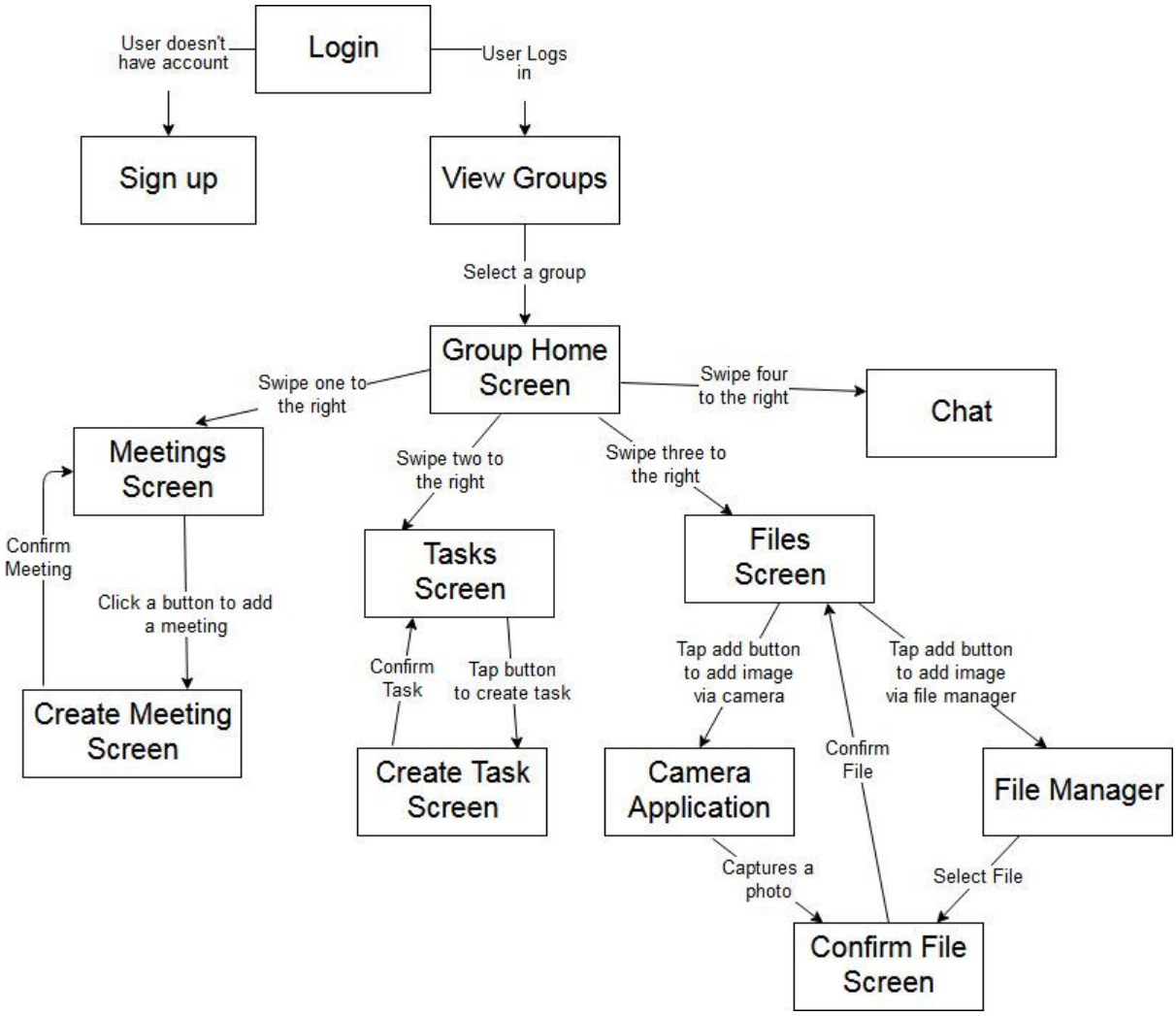

Figure 1. Application Flow Diagram

3. Design

3.1 Overall Components Overview

3.2 Mockups

3.3 Used Application Programming Interfaces

1. Firebase Authentication

Firebase authentication is library that provides many backend services, such as storing user account data and authenticating users. The data is also stored on a cloud storage, which allows for users to have the same experience and data across different devices. I used this library, since it allows the application to store user data on the cloud allowing for login on multiple devices. This library also assists with authenticating users asynchronously, which helps prevent excessive resource consumption on the UI thread. Finally, this library also provides a set of methods which allows the application to extract data as required.

2. Firebase Realtime Database

Firebase Realtime Database is a library that enables applications to store and sync data Firebase's NoSQL (Not Only SQL) cloud database which stores data in JSON. I used this library, since it allows for every user to view changes in real-time, allows users to view the same data if authorised and it also allows the application to operate even when network connectivity is unavailable.

3. Glide

4. Butter Knife

4. Application Structure

4.1 Packages

1. activity

This package contains every activity in the application. The classes in this package include

- CreateGroupActivity
- HomeActivity
- LoginActivity
- RegisterActivity
- ViewGroupActivity

2. adapter

The contains within this package are the adapters that help populate AdapterView's. The classes in this package are:

- GroupAdater
- UpdateAdapter
- ViewPagerAdapter

3. model

This package contains the POJO models of the application. The classes here include:

- Group
- Update
- User

4. ViewHolder

This package contains the ViewHolder classes which help cache views, allowing recycling to occur. The classes in the package are:

- GroupViewHolder
- UpdateViewHolder

4.2 Activities

4.3 Services

4.4 Others

5. Data Structures

6. Testing

7. Conclusion

8. References

References for code snippets (MAD):

1. https://stackoverflow.com/questions/6054562/how-to-make-the-corners-of-a-button-round
- This is for the xml code to make corners round

2. https://www.viralandroid.com/2015/09/how-to-add-ripple-effect-to-android-button.html
- This is for the xml code to make a ripple animation on button tap

3. https://firebase.google.com/docs/auth/android/password-auth
- Firebase authentication API and documentation

4. https://developer.android.com/training/appbar/setting-up.html#java

- Toolbar tutorial

5. https://stackoverflow.com/questions/23578059/make-activity-animate-from-top-to-bottom
- Custom animation for starting a new activity