



CSCI 5708: MOBILE COMPUTING

PROJECT PROPOSAL

GROUP 1:

FAIZA AZIZ UMATIYA (B00899642)

HRISHI PATEL (B00911902)

PARVISH VIJAY GAJJAR (B00912090)

SIDDHARTH KHARVAR (B00897211)

SHRAVYA REDDY GENNEPALLY (B00911193)

CONTENT

1. ABSTRACT .	1
2. INTRODUCTION AND BACKGROUND	1
3. TARGETED AUDIENCE.....	1
4. PURPOSE, BENEFIT, AND USAGE	1
5. FUNCTIONALITY	2
6. ARCHITECTURE	8
7. TOOLS.....	9
8. REFERENCES.....	11

1. ABSTRACT

Our team aims to develop a mobile application that facilitates social interactions to meet new like-minded people. The application would enable users to connect with others with the same interests. The app will encourage users to make friends, showcase their professional or personal work, and participate in activities that interest them. The purpose of this platform is to foster a sense of community. Users will be inspired to broaden their horizons, surpass their comfort zones, and work together to achieve their goals. The user has the option of selecting how many individuals to meet. Our application differs from the usual social media apps, focusing more on meeting people with similar interests. Our application gives options between choosing to meet people with the help of 1) Social feature: a one-on-one matchmaking feature to meet like-minded people, 2) Events: where the application's community could host events, 3) Group Meetups: where people could host recurring meetups and could build a community with the help of a group chat.

2. INTRODUCTION AND BACKGROUND

People lost touch with social connectivity during the COVID epidemic, which resulted in a lack of emotional support and face-to-face interaction. It became challenging to return to normal. Therefore, it's crucial to surround yourself with like-minded individuals during tough times. How can you locate individuals who share your interests and job goals? The simplest solution would be to look for that person using an app that allows you to do so whenever and wherever you desire. The primary purpose of the software is to let users connect with others, establish friendships, find support, expand their businesses, and pursue their interests [1]. Other apps already exist and encourage social interaction, but the one we will create enables users to participate in group and event meetups and solo meetups. We plan to implement these functionalities and allow users to share their profile information and other social media handles so they can connect better with others through a QR code.

3. TARGETED AUDIENCE

When talking about the target audience, we look at the rule of following the big C, which is the context. The application aims to involve students going to universities, professors, other event organizers, friends who would like to meet up as they are interested in talking about something, and strangers who have similar interests.

4. PURPOSE, BENEFIT, AND USAGE

Our app is suitable for all students trying to build a community by meeting up with like-minded people and interacting with them. We welcome students to explore different cultures, gain knowledge on different aspects of life, find their interests and much more. This application will allow the students to explore the nearby get-togethers we call a 'MeetUp' and talk on a specific agenda. We welcome people of all interests, whether talking about the latest technologies or just having a drink and meeting new people.

In addition, the users can also organize their meet-ups to allow others to sign up for them and form a community. Our application can allow befriending other users and forming a group to organize private MeetUps. It often becomes a hassle to find a user by typing their name into a search bar and looking through tens of users before one can find them. Well, MeetUp would allow one to use Near-Field Communication (NFC) friending. The user needs only tap the mobile onto the user's mobile that they want to befriend, and you have your friend.

5. FUNCTIONALITY

5.1. Minimum Functionality

The minimum functionality of the application is as follows:

- Signup
The app will allow users to sign up with email address or with their Google account.

Use Case	Sign up
Actor	New users
Pre-condition	Users must have provided a valid email address, conforms to password specification, and should not already be registered.
Flow	After the user is done with sign up, the user will be sent to the Meetup feature.

- Login
The app will allow users to login with their Google account or with their already registered account.

Use Case	Log In
Actor	New Users (if signing via Google), Existing Users
Pre-condition	A new user can directly login with Google. If the user is using email address to login, the user must be already signed up.
Flow	After the user is logged in, the user will be sent to the Meetup feature.

- Update Profile

The user will be able to update information such as their first name, last name, phone number, gender, their interests, profile picture. The user will also be able to deactivate the account.

Use Case	Update Profile Details
Actor	User
Pre-condition	The user must be registered
Flow	If the user is logged in, the user will be able to update their profile.

Use Case	Delete Account
Actor	User
Pre-condition	The user must be logged in.
Flow	The users will be able to delete the account if they are logged in. After deleting the user account, the user will be returned to the login/sign up activity.

- Solo Meetups

The user will be able to find people with similar interests in Dalhousie University.

Use Case	Find a person with similar interests.
Actor	User
Pre-condition	The user must be registered and must have a profile picture.
Flow	Upon matching with a person with similar interest, the user will be able to chat with the person.

- Event Meetups

The user will be able to have a look at all the events hosted by Dalhousie students. The user could also create an event. For every event, there would be a comments section for general questions and discussions regarding the event.

Use Case	Find events hosted by Dalhousie students
Actor	User

Pre-condition	The user must be logged in.
Flow	The user can view event details after user click on an event. The event details activity will display a banner image for the event, description, number of people who are participating in the event.

Use Case	View QnA
Actor	User
Pre-condition	The user must be logged in.
Flow	For every event, there would be a discussion board in the event details activity. The users will be able to view comments made by other participants.

Use Case	Post a question in discussion
Actor	User
Pre-condition	The user must be logged in.
Flow	The user can use the “Add Question” button to add a comment they have written.

Use Case	Reply to a comment
Actor	User (Organizer)
Pre-condition	The user must be registered and must be the organizer of the event.
Flow	The user can reply to a question made by other user by clicking on “Answer” button for every comment made.

Use Case	Participate in an event
Actor	User
Pre-condition	The user must be logged in.

Flow	The user can click on “Participate” button to participate in the event. Upon successful participation, the user will see event confirmation and a ticket number.
------	------------------------------------------------------------------------------------------------------------------------------------------------------------------

Use Case	View previously participated events
Actor	User
Pre-condition	The user must be logged in.
Flow	The user will be able to view a list of previously participated events by clicking on “View Previous Events”.

Use Case	View participated users
Actor	User (Organizer)
Pre-condition	The user must be logged in and user must be organizer.
Flow	The user (organizer) will be able to view a list of all participants for a specific event.

5.2. Expected Functionality

- Reset Password

The user can reset their password if they forget their password.

Use Case	Reset Password
Actor	User
Pre-condition	The user must be logged out.
Flow	The user will be able to reset their password by clicking on “Reset Password” link from login/signup page. Upon clicking that, the user will be asked to email address and the user will be sent a password reset email.

- Search Events

By default, the user will be shown all the recent events happening around them. Search will provide them with options to search a specific event by name or by category.

Use Case	Search Events
----------	---------------

Actor	User
Pre-condition	The user must be logged in.
Flow	The user will be able to search events by category or title.

- Notifications

The app will send various kinds of notifications depending on the different user actions like when user posts questions, when there is any match and so on. The user will get notifications for chats, event or group meetup related notification etc.

Use Case	Receive notification when a user matches with you
Actor	User
Pre-condition	The user must be logged in.
Flow	The user will receive a notification when a user matches with them in the Solo Meetup feature.

Use Case	Receive notification when a user posts question on event
Actor	Organizer User
Pre-condition	The user must be logged in and must be organizer of the event
Flow	The user will receive a notification when a user posts a question on the QnA section of an event.

Use Case	Receive notification when a user gets reply from organizer
Actor	User
Pre-condition	The user must be logged in and must be organizer of the event
Flow	The user will receive a notification when an organizer replies to a question a user asked on the QnA section of an event.

5.3. Bonus Functionality

- QR Code Share

Imagine you are at an event/meetup where you meet who you would like to connect with. This feature will help you share contact details like phone number, email address, LinkedIn, Snapchat etc. with just one QR Code scan.

Use Case	Share Personal Details
Actor	User
Pre-condition	The user must be logged in
Flow	The user can generate a QR code which has their contact information.

Use Case	Receive Personal Details
Actor	User
Pre-condition	The user must be logged in, the phone must be equipped with a camera.
Flow	The user can scan a QR code of another user. Upon a successful scan, the user will be able to view user contact details and could even save them directly to phone contacts.

- Paid events with payment gateway

Use Case	Purchase event ticket
Actor	User
Pre-condition	The user must be logged in.
Flow	The user will be able to purchase a ticket for a paid event.

- Group meetups

Suitable for communities and clubs. You can chat with other members of the group too. Recurring meetups are possible with Group meetups.

Use Case	View all group meetups
Actor	User
Pre-condition	The user must be logged in.

Flow	The user will be able to view all group meetups.
------	--------------------------------------------------

Use Case	Do group chats
Actor	User
Pre-condition	The user must be logged in and must be part of a group meetup
Flow	The users in the groups will be able to chat with each other (group chat).

6. ARCHITECTURE

We will be using an MVC pattern for the underlying architecture of our application.

The app will have four main features: Signup/Login (authentication), Social (Solo meetups), Events and Group meetups. Firebase offers multiple features with seamless integration with Kotlin. We plan on using Firebase Authentication, Firebase Firestore, and Cloud Storage.

For login and signup, we will be using Firebase Authentication. It also provides integration with sign-in providers like Google. Since Firebase Authentication creates a unique UID for every user, we can use that to identify each user. After a user signs up, the user's information would be stored in a “users” collection in Firestore Database. For the Social feature, we also take advantage of Firestore’s subcollection feature to store chat messages between users. Other entities, such as events and group meetups, will also be stored in their collection. Each event/group meetup will have its document. Firestore is a NoSQL database that works well with real-time data. NoSQL databases provide faster reads than a relational database, and our application can use this to enhance the user experience while using the solo meetup feature. Android Studio provides an inbuilt feature to integrate Firebase onto our application with just a few clicks; hence, for simplification purposes, Firebase is the ideal solution for our application’s backend.

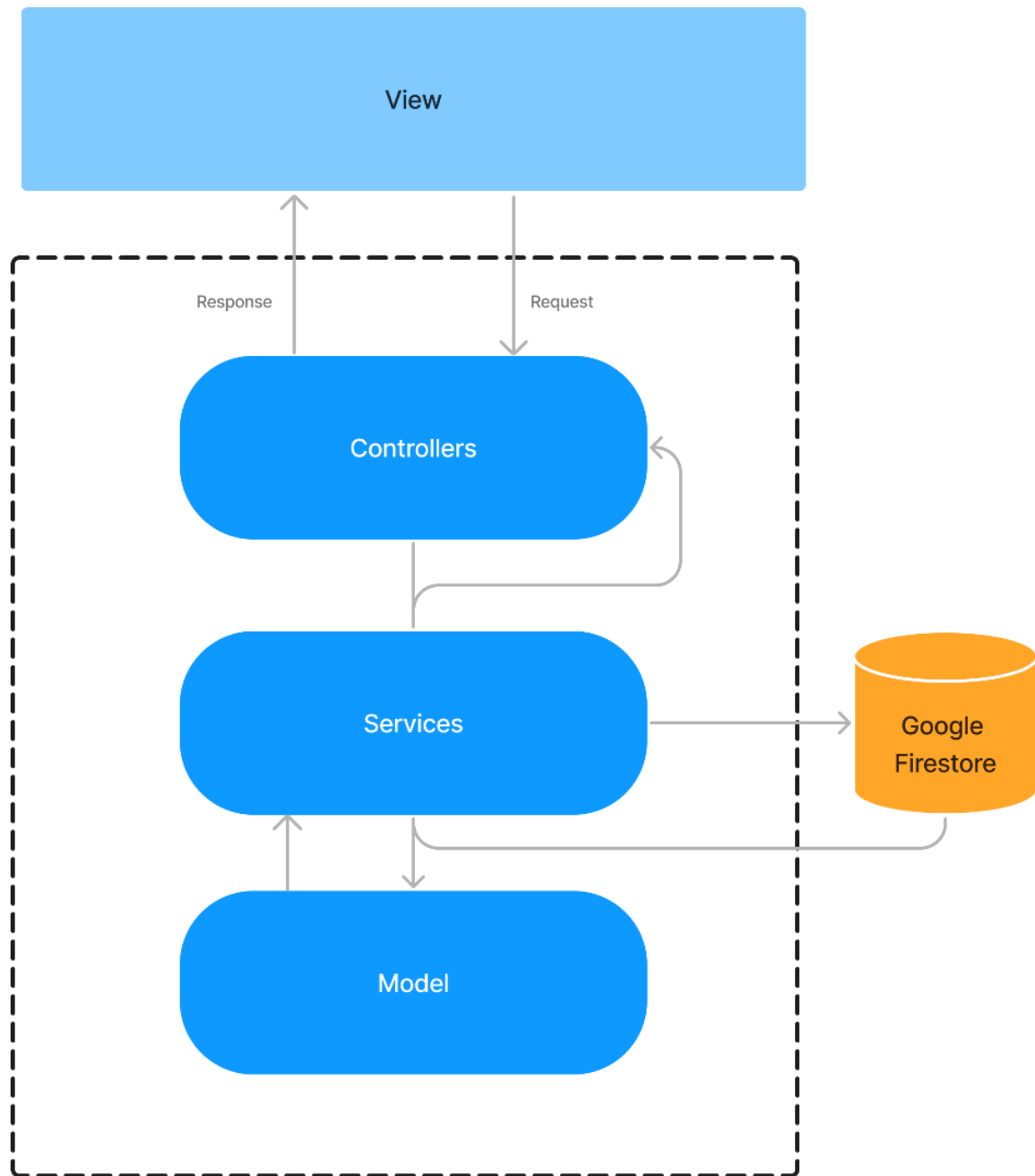


Figure 1: Application Architecture

7. TOOLS

For developing and running the proposed mobile application, we plan on using the following tools:

1. Android

We plan to use Android technology to build our application from the ground up. Android is a free, open-source mobile OS that Google is currently maintaining based on Linux Kernel [2].

2. Android Studio

The IDE that we are planning on using is Android Studio. It provides us with an inbuilt emulator to test our application as we are developing it, along with a Visual Layout Editor that allows us to design the UI easily [3].

3. Kotlin

The language of coding is going to be Kotlin. Kotlin is a modern, safe and concise programming language that is easy to pick up; hence; it helps create powerful applications immediately.

4. Gitlab

For version control of the code, we plan on using git, and GitLab is the control source tool that helps different developers collaborate and work on a single project [4].

5. Firebase

Firebase is an application development platform that helps in building with ease. It is also backed by Google and allows to store data in a remote location; hence it can be accessed from anywhere in the world [5]. Moreover, multiple features require a lot of reads per second, and a NoSQL database provides faster reading than a relational DB. Android Studio has built-in support for integrating Firestore into our Kotlin app and facilitates us with the features to fasten the initialization process. Not only this, but it also provides Firebase Authentication for authentication. So, we plan on using Firebase Authentication, Firestore as our database and Firebase Storage to save user information such as profile pictures and other images.

6. Jira

Our team plans to use the Jira board to track the project and manage user stories. To follow agile methodology, tracking various tasks is a must [6].

7. Figma

Figma is a design tool that combines some of the most powerful design features and an efficient workflow boot. Hence, we plan on using Figma for wireframing and building prototypes [7].

8. draw.io

Before we start our development, we need to ensure that our app's database layout and architecture are fixed. Draw.io has built-in features to build various diagrams, and it facilitates features clearly depicting the requirement [8].

8. Other

A Gantt chart is used to depict the project timeline. It displays various tasks and the time allotted to them over the term, with reasonable deadlines.

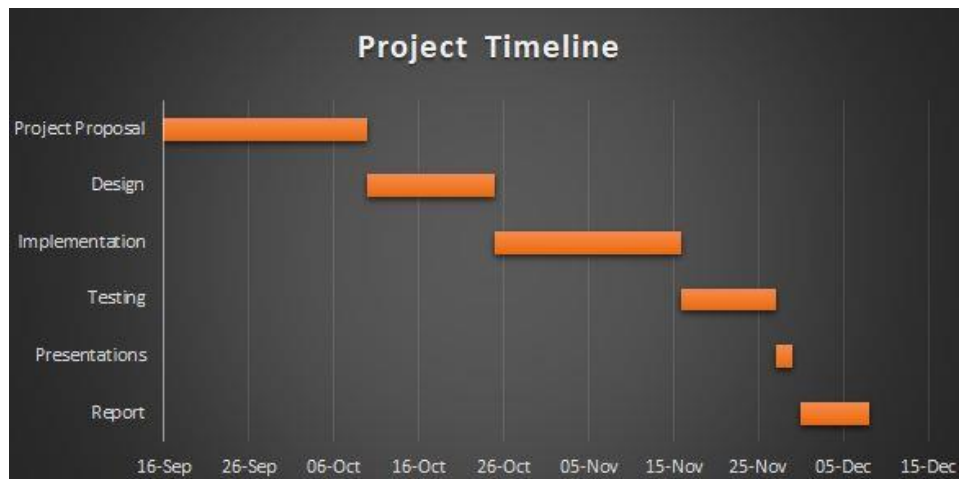


Figure 2: Gantt Chart

9. REFERENCES

- [1] "Meetup - We are what we do," *Meetup*. [Online]. Available: <https://www.meetup.com/>. [Accessed: 09-Oct-2022].
- [2] Android. 2022. What is Android? | Android. [online] Available at: https://www.android.com/intl/en_ca/what-is-android/ [Accessed 9 October 2022].
- [3] Android Developers. 2022. Download Android Studio & App Tools - Android Developers. [online] Available at: <https://developer.android.com/studio> [Accessed 9 October 2022].
- [4] About.gitlab.com. 2022. [online] Available at: <https://about.gitlab.com/why-gitlab/> [Accessed 9 October 2022].
- [5] Firebase. 2022. Firebase. [online] Available at: <https://firebase.google.com/> [Accessed 9 October 2022].
- [6] Atlassian. 2022. Jira | Issue & Project Tracking Software | Atlassian. [online] Available at: <https://www.atlassian.com/software/jira> [Accessed 9 October 2022].
- [7] Figma. 2022. Figma: the collaborative interface design tool. [online] Available at: <https://www.figma.com/> [Accessed 9 October 2022].
- [8] draw.io. 2022. draw.io – Diagrams for Confluence and Jira - draw.io. [online] Available at: <https://drawio-app.com/> [Accessed 9 October 2022].