Project Name: IceBreaker

Working App Name: IceBreaker: Catch Up w/ Friends!

Team Members: Elias Taylor, Fabio Tran

Project Overview:

- IceBreaker is a social networking mobile app that will encourage users to participate in a brief activity with their friends once a day through a randomized notification alert. Examples of activities include having friends play a simple game such as tic-tac-toe or rock-paper-scissors. Another type of activity is prompting a question such as a hypothetical or a “would you rather”. Furthermore, along with this interaction between individual friends, users should be able to join organizations (groups of users) where they will also interact with a random user daily. The goal of IceBreaker is to first help people maintain relationships with friends and loved ones whom we may forget to reach out to during our busy lives. Furthermore, the app will facilitate the growth of potential new relationships as it aids in "breaking the ice" where it might have been traditionally difficult for certain individuals. Prompting activities between users such as games, questions, etc., can encourage social interaction and can lead to relationship development, which is the primary goal and impact of this product.

Software Components:

- Git and GitHub (version control system)
    - Git is the most widely adopted version control system used by both small groups of individuals and large enterprise-level projects to manage the development of a product/s typically involving code.
    - Git is completely free and is classified as open-source software. It is under the GNU General Public License (GPL), which allows us to use it in our project’s development without having to pay any fees or royalties.
    - Git will track file changes and aid in allowing simultaneous work to be able to occur on the same project while making sure the integrity of the project is maintained. Having a complete history of everyone’s work is crucial as team members can collaborate and merge or revert if necessary. This is crucial in promoting coordination and productivity amongst a group.
    - GitHub is an online software development platform that provides hosting for Git repositories along with other useful features to aid in the development, collaboration, and management of a project.

    - GitHub is proprietary software owned by GitHub, Inc. but it is completely free to use at its base level with additional paid plans for advanced features.
    - Many teams and projects use GitHub to host their Git projects remotely online which is how it will be involved in our project as well. GitHub will help us by serving as the central hub for our team members to store, share, and manage code with themselves and others.
- Android Studio (primary integrated development environment)
    - Android Studio is the official integrated development environment (IDE) designed for Android app development.
    - Android Studio is a free-to-use proprietary software developed by Google which is also based on JetBrains' IntelliJ IDEA IDE software. Every development tool and feature are available without cost or fees.
    - Android Studio will be the primary integrated development environment our team members will use in developing our mobile application. This IDE provides a comprehensive set of tools and features for designing, developing, debugging, and testing Android applications. Furthermore, it provides a built-in Android Emulator that allows us to test our product on a plethora of virtual Android devices with varying screen sizes, resolutions, and even operating system versions. As this IDE is designed and promoted specifically for Android development, our team believes this is the right choice to help us build a quality Android application as efficiently as possible.
- Java and Kotlin (primary programming language/s)
    - Java is a high-level object-oriented programming language designed to be written and compiled to bytecode that can be executed on any platform with a Java Virtual Machine (JVM).
    - Java was created by Sun Microsystems in the mid-1990s and is open source under GNU General Public License (GPL) Version 2 making it free to use.
    - Kotlin is a statically typed high-level programming language with type inference. Its design is influenced by Java's shortcomings and is described as a modern language with a focus on safety, interoperability, expressiveness, and productivity.
    - Kotlin is developed by JetBrains and is an open-source language under the Apache License allowing us to use it to develop our Android application without paying any fees or royalties.
    - While Java has been the traditional language of choice for Android, in recent years the use of Kotlin has increased and is officially declared by Google as the primary preferred language for Android app development. Kotlin has

additional features that ease development and maintenance for programmers compared to Java such as null safety and concise syntax. To follow modern trends in the industry, our team will also use Kotlin in developing our Android app and will incorporate specific Java code and libraries when appropriate. Following this strategy should lead to an overall boost in productivity during the development phase of our project.

- Android SDK (primary software development kit)
    - The Android SDK is a software development kit that contains tools, libraries, and resources that are useful for developing Android-based applications.
    - The Android SDK was created by Google and is still continuously updated with new features and improvements to existing ones to improve the performance and experience of Android applications.
    - The Android SDK will be used in our project to aid in development as crucial functions and tools will be already readily available in various libraries for us to utilize rather than having to code every functionality completely from scratch. Doing so would be both redundant and time-consuming in our limited time window for the project.
- Jetpack Compose (user interface framework)
    - Jetpack Compose is a modern toolkit for building native user interfaces for Android applications.
    - Jetpack Compose was developed by Google and is a part of the Android Jetpack suite of libraries. It is a free and open-source tool under Apache License 2.0 which allows us to utilize its toolset without cost.
    - As our team is building an Android application natively, Jetpack Compose will be handy in the creation of an aesthetically pleasing and modern app that is also functional for users which will hopefully result in a satisfactory user experience. Along with benefits for end-users, its toolsets, and features will also help us as developers with productivity during coding as features such as gestures and animations are made easier to implement.
- Android framework (provide Android APIs)
    - Android framework is a comprehensive set of software components that aid developers in building Android-based applications such as providing a wide range of Android APIs for handling various components of the Android operating system and device hardware.
    - Android framework is developed and maintained by Google. It is free and open source under Apache License 2.0 which allows our team to use the source code with few restrictions. Furthermore, Google does offer additional proprietary services and libraries to the framework that are not open source.

    - Android framework will play a pivotal role in our project as its offering of Android APIs will support and be the backbone of most of our backend processes. The framework will help us work with a variety of functionalities and services such as dealing with user interface elements or user input/interactions. Furthermore, implementing notifications and alerts from our application to the user's device is an intended feature that the Android framework would simplify as it offers tools for this exact situation. Overall, these APIs will help our team build a feature-rich app that maintains quality and performance which will raise the experiences for users.
- Retrofit library (provide REST client to handle APIs)
    - Retrofit library is a type-safe HTTP client library for Java (and Kotlin) that is most used in developing Android applications. This library provides a concise and simple way to make API calls where developers can easily send requests and handle responses from any targeted web service.
    - Retrofit library is free and open-source software licensed under the Apache License 2.0 therefore we can take advantage of its offerings in our project.
    - Retrofit library will be used in our project whenever we need to interact with a web service to retrieve or manipulate data through its built-in HTTP client. Whether it's integration with third-party services, handling authentication/authorization, or accessing an SQL database, this library will ease the process of communicating with web services.
- SQLite (relational database management system)
    - SQLite is a lightweight embedded relational database management system (RDBMS). Moreover, it is a C programming language library that implements a fully functional SQL database engine and is intended for mobile and small-scale applications.
    - SQLite was created by D. Richard Hipp in the early 2000s and is still maintained by him and community contributors to this day. It is completely open source and free to use as it is released into the public domain and not limited by any restrictions.
    - SQLite will be used in our project as the solution to database management due to its simplicity and efficiency. Our team does not intend to store large volumes of data or expect a mass number of concurrent user connections, so, a more robust option is not needed since we are not dealing with enterprise-level data management. For our usage and planned app features, a lightweight RDBMS that operates directly within the Android app will be sufficient since it includes the most essential SQL features and operations found from bulkier engines. Furthermore, being embedded will also simplify

the development and management of our app while maintaining data performance and integrity.

- Firebase (hosting and authentication service)
    - Firebase is a comprehensive cloud-based platform that provides a wide range of services and tools for building and managing mobile applications. Many backend services are provided such as authentication which helps developers build apps quickly, scale when needed, and maintain them for their software lifecycle
    - Firebase was originally developed by a company called Firebase, Inc. Before being acquired by Google in 2014 who has maintained it ever since. It is not open source but offers a free tier for their services along with several paid plans that are geared towards enterprises and larger-scale applications.
    - Firebase will be used in our project as it accommodates any project from small-scale developers to large businesses with a plethora of useful tools and services. Our team will use Firebase to host our Android application on Firebase servers to handle certain functions such as user authentication, client communication, and dynamic content without needing to set up and manage our separate servers. Furthermore, integrating with Firebase allows us to take advantage of its other tools and features as needed for software maintenance and scaling, if necessary, beyond the scope of this project timeframe.

Hardware Components:

- Laptop/Desktop (to develop application)
    - A relatively modern and capable computer, whether it is a laptop or desktop, is necessary to develop the project.
    - Android Studio is compatible with Windows, macOS, Linux, and ChromeOS operating systems which gives our team a lot of flexibility with the equipment we choose to work on.
    - Each team member will use their university provided computers or personal computers to develop, execute, and test the application thoroughly throughout this project's lifespan.
- Android device (to run the application or an Android Emulator can be used in place)
    - An Android device, whether physical or virtual, is required to run and test our mobile application.
    - Devices that run the Android operating system are manufactured by a vast variety of companies such as Samsung, Google, and more. Android Studio

provides an Android Emulator that offers many different devices to emulate which are cost-effective for development and testing purposes.

- Without an Android device, our team would not be able to execute and test our application during development, which is problematic and detrimental to our productivity and the app's quality. Furthermore, with Android devices being produced with varying screen sizes and resolutions, it is important to test a reasonable number of different devices to detect issues before the consumer experiences them.

Design Decisions:

- One of the most pivotal design choices our team had to make was the operating system of our app which ultimately influenced our technology stack. While we were considering iOS app development, the requirement to own a device running macOS for Apple's official IDE, Xcode, made it more difficult for all team members to contribute. Therefore, we decided to develop for Android since it is more accessible. Additionally, Android globally owns a greater market share of the mobile operating system market compared to iOS which will allow us to still reach a substantial audience.
- With the desired operating system selected, another important decision we were able to make for this project is our expected technology stack in which each major component is listed in a previous section.
- Since Jetpack Compose will be a major component in our technology stack in building the UI, our Android application will primarily use a Model-View-ViewModel (MVVM) architectural design. In short, the Model represents the data and business logic of the app, the View represents the UI elements, and the ViewModel serves as an intermediary between the previous two components to facilitate actions such as allowing the View to interact with data without directly accessing the Model.
- As we want users in our app to be able to message their friends and participate in activities such as playing games or answering questions, another design decision is that the app must be hosted on the web to facilitate its interactive and dynamic online features.
- In researching the mobile app market for similar products, our team found that there is nothing out there already that matches our project outline. The only remotely similar application is called BeReal where each day it will randomly alert a user to take a snapshot from their cameras to upload. To differentiate ourselves, it is essential to our design to implement activities other than taking pictures which so far includes brief games and intriguing questions.

Future Design Decisions:

- Are there other technologies undisclosed to the team yet that could simplify app development? How easy would it be to integrate it into the project at a given notice?
- What kind of user or system data will be stored in the SQL database server? What will the data model look like in terms of entities, relationships, constraints, etc.?
- How will the user interface look and feel? Is there a target theme or style?
- How will the application handle security and authentication?
- How will we handle errors in our systems? In the case of failure, how will we recover?
- How will we test our product and ensure its quality? Will there be a procedure for software quality assurance?
- How will the application handle performance and scalability?

Project Goals

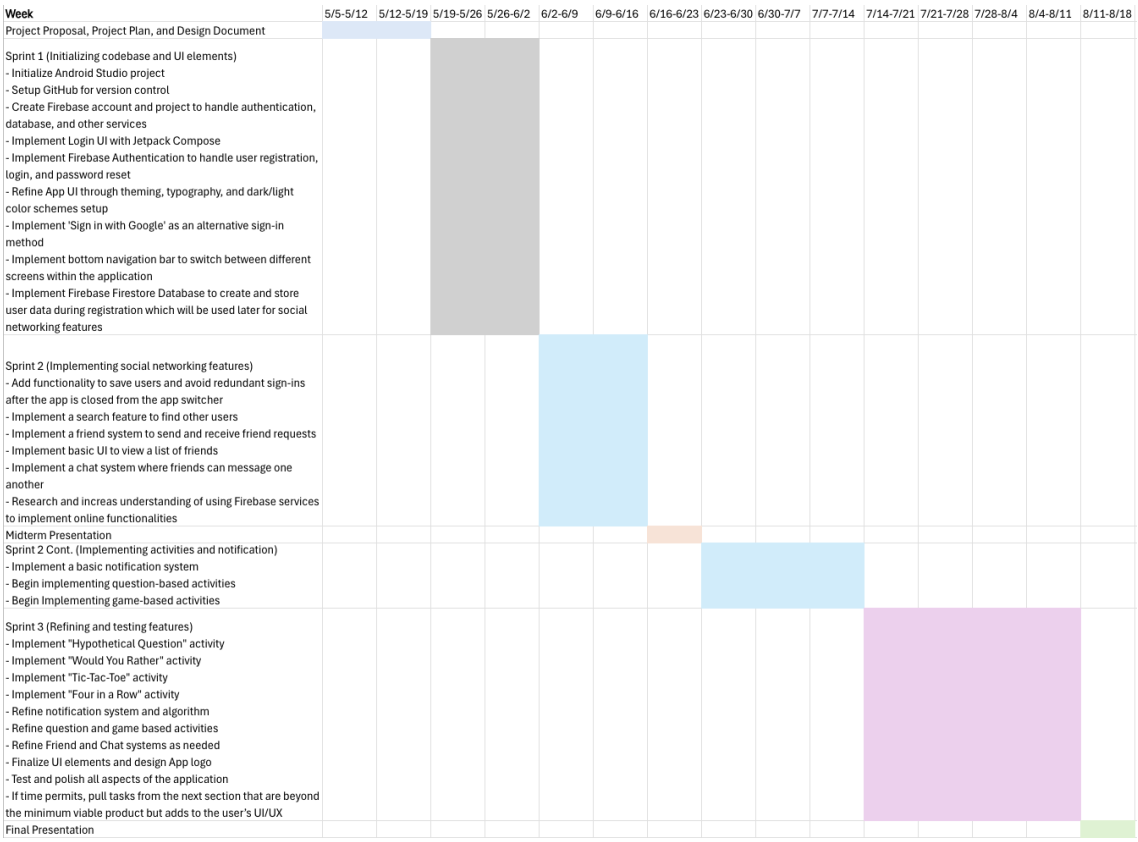
- The main goal of our project is to build a mobile application that helps users maintain and grow relationships with their friends, family, or significant others. Often in our lives as we get busy with responsibilities and obligations, it is difficult to actively reach out to loved ones or create new connections, therefore our team wanted to address this issue by developing a software solution.
- In terms of deliverables, this project will result in an Android-based mobile application that will be packaged into an APK file and submitted with its associated documentation.
- Another goal and focus for the team will be on creating a quality, functional, feature-rich, modern mobile app using established Android development tools and software engineering principles we learned through our studies at Wentworth Institute of Technology.

Additional Goals

- Depending on the duration of the app review process, getting our Android application published to Google Play (Store) will be important in determining its success as a commercial product through performance metrics.
- Once the app is published and available to the public to download, tracking total lifetime users, total active users, and most concurrent users over a specified time frame can be used to create an important baseline to measure current and future success.

- Furthermore, receiving positive app reviews from nonaffiliated parties (anyone other than friends, family, and Wentworth affiliates) is an additional goal of the project and would reflect its commercial success in the Android app market.

Updated Project Plan and Timing:



- The Project Proposal, Project Plan, and Design Document task has been completed early so the extra week has been reallocated to Sprint 1. Further updates will be made to the diagram when needed as the project moves along.

Unknowns

- Since team members have little to no experience with the chosen related frameworks and tools for the Android app development technology stack, if the learning curve is longer than expected then this could lead to the project not being completed on time.
- Lack of communication between team members could lead to a decrease in productivity and potentially missing or unpolished features.
- Poor code quality and software engineering practices could lead to reworks that may take longer than original code production.

- Major changes to third-party services, frameworks, and other software components could lead to extended rework times depending on the severity and dependencies.
- Physical damage to any hardware such as unforeseen breakdown of work computers could lead to loss of development time.