CSIS: 3126 – Design Project I

Project Requirements – SnapChat Clone

**Stakeholders**: Google (API), Microsoft (Computer Vision), SnapChat (Competitor)

**Project Team:** Jonathan Winter

**Project Requirements Update Log**

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Author | Description |
| 1.00 | 12/02/2018 | Jonathan Winter | Initial Version |

**Project Proposal / Description**

Many applications, such as SnapChat, store all of their user messages in large data centers. This kind of data storage can causs distrust by the user, companies to misuse the data, and leaves the data open to possible theft. This project will capitalize on the desire for a private and secure messaging social media application. The application will allow users to send texts, pictures, and video messages to one or more users. The application will store user data sent over the application for 24 hours before being destroyed. Facial recognition software will be implemented in the program to allow users to take funny pictures.

**Front-End**

The application will run on Android devices. The software will be designed for Android Ice Cream Sandwich and later to allow for approximately 99% of Android phone and tablets to run the program. Development will be done using the Android Studio IDE. Testing will be done with Android Studio’s built in android emulator and with my personal Galaxy Note 9 device.

**Back-End**

Google-Cloud and Firebase will allow for limits on-device data by saving JSON data in the Firebase real time database. Notifications can be sent with Firebase’s Cloud Messaging System. In addition to automated real time data synchronization, Firebase will also handle offline cases gracefully. Users can be authenticated through a variety of identity providers which will insure rapid development of the back-end service for my project.

**Project Scope**

Signup

Accepts user input from two text fields as an email and password, passes information into Firebase and secures the password.

Login

Accepts user input from two text fields. Successful credentials will allow for a user to login.

Camera Frag

The main activity screen is split into 3 fragments. The first fragment which is presented at default once a user has logged in is the Camera Fragment. This fragment has a logout button, a find pals button, and a capture button with the main screen being used by the camera.

Find Pals

The Find Pals button will prompt the user with a text field to enter an email address. A Firebase call will be made using the String to check all of the users with the given user input. If an email contains the String then it will be shown to the user with a button (Follow or Following to show if the user has followed already or would like to follow) If the user enters nothing, it will display all of the emails. Followed users will be able to show each other their story or send messages to each other.

Capture Button and Display Image Activity

The capture button will take a picture with the camera and display the image onto another activity screen. The following activity will have a button called send image. Note: All images are objects which contain a string and timestamp to determine if they are stories, chats, or if they need to be deleted or not. All images are destroyed after 24 hours.

Send Image Button

The send image button will bring the user to another activity that has a series of check boxes. The first checkbox is Story, the following are all of the users that the user is pals with. If the user checks story, the image will be attached to a string “story” which will let Firebase know that this is a storyboard image to attach to the user’s story for 24 hours. If a user is checked, the image will be attached with “chat” and Firebase will determine the userId and send the image to the correct user.

Story Frag

The story fragment can be found on the main screen by swiping to the left. The user will be able to see any stories that their followed pals have posted in the last 24 hours.

Chat Frag

The chat fragment has a button which can be used to refresh the page (In technical terms, the refresh button deletes all objects on the page, calls the server, and the server sends the chats saved on the server to the user.) If you click on a chat, a image will be shown that lasts for 5 seconds.

Logout:

This will log the user out, finish all activities, and close out Firebase connections.

**Potential Problems**

This will be the largest project I have ever built in my career as a Software Engineering Student. I may be underestimating the time it will take to create a project of this caliber. I’ve never worked with Android Studio or mobile application development before so it will take some time to get used to. Also, I will be working with Microsoft and Google’s Vision for facial recognition. This technology is complicated and will be challenging to implement in my project. Creating my own graphics will be a challenge as well because I do not have a graphics design background.