REQUIREMENTS

Sebastian Battle, Kyle Glaws

Introduction

For our capstone project, we will be building a video sharing app for Android phones. Users will record short 7-10 second videos, and be able to share them with other users. The purpose of having this hard limit on the length of the videos is to maximize interesting content within a short period. Users will also have the option to follow their friends and favorite app users, and accumulate a video feed which they can browse through. By simply tapping their finger, a consumer can quickly cycle through all videos from their followed producers. Ideally, popular content creators will upload particularly funny or amusing videos. Users can react to videos with a thumbs up or thumbs down (subject to change). Our ideal market are millennials, and we will target our app to a generally younger audience.

Possible Project Names

Views

Floors

Heights

Technical Details/Challenges

Android apps are built using Java, so we will be using the Android Studio IDE. In order to accomplish our design, we will also require the use of a database with a remote server from which to host the app's video content. We will need to connect multiple users to the database and load the videos that they record in the app onto the server. Also, we will need to maintain a cache of videos on the users local device to streamline the viewing speed.

We will require the use of a server, however we are already in possession of a tower with ample storage and processing power. Our server will be running Ubuntu with MySQL and Apache. We will rely on a web framework (likely Flask, other options specified below) for the server-side operations. In terms of the client, we will be utilizing the Android API for most of our goals, and we will acquire additional

libraries as needed. For transfering files to the server, we will be using the Java OkHTTP library. We will also rely on the phone's back facing and front facing camera, so we will obviously need to request the user for those permissions in order for the app to fully function. If the app is successful or if we are able to make significant progress, we will port the app to iOS.

Hardware

- Dell PowerEdge T320
    - 16 GB RAM
    - 4 TB Storage
    - 8 Cores
- Android phones
    - LG G6
    - Motorola Droid Turbo 1
    - Various virtual machines

Deliverables

Android App

- Camera usage
    - Android Camera API - camera2 class
- Video Streaming
    - Android VideoView class
- File Management
    - Android File class
- Server Interaction

        - OkHTTP library
    - Additional libraries will be acquired as needed

Server Software

- Ubuntu
- Web Framework
    - Python Flask Framework OR Express.js on Node.js (strongly leaning Flask)
- MySQL
    - Table structure TBD
- Apache Web Server

Timeline

March 5th - Basic video recording with UI, server online

March 12th - Video File uploading

March 19th - Working database (properly store videos)

March 26th - User database/ account creation

March 31st - Flesh out UI

April 7th - Video Streaming

April 14th - User interaction, following/video upvoting/downvoting (reflected in database)

April 21st - Continued work on user interaction

April 28th - Testing multiple users

May 5th - Finalize UI

May 12th - Deploy

Stretch Goals

Video Caching

Personalized User “Wall”

Framework Comparison

Possible Options:

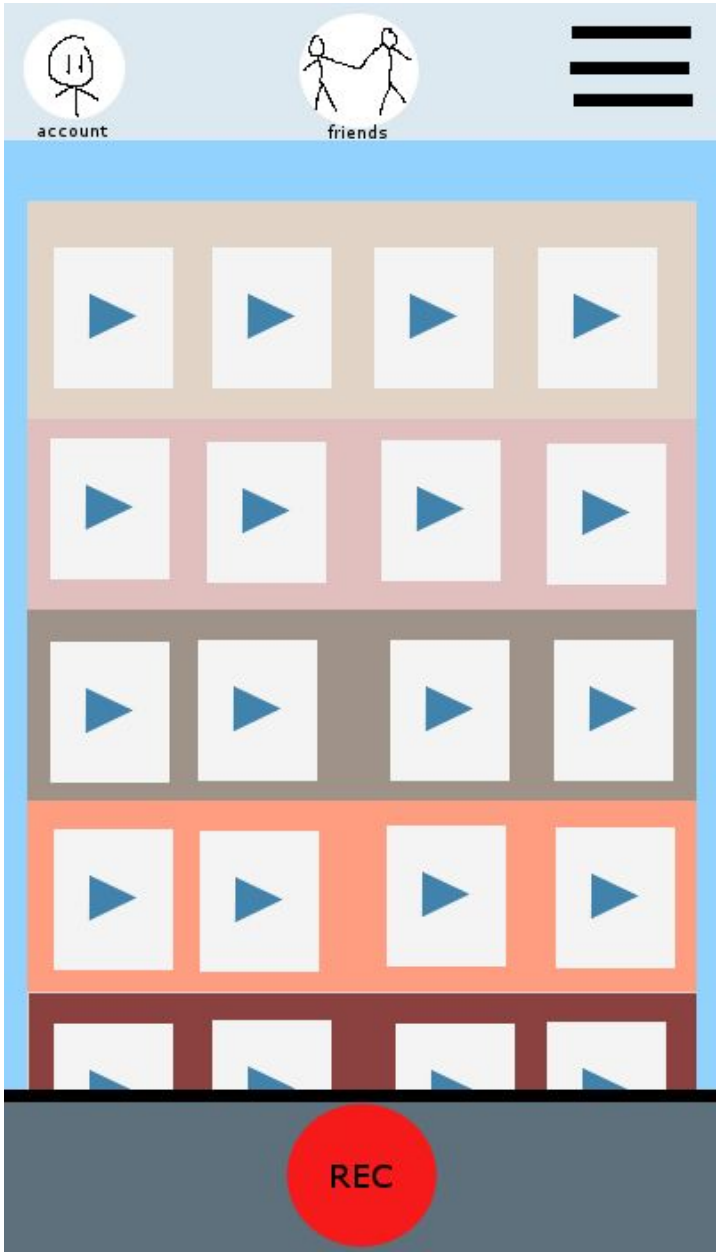
Flask - (Our current preference) Python microframework, easy to implement, ample documentation, previous experience with Python (Kyle)

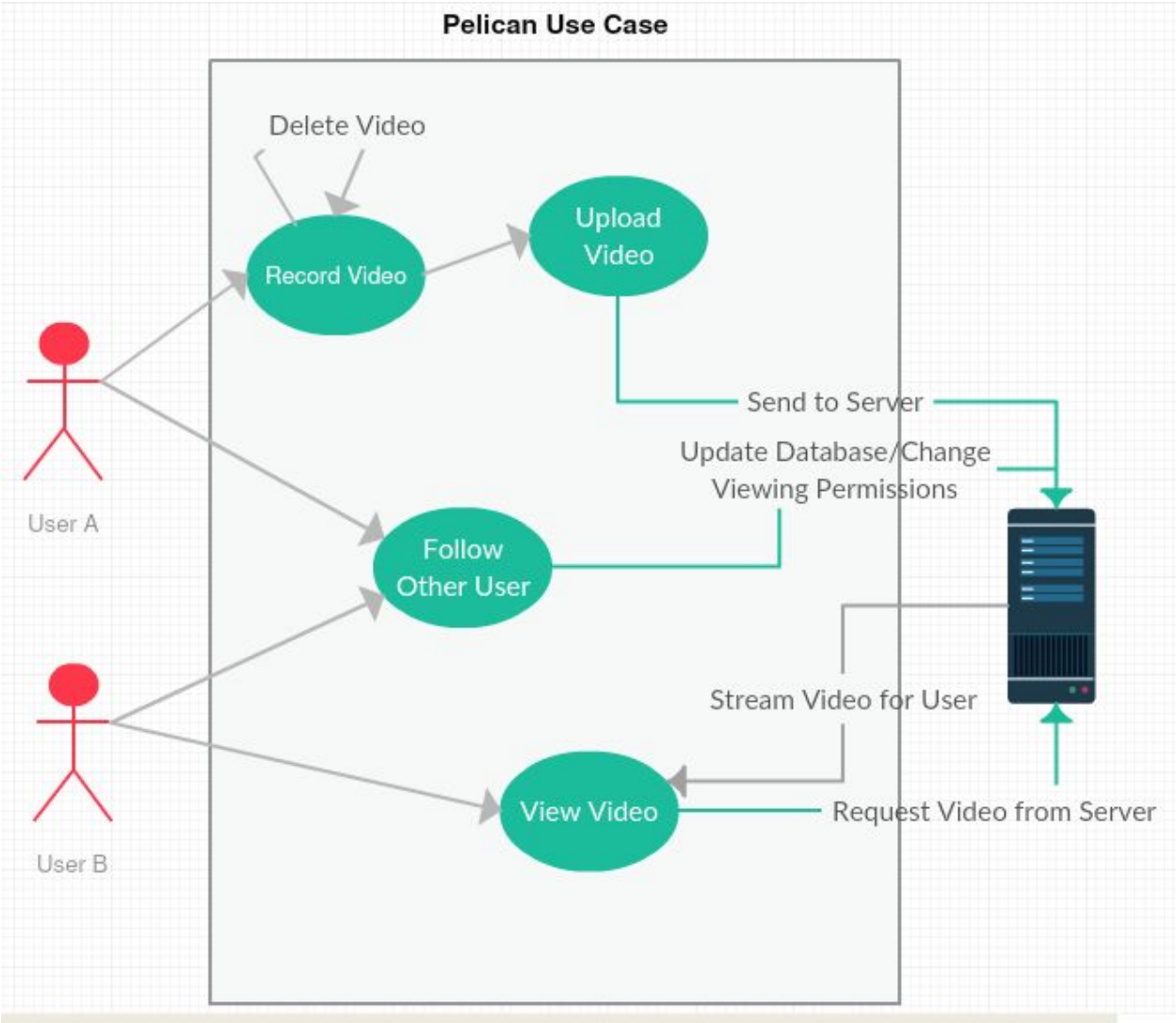
Django - Python framework, more built-in functionality, bells and whistles, more out-of-the-box ready to use.

Express.js with Node.js - JavaScript “un-opinionated” framework, previous experience with Node.js (Seb), easy asynchronous functions

Rough Mockup

The below design is a mockup of our potential user interface for the app. Easy access should be provided to the record button, and one should be able to easily see their “followed” friends. In our very early concept, the videos are presented in the style of a building, with each floor representing one of a user’s followed producers. Each window represents one of their uploaded videos that can be clicked on. Once clicked, the video will blow up to a full screen size, and one can simple tap their finger to cycle through the videos.


account
friends
REC



Use Case Description:

User A - Records video, has the option to delete and record a new video or upload to server where it will be viewable to other users who have followed User A.

User B - Follows User A/other users and is then able to stream videos belonging to that user from the server.

Server - When one user follows another, the SQL database is updated to enable that user to access the followed users videos. The server then streams those videos to other users when requested. When a video is uploaded, the video is associated with the user on the database.

Skills to Acquire

We will need to learn how to send files to a server using the OkHTTP library, and how retrieve them from the server. We will have to dedicated a large amount of time to learning the Flask framework, and also a large amount of time to learning the Android API. We will both need to brush up on our Python. We will need to learn the specific classes very well, in particular the ones involving the camera and videos. Also, we have made simple apps in Android Studio before, but this one will be much more involved and have a far steeper learning curve. Neither of us has worked with a database before, so that will be entirely new terrority.