DigiProf Requirements Document
For Version 1

Group 13

Introduction

We aim to create an app that bridges the student and professor when they cannot meet for office hours due to a difference in time zones or schedule conflicts. The main functionality of the app would be to send and receive video messages between students and teachers. This will involve an online media database that stores videos. The app should let users log in as professors or students. The student should be able to record video questions that are otherwise difficult to ask with text and photos.

With this app, the students and teachers have a way to connect efficiently without having to physically meet. People who are having trouble adapting to current teaching conditions due to COVID will be able to get a better learning experience through the app.

Intended Audience List

Students:
These students should be the ones who are unable to physically meet up with the professor due to time zone differences, scheduling conflicts. They should also be university students because in high school and elementary there is rarely a need to connect with teachers outside of the school environment.

Professors:
As mentioned in the previous point for students. This app should be more tailored to university students because there is rarely a need before university. Professors should have sufficient knowledge to reply to each student. Professors should have time to reply to the students.

Features/Functional Requirements

- Creating different users and login
- User verification via email.
- Recording a video and access that video locally.
- Able to select a local video and delete.
- Selecting an existing video to upload to database.
- Able upload to the database
- View the uploaded videos in the database with another device via website.
- UI & UX preliminary design, with inputs from teammates.

- Sent a video to a specific recipient.
- Create grouping for students and professors
- Create groups for different classes.
- Search ID for students and professors.
- Professors can delete videos from online database.
- View local video within the app.
- Have a collection of videos within the app.

- Able to sort video by date, length, views.
- Keeping track of the video replies with IDs, like a thread. Saving threads of video.
- Able to share threads via an link or ID.
- Saving threads of video to the local storage within the app.
- Access and edit threads and location of individual video.
- Create customize video thread for specific user.

Non-Functional Requirements

- The app must be aesthetically pleasing.
- There should be a general color theme
- The app should be simple and straightforward. There should be the shortest pathway to their desired function from the main menu.
- UX should have a smooth workflow. Smooth is defined as, users are able to logically access their intended features at any given point within the application, there should not be an instance where the user is unable to navigate or required to restart the app in order to access a certain feature.

Example Tutorials

Example 1: The user is working on a long and tedious math proof and needs help but feels that it would take too long to write it out in email and cannot make the professor's office hour.

i. The user opens our app and signs up for the first time.
ii. The user then records a video of their work and explains where they need assistance.
iii. There is a "Share" icon that the user clicks to send their video.
iv. The user chooses the recipient to send the video to.
v. The user can view their video in the database where all the videos are stored.

Example 2: A student would like to schedule a meeting with the professor for offices hours to ask questions

vi. The student is unable to do so because the office hours conflict with his other courses
vii. The student then turns to our app to connect with the professor with video.
viii. The student takes a video and sends it to the professor.
ix. Professor later replies to the student with his video answer.
x. The student successfully connects with the professor to ask questions.

Example 3: A student asked the professor a video question.

i. The professor had another student asked the exact same question before
ii. The professor just finds his previous video reply and sends it to the student.
iii. The student has their questions effectively answered, and the professor didn’t have to spend additional time answering repetitive questions.

Example 4: A student remembered he asked the professor a question before.

i. The student forgot the answer to the question and would like to review it.
ii. The student goes on the app and sees a video response.
iii. The student has his question answered again.

Example 5: The student asked an excellent question on the app that the professor hadn't even thought of.

i. The professor replied with a lot of details and a great explanation
ii. The student asked more excellent questions based on the response.
iii. The professor was even more surprise and answered with uttermost passion
iv. The professor wants to share this thread of video with all students.
v. He shared the link which included all the back and forth videos to all students in his class.
vi. All students in his class benefited from the interaction.

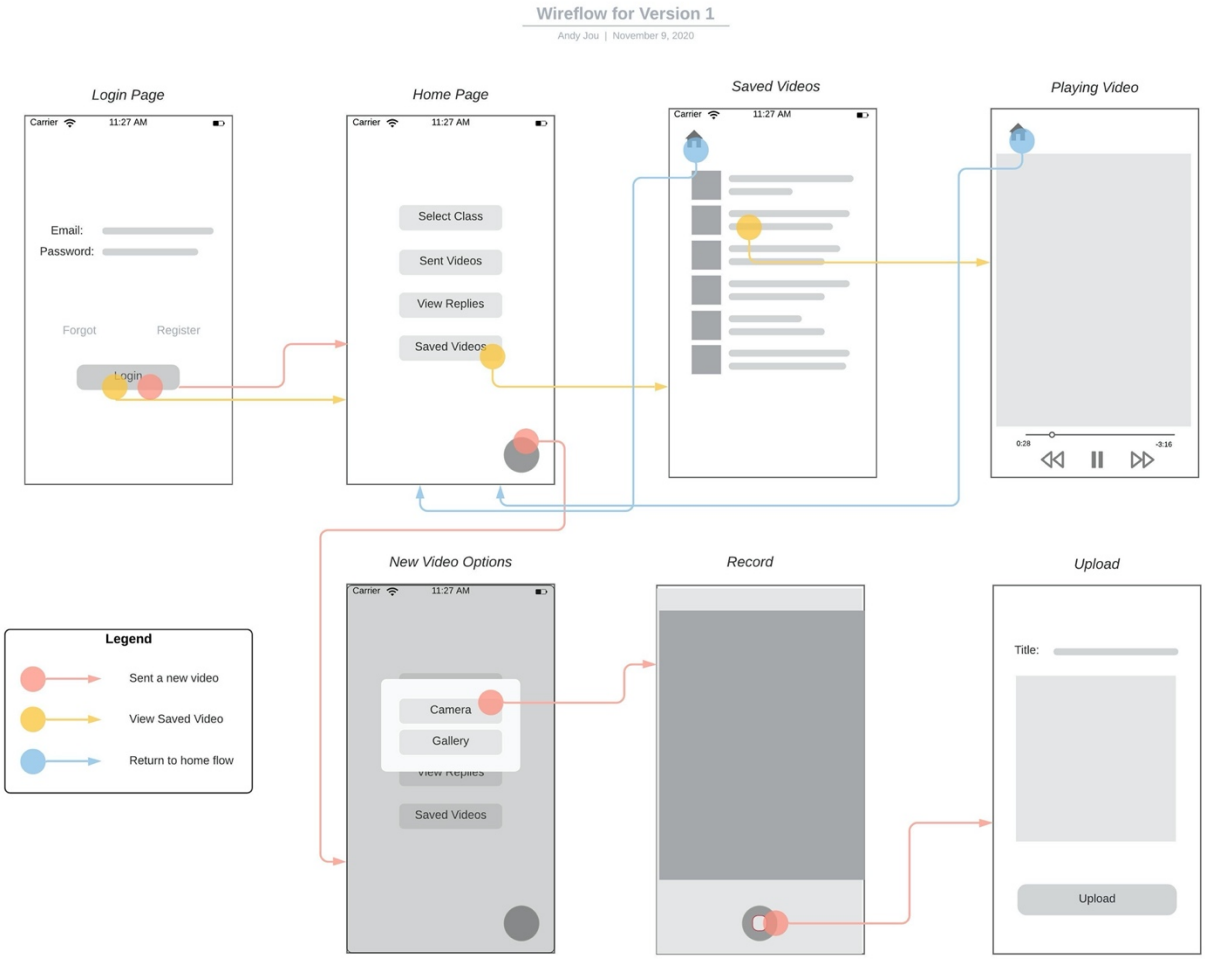

Figure 1. Wireflow diagram for example workflow for DigiProf app.

Figures

Figure 1: https://app.lucidchart.com/invitations/accept/520e4e2e-ca52-440e-9a82-e28b52d2db2d

Glossary

Student: People who are studying for a degree at a university.

Professor: People who have a Ph.D. certificate and is currently teaching a course at university.

Office hours: Professor's designated hours for additional questions.

ID: Identification, is used to associate a tag with a certain thing like a person or video to identify them with a unified tag.

COVID: Corona Virus Disease (COVID-19), Scientifically known as SARS-CoV-2.