While thinking of a project for our Mobile App Development class, my partner Brandon and I thought of some issues that we could create an app to address. One of these issues was a lack of a good online communication and collaboration space for students. While several services have tried to accomplish this, none of them have done so to the extent that we envisioned. Most other applications are either purely just a tutoring source, or just a study flashcard service, or just a website for rating instructors. What we wanted was to make an application that would combine these into one.

We believed that this issue was important for a multitude of reasons. Many students have a very hard time asking for help in person. An online app where students can type out their questions and communicate with peers via text would be beneficial to those who struggle with asking for help when class is in session. An online form of communication can put less pressure on shyer students and allow them to ask questions and get the help they need in a stress-free environment. Another reason why we believed this issue was important is that for students taking a course online, they have little to no physical contact or connections with their fellow classmates or instructors. By having a central online place to chat and connect with their peers and get valuable resources, students taking an online course could have a better experience and maybe even perform better in their online classes. Our goal is to create an app that leads to students feeling safe and successful.

Our initial idea for our app was to create a place where university students could communicate and collaborate. We wanted to create a controlled environment in which students could discuss classes and professors, share tips and resources, and get to know classmates. A few platforms have implemented these features, but none of them have combined all the features like we wanted to. A few examples include Quizlet, Brainly, and Rate My Professor. Quizlet offers study materials designed by students and professors for specific courses but has really limited themselves to a “study set” only kind of service. There is no sort of communication on Quizlet. Brainly provides tutors and Q&A spaces, but there is a lack of conversation as a community. Rate My Professor allows students to give a personal

review of their professors and their respective classes, but there is no way to verify that the student actually had the professor or even is enrolled at the school. Rate My Professor reviews also tend to be biased in a negative way, as students who did poorly or had a bad experience with the professor are more likely to leave a review as opposed to a student who did well or had a good experience. We wanted to create a space where students had to verify their enrollment at the school and in the courses they would communicate in and/or review. By making students verify that they go to the school, we would be able to keep the reviews, sources, tips, and other ideas shared on our app as authentic and veritable as possible. We were also thinking that we could make students review courses and professors on the app when they finish a class. That way, we can balance out the good and negative reviews if everyone is required to submit a review (as opposed to something like Rate My Professor, which most people only go to if they want to complain).

Once we had an idea in mind for our project, we began to think of what technologies we could use to implement our app and bring it to life. The first and obvious option was to use Android Studio, as all of this class's assignments use Android Studio so we would have the most familiarity with it as opposed to other potential mobile application development tools. This is where all of the main development would take place. The next obvious choice was to use GitHub so that we could share our code with each other and collaborate. GitHub makes it easy for developers to work on the same project without the risk of altering each other's code thanks to the branches feature. Brandon would do most of the backend development on his own branch and I would do most of the frontend development on my own branch. For the frontend-oriented content, we thought we would use Material Design and Figma. Material Design is a set of developer tools designed by Google for implementing the best practices of user interface design. We primarily used Material Design for their array of system icons, which we found useful to make our app look sophisticated and appealing to the eye. Almost all of the icons in our app's user interface come from Material.io, the website for Material Design. Another design interface tool we

thought about using was Figma. Figma focuses on user interface and user experience, allowing designers and developers to collaborate and create beautiful interfaces for their apps. At the time of writing this paper, we have only used Figma to create our mockups. However, we plan to use Figma later in our project's timeline to polish our application's user interface and make it really stand out. For data storage in our app, we would use Firebase. Firebase is a development tool created by Google designed to create mobile and web applications that require databases.

Development of our app began with mock-ups. I designed a sign up page, a login page, a home screen, and a discussion forum screen. While creating these mock-ups, I started thinking about what color scheme we would want to go with for our app. I even used some color theory to make the final decision. A lot of technology companies and social media platforms tend to gravitate towards using blues in their color schemes for their apps, so I decided to stay away from that. Color theory suggests that people associate the color green with prosperity, success, safety, and approval. These values resonated well with our app's idea of promoting student success and a safe space to collaborate, so I thought that green was a good choice for our app's color scheme. For these reasons, I decided on a nice lime green for our app's main color.

Once we had our designs in mind, we began the actual coding part of the project. Neither Brandon nor I have ever used Android Studio before this class, so a lot of the initial process involved lots of reading of the Android developer guides online. We began the actual coding part of our app by choosing to use the fragment activity preset in Android Studio. We thought that fragments would be the most useful way of navigation for our app since it would have many different screens that the user could go to. We made a simple fragment activity so that we could explore how navigation worked with fragments. Once we got this working, I started designing the basic user interface on Android Studio, starting with some simple buttons. Eventually, I began designing the home screen's user interface as well. I had some problems figuring out constraint layouts. I find them confusing to work with and cannot

quite get them to work, so sometimes our app's user interface moves depending on screen size. I have been looking into linear layouts as an alternative, as I find these much better to work with.

Once we got simple fragment navigation working, we ran into some backend issues. We realized that programming solely with activities is not common coding practice, so in order to make our code more "correct" per se, we tried working with both activities and fragments in our project. We were unable to get data to be passed from fragment to fragment. To remedy this, we tried using template code to see if that would help. It did, but only to a point, because when it came to adding our own extra coded that we needed for our app, it became difficult for us to understand how to take the template code and work with it and around it. Brandon tried removing the boilerplate code so that only our 100% original fragment code would remain. After following countless tutorials and different methods of implementing fragment navigation, we were unable to get it to function at all, let alone the way we wanted. We have used over 3 types of tutorials for navigation, including the official Android Studio developer guide, various channels on YouTube, and other online guides, all to no avail. This is the biggest roadblock in our project at the moment, and the chief reason why we are currently unable to move forward.

While we are not where we originally aimed to be in our projected timeline for this project, Brandon and I have learned a great deal in the process of making this app. We ran into some problems with Android Studio not compiling our project properly, issues with fragment navigation and data passing, and issues with constraint layouts. These problems have pushed our project timeline back, and unfortunately with other course requirements and projects, we were unable to get to the point where we initially wanted to by this time in the semester. Perhaps we dreamed a little too big for a semester project and took on more than we could handle. But in the process of trying to make this app, we have learned more about Android Studio and the design process than we knew when we first entered this class. Though we most likely will not have a fully finished and working app at the end of the semester,

we at the very least put in our best effort and learned about the mobile app development process while working on this project.

## Links to all sources used so far:

sources used for 11/3 progress report:

- https://www.youtube.com/watch?v=Enwgwh5ZGy4&list=PLQ9S01mirRdV0bX9i_-4c1cpr7F8ciS8I
- https://developer.android.com/guide/navigation/navigation-principles

sources used for 11/10 progress report:

- https://www.youtube.com/watch?v=Uk5TnFL7jh4&list=PLlcnQQJK8SUjuzpRx0U-VEUzhmJD7vGbO
- https://git-scm.com/book/en/v2/Git-Branching-Basic-Branching-and-Merging

sources used for 11/17 progress report:

- https://firebase.google.com/docs/auth/android/email-link-auth?authuser=0
- https://firebase.google.com/docs/database/admin/save-data