

Company Mentor: Citrix

Team Name: Citrix are for Kids!

Team Members:

O'shea Anaya - osheaanaya@live.com

Shadee Barzin (Scribe) - shadeebarzin@gmail.com

Andrew Ferguson - aferguson@umail.ucsb.edu

Brendan Murphy - bmurphy@umail.ucsb.edu

Peter Werner (Team Lead) - peterwerner@gmail.com

Study Together: Vision Statement

Project Title: Study Together

Google Group: <https://groups.google.com/forum/#!forum/citrix-capstone>

Github Repo: <https://github.com/peterwerner/citrix-capstone>

Problem Statement

Throughout their high school and university experiences, students spend much time collaborating with peers, teachers, and TAs, as well as participating in group projects, study groups, and review sessions. However for many students, physically getting together in a group setting can be a challenge. There are many non-traditional students such as those who work full or part-time, parents, and disabled people, who find it difficult to set aside the time to travel to another location for a group meeting when they have many other priorities. Also, finding a physical meeting space for a group can be difficult, especially during high traffic times such as midterms and finals. Currently, the best solutions for remote collaboration are a combination of Skype or Google Hangouts and Google Drive, but these are not integrated with one another and are also not focused on student study groups.

Project Outcome

The aim of this project is to make studying together significantly easier. Study Together is a platform that combines audio and video meetings, screen sharing, and document editing to make group collaboration possible without being physically near the people in the group, and make studying even better than traditional study means. Not only will it improve studying when separated, but it aims to improve conventional study meetings as well. Study Together allows students to enter their school and courses for the term, and displays a list of their classmates as well as current study groups. The users are then able to create additional study groups or join an ongoing one; the aim of this feature is to make finding a study group simple. It also provides audio and video services as well as chat to make speaking to one another easier. Screen sharing and file sharing are also available so users can easily send information to others in the

group. It integrates with Google Drive to cohesively integrate document editing into the application. To make remote collaboration even better, Study Together has built-in smarts that automatically bring up web pages, articles, and images relevant to the topic being discussed in the group meeting. Study Together records each individual study session and creates a written transcript as well, both of which become available to the group members once the session is complete.

Milestones

Our first milestone is simply to fully specify and design the project. This includes defining a basic look for the user interface, deciding on necessary and stretch features, and figuring out which technologies we will be using. Learning about any new technologies or languages that we are unfamiliar with is also a key portion of the first milestone. This will evolve as we meet with our mentors. The following milestones outline our current plan for implementing this project:

For our first iteration, we want to support a user account system and allow users to create and join rooms with other users. Users should be able to communicate via text chat. This implementation will test to see if we can connect multiple users to a joint session, and once we have this done we can continue to build further.

In our second iteration, we plan to allow study groups to persist after the creator and other users sign out, maintaining chat history and allowing users to join the group later on. We will also start implementing video and audio chat.

Next, we plan to allow users to share local files and files hosted on Google Drive via text chat. We will also allow students to find study groups based on which courses they are taking.

Our stretch goal is to use natural language processing and speech recognition to provide voice commands and instant search suggestions based on the current conversation.

Implementation

Implementation details will change quite a bit as our project specification evolves, but we tentatively anticipate using the following technologies. We plan to use a NodeJS server and Express as the framework for our web application, with ReactJS in our front-end. Audio and video chat will be achieved using WebRTC (peer-to-peer communication). To integrate Google tools, allow file sharing, and collaborative document editing, we anticipate using multiple Google APIs. We also will need a database to store user information, session information such as persistent chat, as well as store video session recordings and written transcripts. The database we plan to use is Firebase since it is fairly quick to pick up and provides a nice user interface. To do our natural language processing for the built-in smarts, we plan to utilize NLTK or Stanford NLP APIs. We will also use the Google Speech API for speech recognition, which is required for the session transcript.