

Team Number: 101-7

Team Name: Watery Ducks

Team Members: Joe Rickard, Aiden Prohaska, Nick Julander, Matthew Spallas

Application Name: Simple Store

Application Description: Simple Store will allow groups of users to share documents between them. User accounts will be able to upload documents, and then choose which other users, if any, will also be able to see and/or edit the documents.

This will be accomplished through either a well featured CLI, or a browser based UI, both leveraging HTTP requests to interact with the file store.

Vision: For students who need to share files between their colleagues. The Simple Store is a cloud based storage system that allows users to upload their files to AWS and access them anywhere. Unlike Google Drive our product is for the simple mind, we don't have a bogged down UX, in fact, we don't have one at all.

Version Control:

<https://github.com/joerickard/csci3308> (folder that contains the three repositories)

Development Method:

We will be using an agile/scrum development method. We will be assigning 1 week sprints, at the end of each week we will come together, share what we have learned, and assign new tasks. The sprints for the first week are individual so that we can create the foundation of our Simple Store cloud storage web application.

Communication Plan:

We will be using Slack to communicate, and in person meetings on Tuesdays at 5-7pm. Through Slack we will share research information, and links to resources that aid us in development. We have two channels, one for front-end and one for back-end, and a general channel for any conversation that the whole team should be in the loop on. In person, we will be able to work as a team to answer problems that arise.

Proposed Architecture Plan:

For the front end of our application we intend to use a CLI. To do this we will use python and the argparse library. This will also allow us to leverage the requests library to communicate with the backend more easily.

For the back of our application we will be creating a back end database to store the user key, and their associated file. With a compression algorithm, we will compress files to be uploaded to the back end database. If a user wants to access a file, we will decompress the file and provide it to the user. Also, we may encrypt the files that are uploaded to the database. We will launch the application on the public cloud using AWS EC2 or AWS Beanstalk. Alternatively, we may use AWS S3 in combination with AWS Lambda to respond to changes in S3 buckets, etc.

Meeting Plan:

We will be meeting Tuesdays at 5pm, scheduling through Slack and we will be reserving different rooms (depending on availability) to meet in person.