EECS 581
Initial Project Description
10/17/2019

Team Number: 8

Team Members:
Seth Peterson, Moe Zeid, Blake Jordan, Austin Wildman, Bunty Dey

Name:
Snap Shot **???**

Synopsis:
A Chrome extension that allows users to screenshot a section of the screen and copies the recognized text to the user's clipboard.

Description:
The main use case for this Chrome extension will be coding tutorial videos. Take this scenario for example: a student is watching a YouTube video about a coding topic. The student wants to copy the code directly to their clipboard, but cannot. Therefore they spend time switching between an IDE and the video manually typing the code. The Snap Shot extension will solve this problem. Outside of YouTube, this service can be used in any number of contexts where code is displayed in a non-text format.

The end result of the project will be a Chrome extension that, when activated, will prompt the user to select a section of their screen. When completed, the extension will convert the image to text using Base64 encoding. Then the text will be sent to the text recognition API. The result of the API call, if valid, will be applied to the user's clipboard.

Milestones:

Fall Semester:
1. Determine a language best fit for the requirements of the project and set up a basic development environment **(September 25th)**
2. Put together a high level diagram of how the application receives a request and sends a response **(October 2-5)**

3. Put together plans for the individual components of the application i.e the front end, the image capture, the image compression & send, character recognition, and more **(October 2-5)**
4. Decide on a database service and if applicable, determine a budget for using that service **(Mid October)**
5. Solidify plans and begin working on the individual components **(Early to mid November)**
6. Implement the Javascript frontend **(Mid to late November)**
7. Implement the image capturing and sending capability **(Mid to late November)**
8. Implement the image to text recognition capability **(Late November to Early December)**

Spring Semester:

1. Implement a Python script that takes a picture as input and returns the recognized text as output **(early spring)**
2. Implement basic screenshotting functionality within the Chrome extension **(early spring)**
3. Port Python text recognition script to an API **(early-mid spring)**
4. Connect the API and the Chrome extension **(mid spring)**

Budget:

Computing resources needed include access to Intellij IDEs and probably a database as well. Being students, we can get licenses for most IDEs so the only cost that may come out of this project lies in database leasing.

Work Plan:

Image-based text recognition API: Seth Peterson, Moe Zeid
Chrome extension: Blake Jordan, Austin Wildman, Bunty Dey