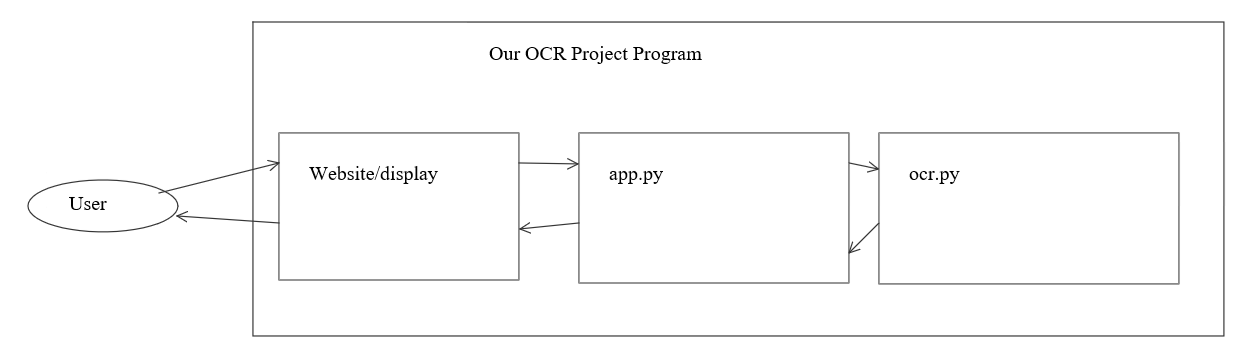
Design Documentation Project 3

1. Identify the Design Paradigm, explain
   1. Service oriented design - Our goals for project three are to make a website that can read words from an image file and transfer the words in the image to text that can be copy and pasted on the computer. We want our website to be able to read text and handwriting, in other words, we are making an OCR (Optical character recognition) program. We do plan on adding more features such as translation to this program for project four. Nonetheless, our initial intensions are to automate away tedious tasks for our users while having a focus on photo recognition. Our website supports the majority of picture file types such as pdf, png, jpg, jpeg, and gif. By having a website, it is easier for users with no computer science background to interact with our system. With the easy setup, our users should be able to navigate around our website with ease. This satisfies the black box quality of service-oriented design for our users. Since our users don’t have to be aware of out service’s inner working to use our websites to use our website.
2. Describe Software Architecture of Our Prototype
   1. So far we have two main components in our program, the logic for our OCR, and the user interface of our OCR. The logic portions of our program are in a ocr.py file, and our user interface is in a app.py file. Other than that, we have a index.html for our website and some formatting files for formatting. The program starts with the user uploading a file on our website. Then, app.py passes the image file to our ocr.py, our ocr.py reads the image and passes the read text as a string back to app.py. In our final step, our app.py displays the text on our website for the user to see. I believe our project has the pipes and filters architecture, since we out system is a form of transforming input data through a series of computational components (the image) into output data (text). It has a rather simple pipes and filters architecture, but a pipes and filter architecture, nonetheless.
3. Use one or more UML to Design
   1. 
   2. Use Case Diagram - Our user interacts with our programing by using our website, app.py receives that photo and sent it to ocr.py. ocr.py process the imagine into text and sent the text to app.py. app.py display the text on our website for the user to see.
4. Identify Design Patterns and Explain how to apply them
   1. Interpreter (Behavior) – We have an image text interpreter that reads from an image then interpreted it into printed text.
   2. Chain of Responsibility (Behavior) – User upload an image, app.py receive the image, pass it to ocr.py. ocr.py interoperate the text and pass it to app.py to display back to the user on our website.