Kevin Altermatt

Josh Mandell

Touhedul Islam

RSEG 176

Group Assignment 1

**Introduction to Project:**

We initially set out to build a web-based business tool. These tools exist in many different formats, such as EasyBib, online document format, translation the list goes on. The online tool that we initially purposed is a document conversion app. Document formats are vital to organization, meta-tagging, and use of the document. The specific use case we targeted is translating a text document to a pdf document for storage. Converting documents to pdf allow the document to tagged with meta-data, stored, and retrieved very easily. In this format one could also add other needed business functions such as watermarks, compression, or harmonized formatting.

In order to accomplish this task, we initially started with an ideal concept framework. Where a front-end user interface acts as a medium between the action of a user uploading and tagging (with meta-data) the document and the back-end manipulation of the document. The back end converts the document adding a watermark and stores the document on AWS. The user can then retrieve the document using the meta-data or document name in the new converted form.

This architecture purposed for this solution is the front-end application API connected with a AWS Lambda function to initiate the business functions and Amazon DocumentDB to store the document. This architecture would run in reverse for document retrieval.

**Issues and Troubleshooting:**

One issue that we found in building this application was a successful front-end application that could receive the document. We started working with Amplify but there were significant issues when trying to attempt to push a basic html code written to the console. When this was unable to be resolved other ways were attempted. One such way was creating a React application in Visual Studio however although the code compiled without failure it did not load in the webpage. We had also looked at using Amplify CLI instance instead but were met with another issue that we had found is crashing CLI instances on connection attempts.

The biggest hurdle was that the issues slowed our teams progress from developing the connected architecture in AWS. However, we had significant progress in developing the Python application to perform the business functions as well as some success in connecting Python through a EC2 Instance for future front-end application development.

**Application Elements:**

The final application is meant to consist of two main elements: a front end through which a user can upload files to be converted, and a back end which performs the conversion and file manipulation, and passes the finished files back to the front end to be returned to the user as a download. The back end code has been written in Python to perform file conversion from PDF to DocX or RTF formats. It is possible to perform other file manipulations as well depending upon the user’s needs, which can be added to the application later. The front-end will be created using an AWS framework, but as of this writing the front-end is yet to be fully up and running. All that is necessary is to attach this code to a system which can handle file uploads and downloads.

**Forward Looking:**

We would like the application in the future to have an improved interface and additional functions (such as more manipulations that can be done on files, and actual support for more types of file conversions and formats).