

Criterion A – Planning

Defining the Problem

My client/advisor *Mr.S* is the president at *P. Incorporated*. *P. Incorporated* provides drafting and document control support to clients in the field of pipelines, facilities construction, and/or fabrication business. During our first discussion recorded in Appendix A, Mr. S told me that he often receives requests from his clients to capture **information tags** (containing information regarding equipment and instrumentation) from drawings and some other engineering documents in PDFs. They normally follow a **uniform** convention of tagging standard. These information tags could be embedded all over the documents.

He told me that there were many issues that arose when this process was manually undertaken in the past. Firstly, he described how **laborious and inefficient** the process of capturing the tags manually could get¹. Wasting time and resources on this process could get expensive, especially when larger documents are used. Furthermore, he suggested that there was a potential risk of losing accuracy and missing tags due to human error.²

To overcome this issue, *P. Incorporated* has approached me to develop a solution that will capture the tags using a software that will eliminate the intermediate step of capturing the tags without any manual intervention. It will provide an editable, computer-text output that will be delivered as a **CSV** file³ to a desired location. This would be the best solution as it provides tremendous time savings as well as the accuracy of capture and makes the process more efficient for *P. Incorporated* itself.

¹ Appendix A (Mr.S, 2020, L. 12)

² Ibid (Mr.S, 2020, L. 13-14)

³ Ibid (Mr.S, 2020, L. 27)

Rationale for the Solution

After consulting the client, Mr.S and I decided that a computer-based application would be best suited to alleviate the problem. I decided that the program itself will primarily utilize **Optical Character Recognition (OCR)** pattern recognition inside the document to efficiently read the tags on the drawings and transfer them to computer-generated text. The program will sort tags based on pre-determined requirements and store them in the desired user-chosen external file. All of this will be done on an easy-to-use **Graphical User Interface (GUI)**. I have decided to use Java to complete this task for the following reasons:

- *It is a language that I am well-versed and confident in*
- *Java is known for its object-orientated programming, which means it will be good for processes such as **encapsulation***
- *Java has one of the largest and most diverse APIs of all the programming languages*
- *Java is **platform independent**, meaning that code only has to be written once before it is run on all other devices*
- *Java has detailed documentation support for quick reference*
- *My preferred IDE (Java **Eclipse IDE**) is available in the Java language; it provides a renowned debugging tool that will help immensely during development*

Criteria for Success

To determine that the solution is successful, it must accomplish the following:

- *Present the user with easy-to-find and use commands; have clear instructions*
- *Give the user some options to edit the image (ex. Crop, resize)*
- *Pre-process image data to make it more readable (convert inputted image to grayscale, filter out unwanted areas)*

Candidate #: hpg293

- *Scan the file and find all present tags*
- *Provide a CSV file output*
- *Outputs the contents in an organised manner*
- *Inform the users of any potential errors during operation*
- *Have code/ideas that could be modified in the future*

423 Words