Criterion A: Planning

Defining the problem

Client Mr Lam is my father, and he works in accounting in an airline company. During Covid time, his company couldn't pay in time for many services. Because of this, he needs to take a lot more time than before to manually check through his Oracle's invoices logs to see which invoice is overdue and which invoice needs to pay first and fill them in Excel reports. This takes him a lot of time, and he complained about this to me on a video call in May 2021. He said he wanted a program that could help him record invoices and automatically export various invoice reports to various platforms like Excel, PDF depending on pre-set filters.

As he was saying, I thought that I could create that program for him through my Internal Assessment. A few days later, I told him that I wanted to create the program for him, and Mr Lam accepted my offer. He then sent me an email detailing important functions he is looking for in the program, images of how Oracle kept track of his invoices, ways and formulas to export them to Excel files in order to give me an idea about the program he wants me to create. (Email, 2021) in Appendix.

In order to understand more about the program for Mr Lam and the success criterias, I scheduled an interview with Mr Lam. (Interview 1, 2021) in Appendix.

Rationale for Proposed Solution

My proposed solution for Mr Lam would be a desktop application because he said that he has to input invoices into the program both when he doesn't have access to the Internet and when he has access to the Internet and also because he can only works with computer to deal with the invoices as this is his company's regulation. A website needs Internet connection, and cannot meet this need by Mr Lam. Similarly, a mobile application would also not meet Mr Lam because he does not use mobile phones to input his company's invoices.

For this desktop application, I will use Python to develop its back end, Kivy and KivyMD to develop its front end. I justify the reasons why I choose to do as following:

- Python can interpret the source file at run time, removing the need for my computer to compile the application before executing it. This allows me to quickly code the application little by little bit and get quick feedback from Mr Lam, which

then allows me to make sure that my program is meeting his success criterias and save me time from further developing something that he doesn't want to have in the application.

- Python's syntax is simple, not complicated, which helps me to spend less time coding basic functions and spend more time coding more complex functions, which are needed to be able to meet my client's requirements.
- Python provides various libraries that I can import into my program that can help me develop functions that meet my client's requirements. For example, I can import the xlsxwriter library and be able to then have a back end framework to export excel files from my application, meeting one of my client's requirements of being able to export invoices into Excel files.
- Kivy allows me to easily create multiple front end designs/screens that can show many of the different functions that my client requires.
- Kivy and Python structure is similar, and this helps me to avoid or minimize mistakes while developing the program both in the front end (with Kivy) and in the back end (with Python). For example, each front end design with Kivy can be built by a class in the back end with Python, so if I make a mistake in one class, it will not affect other classes .
- Kivy allows me to add in multi-touch screen functions (such as scrolling up and down), which would be crucial to meet the client's need to be able to store and view many invoices. An unscrollable screen can only display a limited amount of invoices, and the client doesn't want that.
- All front end widgets of Kivy are custom-drawn, which ensures the front-end of my program for the client to be the same when he uses it on his computer as when I created it. This means that I don't need to worry about the usability of my program for my client after I made sure that the usability is good during alpha testing on my computer.
- KivyMD, a collection of Material Design widgets for use with Kivy, provides a much more attractive front end GUI (Graphical User Interface) for my program, which boosts the usability of my program for my client. Usability is extremely important for my program because if it's hard for my client to use the program, there won't be any incentives for him to use my program (even if it fulfills all of his requirements) in the first place.

Stating Success Criteria (checked and agreed upon by the client)

1. The program will have a login and logout screen.
2. The program will let multiple users create their account and passwords, storing those passwords in hashed (the client has employees who work for him, and he said he wants them to use the application to add/edit/remove invoices for him as well. He also wants different employees to use the program with different accounts because this holds each employee accountability when they add/edit/remove invoices in the application)
3. Let the client add/edit/remove invoices.
4. Program lets the client input fixed data columns about the Trading Partner and pre-stored data in these fixed data columns for every subsequent invoice added to the program after the first time, removing the need for the client to add the same information about a Trading Partner for multiple invoices with the same Trading Partner.
5. Let the client filtered/search invoices based on required data columns such as Invoice Number, Trading Partner Name, Invoices added date, Invoices date, Payment information into tables that are viewable on the screen
6. Let the client export the filtered/search invoices tables (the invoices don't need to include trading partner data columns) to reports in Excel format.
7. Let the client create 2 main types of reports: Overdue Invoice Report and Invoice Payment Schedule Report and export them to reports in Excel format.