Final Internship Report

Submitted by

Chia Yu Ying
1004609
Engineering Systems and Design (ESD)
Year 3
21 August 2021
Undergraduate Business solutions Intern

Submitted to

Oversea-Chinese Banking Corporations (OCBC)
Group Finance- Treasury Financial Control & Advisory- Business Solutions & Control
Leo Lin Zhi
Vice President 3

SINGAPORE UNIVERSITY OF TECHNOLOGY AND DESIGN

Acknowledgements

Foremost, I would like to express my sincere gratitude to Leo and Jean for giving me an internship opportunity at OCBC.

A debt of gratitude is also owed to my supervisor Leo for the continuous support of my internship OCBC, for his patience, motivation and immense knowledge. His guidance has made my learning at OCBC more enriching and enjoyable. I could not have imagined having a better mentor for my internship journey in OCBC.

My sincere thanks also goes to the team, for offering me help and guidance in times where I needed, I have also learned a lot throughout from the team for the past few months of intern in OCBC, and it has been a great pleasure working with them.

Executive Summary

This final internship report records the work I have done thus far as an undergrad intern at OCBC from 20 June to 13 August 2021.

Similar to the first five weeks of my attachment, my allocated tasks from week 6-12 of my internship also centered around tasks that aims to increase "Business As Usual" (BAU) efficiency, by automating daily mundane manual tasks, by using Python and Qlikview. On top of that, I have also converted "unstable" automated tasks which runs daily, to a more "stable" and reliable program using Python script.

To automate a task comes in mainly 2 phases – one of which is the development phase where a working code is written out and executed, the second phase is the user testing phase whereby the code will be tested out by the user at the designated daily scheduled timings, and feedbacks on how the automated process can be improved will be gathered from the user. Through the feedbacks, I proceeded on making the necessary amendments and ran through phase 2 once again until the user is satisfied with the code that I have written. During the process of automating tasks, I have met and overcome many challenges. The challenges that I have faced through the process will be documented in this report.

Besides automating tasks using Python, I was also given an assignment to automate a task using Qlikview (QV). Through the process of automating the task, I was also given an opportunity to design a dashboard for Qlikview which combines and converts existing manual processes to a more stable system.

Table of Contents

1. Introduction

In the first five weeks of my attachment, my allocated tasks centered around the creation of Macros using Visual Basic Application (VBA) in Excel, to increase "Business As Usual" (BAU) efficiency. The Macros that I was tasked to create aims to fully automate some of the essential but mundane manual tasks that needs to be completed daily. The creation of the Macros comes in mainly 2 phases – one of which is the development phase where a working code is written out and executed, the second phase is the user testing phase whereby the code will be tested out by the user at the designated daily scheduled timings, and feedbacks on how the Macro can be improved will be gathered from the user.

One of the main challenges faced during that period was having to self-learn VBA and write the Macros at the same time. On top of that, even though VBA was relatively easier to pick up as compared to other languages, it was still a challenge for me to constantly find ways to find the most efficient way to write the Macros, to make it as flexible and user-friendly as possible.

The experiences of the first six weeks prepared me for the work expected of me during the rest of the internship period, as I continued to improve the Macros that I have written in the first five weeks. At the same time, I was also given new responsibilities – which includes converting the "unstable" Macros to Python script, reduce daily manual tasks by migrating the MS Access Program to Qlikview program and designing a dashboard for data presentation by using Qlikview.

2. Work Experiences

2.1 Python Script for Daily File Transfer

Background: New data and files come in daily and hence, many files need to be copy and paste to new locations to keep things organized and ready to be used whenever needed. Currently these file transfers are automated using MS Access program. However, as the number of files to be transferred increase over the years, file transferring using MS Access has become more and more inefficient.

My first task aims to:

- Increase BAU efficiency by converting the MS Access program for file transfer to Python script.
- Allow flexibility for different BAU controllers to customize
- Script to be scheduled to run hourly everyday

2.1.1 Development Phase

During the development phase, given a list of directories (Source & Destinations), I wrote a Python script that:

- Copy and paste files from source to destination (Using wildcards for files)
- There are 4 different types of destination path that needs to be accounted for:
  1. Contains a subdirectory, does not have a Date folder (e.g., DstPath\ SubDir)
  2. Contains a subdirectory, has a Date folder (e.g., DstPath\Date\ SubDir)
  3. No subdirectory, does not have a Date folder (e.g., DstPath)
  4. No subdirectory, has a Date folder (e.g., DstPath\Date)

New files to be copy paste automatically can be added anytime into the rows of formatted excel sheet (figure 1), and the Python script will pick them up and run.

FileName	SrcPath	DstPath	DstSubDir	SubDirDate
filename1*.*	C:\Users\chiay\Desktop\Python\source1\	C:\Users\chiay\Desktop\Python\destination1\	destination1a	FALSE
filename2*.*	C:\Users\chiay\Desktop\Python\source2\	C:\Users\chiay\Desktop\Python\destination2\	destination1b	TRUE
filename3*.*	C:\Users\chiay\Desktop\Python\source2\	C:\Users\chiay\Desktop\Python\destination3\		FALSE
filename4*.*	C:\Users\chiay\Desktop\Python\source2\	C:\Users\chiay\Desktop\Python\destination4\		TRUE
…	…	…	…	…
…	…	…	…	…

Figure 1

2.1.1 User-testing Phase and Further Enhancements

The user-testing phase is a very important phase as it is the phase where most problems, or any inadequate areas will be identified and improved.

From the user testing we noticed that the Python script is significantly more stable and faster than the MS Access.

On top of that, we have also identified an area of improvement:

- As the script will be run in an interval of 15 minutes daily, we noticed that every time the script runs, the files which were previously already copied and pasted were once again being copy and paste, which changes the "Date modified" of the file (which is not what we want)

To solve this problem, I modified my script to check if the file already exists in the destination folder before copying over the file from the source path.

2.2 Automation using Qlikview

Background: Currently, several processes are being performed daily to produce the final Analysis report. These processes involve the filtering of data, massaging of data, and classifying the data in accordance with a set of classification criteria, before they are finally combined to form the final Analysis report. As these manual processes are disintegrated, the execution of task becomes tedious and time-consuming. Furthermore, as the processes are manual, the production of the final Analysis report is prone to human error.

My second task aims to:

- Merge the individual processes into a straight-through process in Qlikview
- To increase efficiency and accuracy by automating and converting existing manual processes to a more stable system
- To reduce end-user's manual process in performing related BAU task as much as possible (reduce human errors)
- As part of the automation, I will have to first migrate the existing MS Access Program to Qlikview and then design a dashboard for end-users

2.2.1 Development Phase

There are 3 parts to the development phase:

1. Migration of MS Access Program into Qlikview
2. Designing of Dashboard for BAU in Qlikview
3. Creating Macro for conversion and combining of files to produce files for final report

1. Migration of MS Access Program into Qlikview:

For the first part of the development phase, I was required to understand the logic behind the MS Access that is being run daily- this includes understanding over >40 SQL queries that were written in the program. My first goal in the migrating of MS Access program was to completely replicate the logic of MS Access program into Qlikview. After I have completely replicated the Access program into QV, my second goal was then to look at the flow of logic once again and from there, try to optimize the script of the QV app, by improving the code (remove redundant parts, integrating parts of the logic together to shorten code etc.) while achieving the same output. By doing so, I was able to shorten the loading time of the script significantly.

2. Designing of Dashboard for BAU in Qlikview

After migrating the MS Access program into Qlikview, using the output of the QV script, I have also designed a dashboard for data presentation which helps to reduce end-user's manual process. As much as I try to automate all the processes in Qlikview, there is still a short process which might still require manual work on some circumstances on some days. Hence, the Qlikview App that I have designed does not fully automate the processes, but rather it significantly reduces the end-user's manual work daily.

How the QV App helps end-user's:

1. QV automatically pick up files and run daily to produce csv files for final report
2. User goes to Qlikview dashboard and check for summary
3. If criteria are met in summary, no need for further manual work
4. If criteria not met, some manual work needs to be done, then steps 1-4 will be repeated until the criteria are met on the summary

Additional: Besides having the summary tab, there are also multiple tabs on the dashboard with the data presentation of the output of the QV program which allows end-users to do filtering of the data easily and observe the changes anytime should they need to.

E.g.	Process 1	Process 2	Process 3	Process 4
Previously (without QV):	Manual work →	Manual work →	Manual work →	Manual work
Currently (with QV):	Automated →	Automated →	Manual work (yes/no depending on summary output) →	Automated

3. Creating Macro for conversion and combining of files

Finally, because the file output by Qlikview can only be of cvs/txt file, but the files need to be in xls format for the generation of the final report, I have thus written an excel Macro which does the conversion and combining of files for the generation of the final report.

Some challenges that I have encountered and overcome with Automation using Qlikview assignment:

- Scripting and dashboard-designing in Qlikview were relatively new to me, so it took me a while to get a hang of it
- Understanding the individual queries (from MS Access) went relatively smoothly but however, putting >40 queries together to understand the overall logic of the MS Access (for optimization) Program was challenging for me → Organizing and summarizing the all queries in an excel sheet helps me a lot in understanding the logic as a whole

2.2.2 User-testing Phase and Further Enhancements

The QV App is still in the process of user-testing, any further enhancements/improvements will only be done after the App has been run and tested by users.

2.3 Enhancing Daily Broadcast Email

Recall: As documented in my interim report, I have written 3 Macros for automation- daily BC Source files status Email and 2 Daily QV status Email Macros, which helps to monitor critical Apps and files for BAU.

Though the 3 macros work fine initially but upon further testing, we realized that they tend to be unstable especially when they are other software/macros running in the background of the pc. Hence, I was tasked to convert the Macros into Python script, to make the daily automation become more stable and reliable as compared to using Macros.

My third task aims to:

- Improve the overall aesthetics of the email body
- Convert the 2 Macros logic from VBA into Python, to make daily broadcast more stable and robust

2.3.1 Development Phase

Having written the 3 Macros in VBA, I already have a clear understanding of the logic behind the Email Broadcast Macros. Hence, converting the Macros to Python script went quite smoothly for me. In my opinion, coding in Python was relatively cleaner and easier because:

Firstly, Python has many useful libraries that makes the manipulation of data easier (E.g., Pandas library in Python allow me to manipulate data easily with data frames)

Secondly, Python can read excel data without having to open workbooks, which cannot be done using VBA. This makes testing and implementation a lot faster and easier.

During the development phase, I have also added a few enhancements to the automation:

1. If any file/App is unrefreshed, check if that day is a Public Holiday for the country, if it is, state that under remarks in the email before broadcasting
2. Make the automation more user-friendly. e.g., if any additional file/App/Drive is to be checked, users can just input it in the excel files for users anytime and the script will pick them up without having them to edit the script.

2.3.2 User-testing Phase and Further Enhancements

The Python scripts are still in the process of user-testing, any further enhancements/improvements will only be done after the they have been run and tested.

The figures below show the old (macro) VS new (Python script) format for the automated email broadcast.

Daily Qlikview status 9AM email broadcast (old (macro) vs new (python)):

Old:

Good morning teams,

Please find the Qlikview Status for today.

Application	Status	Last Update Time	Remarks
Critical App 1	☺	25/05/2021	
Critical App 2	☺	25/05/2021	
Critical App 3	☺	25/05/2021	
Critical App 4	😐	24/04/2021	• will be refreshing at 10AM
Critical App 5	☺	25/05/2021	
Critical App 6	☺	25/05/2021	
Critical App 7	☺	25/05/2021	
...			
K Drive	😐	25/05/2021	• low on space

Best regards,

New:

Good morning teams,

Please find the Qlikview status for today below.

- Refreshed
- Pending
- Failed

Status	SLA	Application	Last Update Time	Remarks
●	09:00:00	Critical App 1	11 Aug 2021, 09:06	Refer to ** note below
●	09:00:00	Critical App 2	11 Aug 2021, 09:06	
●	09:00:00	Critical App 3	11 Aug 2021, 09:06	
●	09:00:00	Critical App 4	11 Aug 2021, 09:06	
●	09:00:00	Critical App 5	11 Aug 2021, 09:06	
●	09:00:00	...	11 Aug 2021, 09:06	
●	09:00:00	...	11 Aug 2021, 09:06	
●	09:00:00	...	11 Aug 2021, 09:06	
●	09:00:00	...	11 Aug 2021, 09:06	
●	09:00:00	...	11 Aug 2021, 09:06	
●	09:00:00	...	11 Aug 2021, 09:06	
●	10:00:00	Critical App	In progress	

** All Controllers to take note:

•
re
•
ir

Non-QV status:

Usage	Free	Shared Folder	Date Checked	Remarks
48.1 %	247.2 GB		11 Aug 2021	low on space for housekeeping
48.1 %	247.2 GB		11 Aug 2021	low on space for housekeeping

Daily Qlikview status 11AM/2PM email broadcast (old (macro) vs new (python)):

Old:

Good morning teams,

The table below shows the App(s) that are yet to be updated

Application	Status	Last Update Time
Critical App 4	😑	24/04/2021

Best regards,

New:

Good afternoon teams,

The table below shows the App(s) that are yet to be updated.

	🟡 -Pending	
Application	Last Update Time	Status
Critical App	2021-08-10 09:06:00	🟡

Best regards,

BC Source file status email broadcast (old (macro) vs new (python)):

Old:

Good afternoon teams,

Please find the BC Source File Status as of today 9am.
All controllers to check File Last Update and report if any issue (BS/IT):

Report Date	25/05/2021			
Entity	**Type**	**Name**	**Last Updated**	**Remarks**
ALL	QV	App1	25/05/2021	refreshed
ALL	QV	App2	24/04/2021	refreshed
SG	QV	App3	25/05/2021	refreshed
SG	OR	Txt	24/05/2021	controllers to note
SG	OR	Txt	25/05/2021	refreshed
SG	OR	Txt		controllers to note
TH	FR	qvd	25/05/2021	refreshed
TH	FR	qvd	25/05/2021	refreshed
...	...	...	...	...

Best regards,

New:

Good afternoon teams,

Please find the BC Source file status as of 2pm.
All Controllers to check file last update and report if there are any issues (BS/IT):

- Refreshed
- Controllers to note

Report Date	10/08/2021				
Entity	**Type**	**Name**	**Last Updated**	**Status**	**Remarks**
SG	...	popcorn.TXT	No file	●	10/08/2021 is SG holiday
HK	...	cookie.TXT	No file	●	
MY	...	poppycorn.txt	No file	●	
NYK	...	monkey.txt	No file	●	
ALL	...	lala.txt	11/8/2021 9:06	●	
ALL	...	cookie haha2.qvw	11/8/2021 9:06	●	

Best regards,

3. Things that I learnt/ Achievements

- I have gained technical skills like VBA, Qlikview, SQL, Python MS Access, Excel, and also soft skills like time management and communication skills
- I have also improved my understanding on data massaging and data presentation
- The user-testing phase is a very important phase as it is when unprecedented scenarios are being encountered and it is through this phase whereby problems are identified and fixed, before it is put to actual use
- It is important that the code offers flexibility for people to easily customize to use in the future for their own needs
- It is a good practice to look for ways to improve your code, things like looking for ways to improve your coding style can make your code run faster, which will greatly increase efficiency and productivity