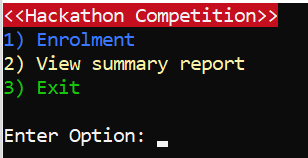
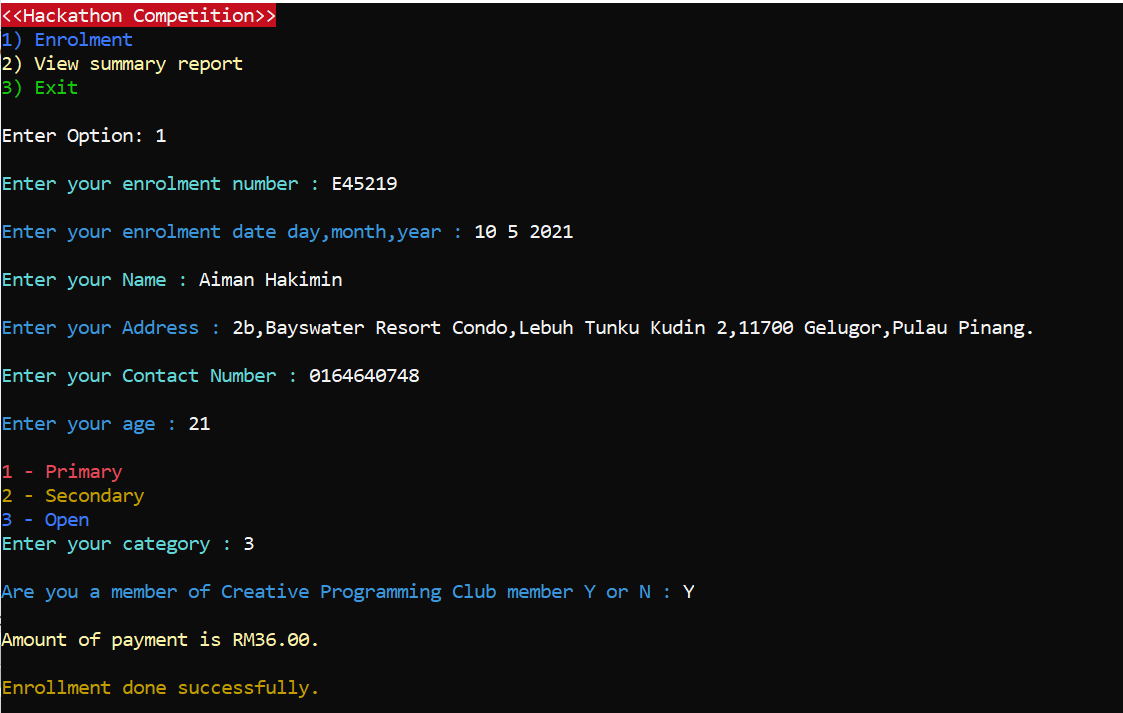
Report

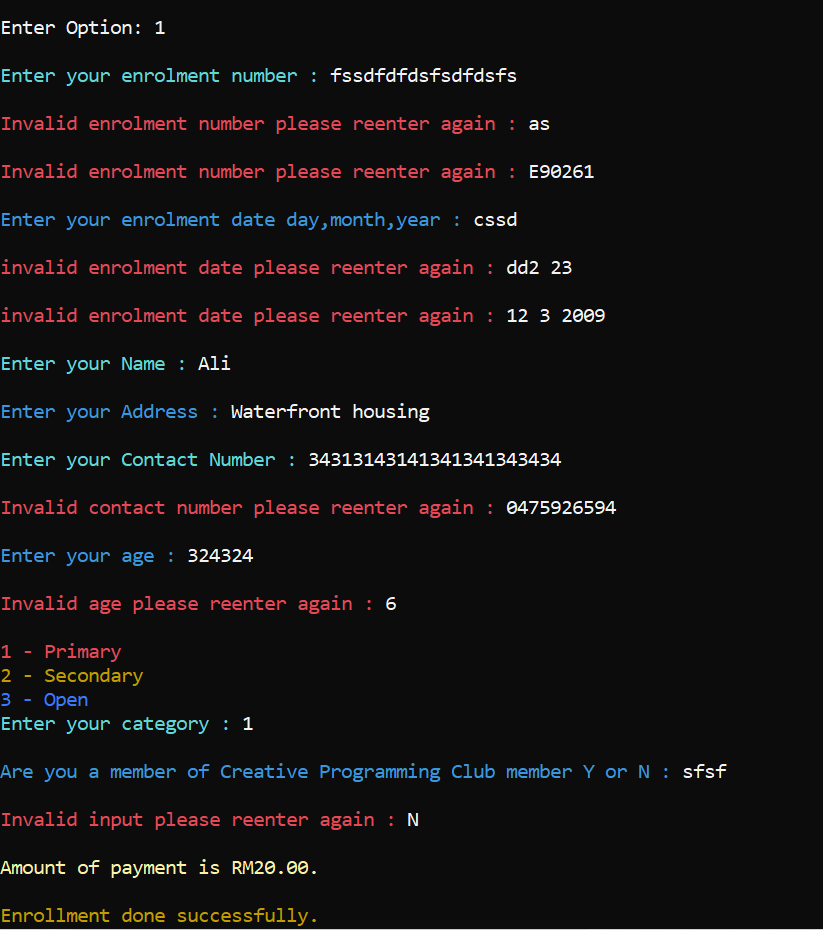
Screenshots:

1.Main Menu

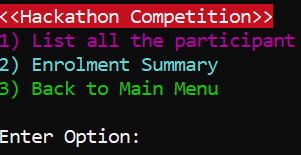


* 1. Enrolment

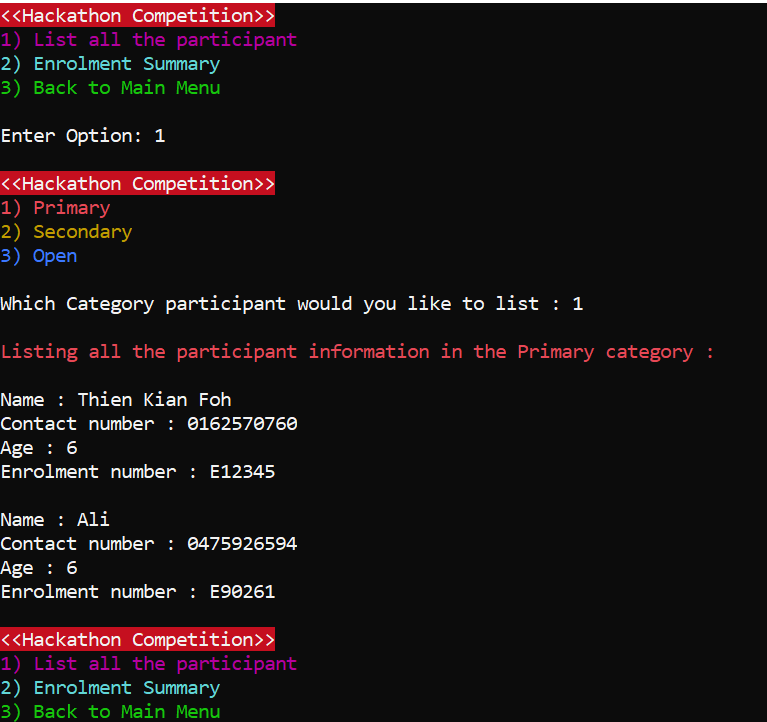


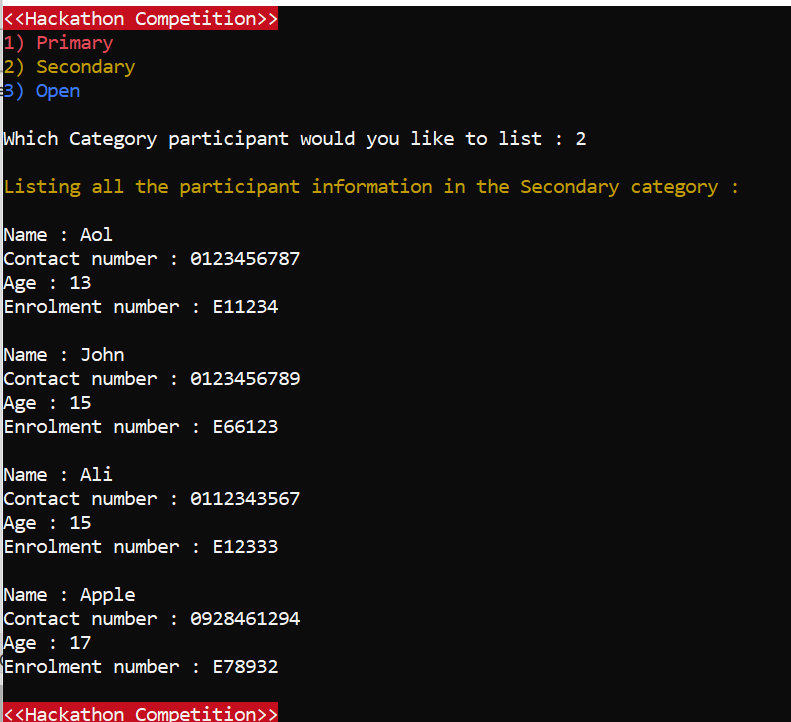


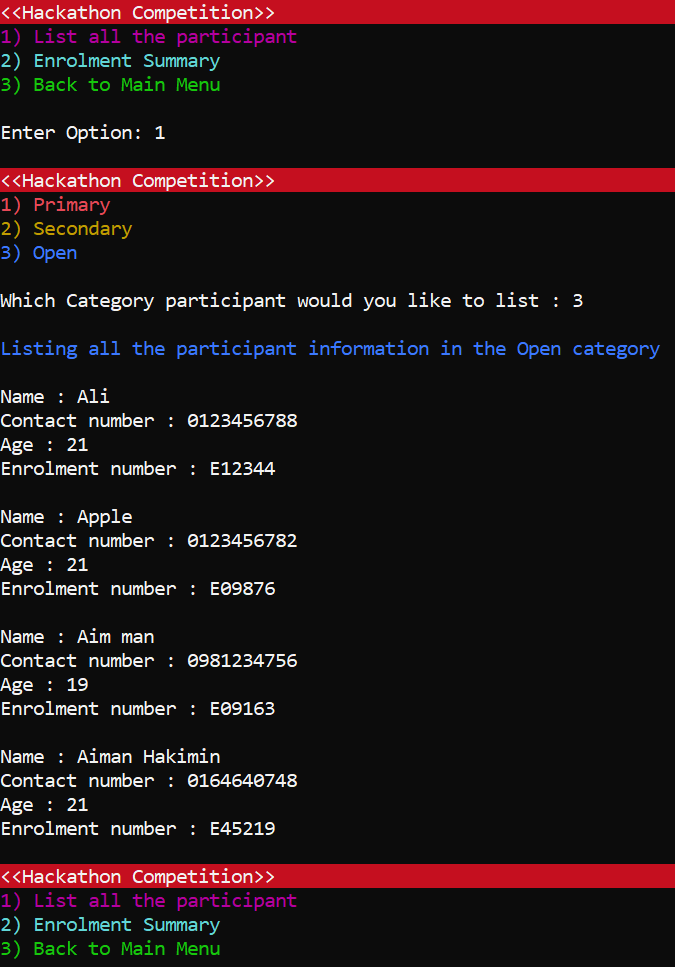
2. Summary Enrolment Menu

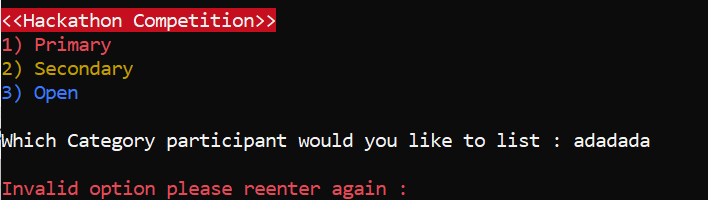


2.1 List all the participant





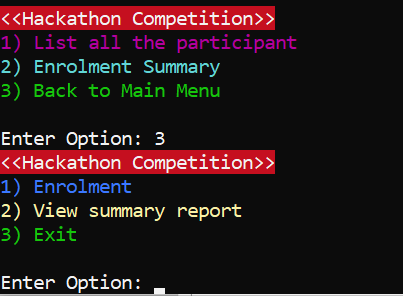


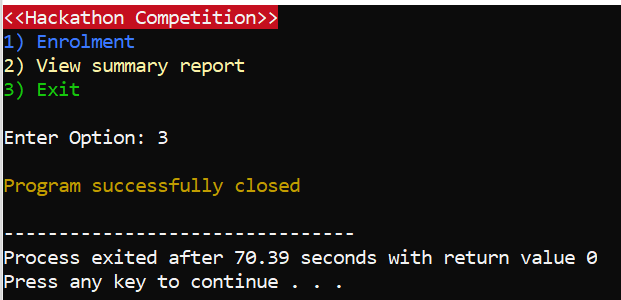


2.2 Enrolment Summary



3. Exit





Weakness:

* If you enter into enrolment, it must be fully completed
* Enrolment time is not included into the program
* It’s a terminal application, not a GUI (Graphical User Interface) application
* There is no help option, and requires intuitively knowing the options
* Menu will not be refreshed after entering wrong input

Strengths:

even though we are very confident to say that our program also has a lot of strengths even though it might not be perfect.

Here are some of the example to demonstrates the strengths of our program:

**Validation**

Our program is able to validate most of the data that is entered by the user as shown above with screenshots. when an input failure occur our program is able handle it by clearing the input buffer and the input failure to prevent infinite loop in our program. In the main menu the option selection our program can validate the data entered by the user if it’s an incorrect data our program will keep asking the user to input until a correct data is enter same goes for other option selection. As for the enrolment number our program can validate and make sure the enrolment number start with E and following by 5 digits. Additionally, every enrolment number can only be use once if a same enrolment number that had been previously used is entered, our program will ask the user to enter again. The enrolment date must start with day, month and year, our program can even validate and making sure that the month and day entered by the user must be within range for example January can only consist of 31 days and February 28 days unless it is a leap year then the maximum number of days in February will become 29 days. Our program also can make sure the contact number user entered must be integer and must be ten digits. Overall we believe that one of the greatest strength in our program is, it can validate and make sure the data from user is correct.

**Data remain after closing the program**

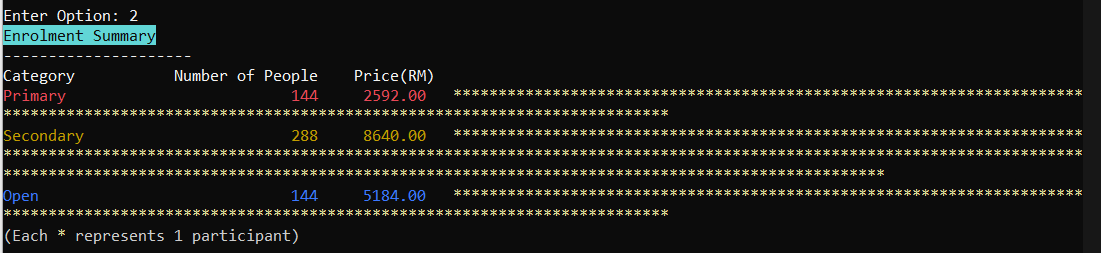
Most of the program we had done before is unable to save data into secondary memory meaning once the program is close the program will be reset. But our program allow user input data and validate to make sure it’s correct then store it into text file. By doing this even though our program is close the next time the user run the program again they can still access the data they had previously entered. This is because of our program actually save all the data from the user into text file and then when the user need to access it again our program can just read thought the content in the text file and display it on the terminal.

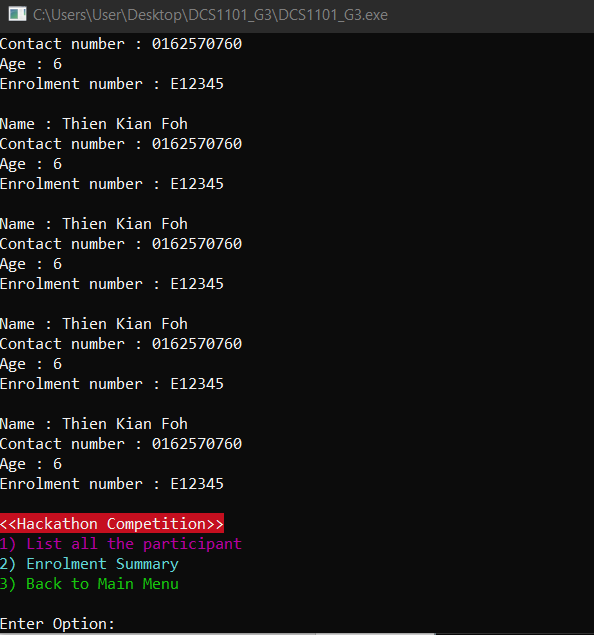
**Allow User to do enrolment repeatedly**

Our program allow user to do the enrolment repeatedly after they finish an enrolment the program will bring the user back to the main menu allowing user to do the enrolment again and again there is no limit to how many enrolment the user can make. After finish doing the enrolment user can decide to view the summary report or exit the program.

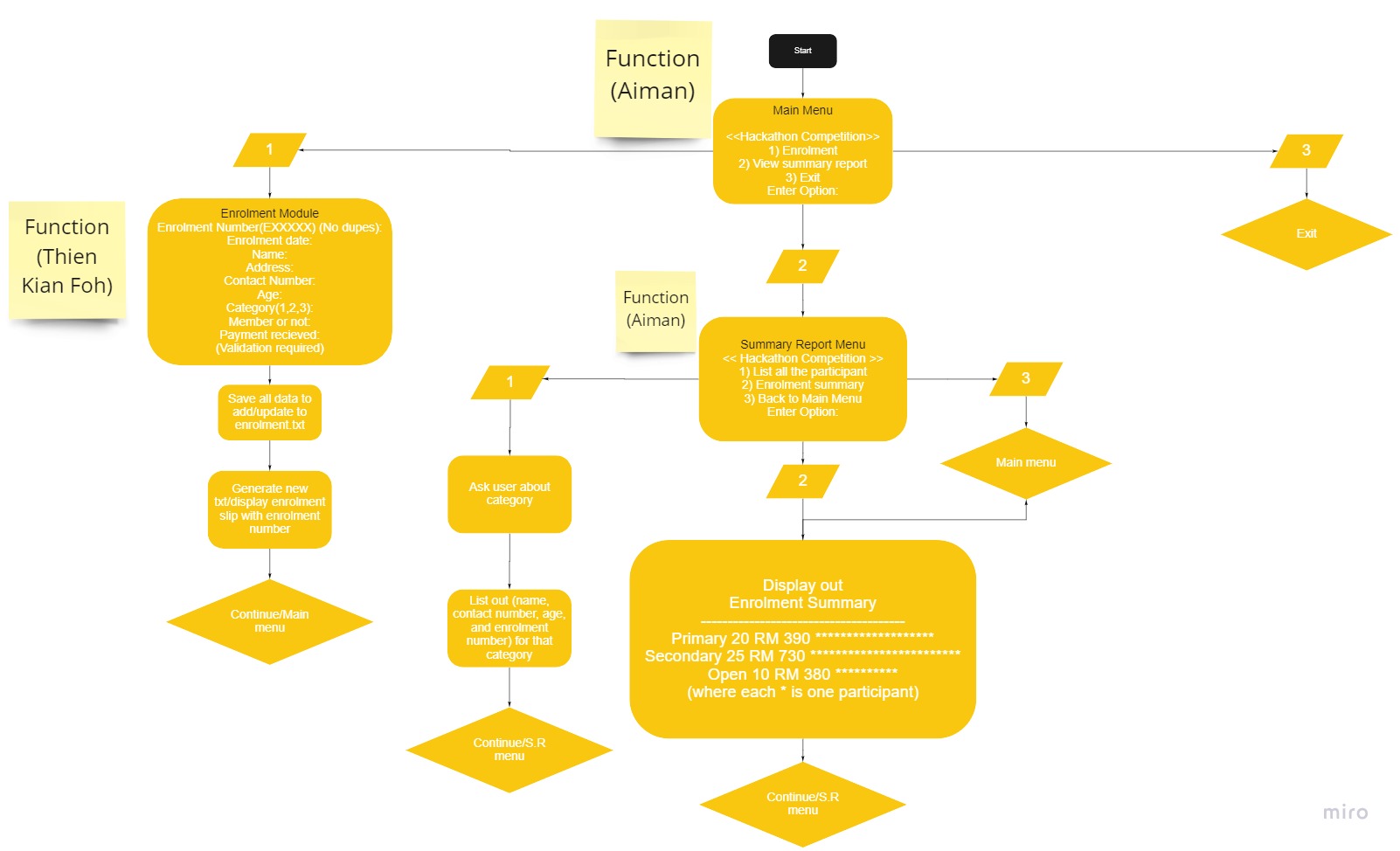
**Able to Handle Large Amount of data**

Shortly after we completed our project. We do some stress test to see can our program handle a large amount of data with more than five hundred enrolment information in the text file our program is still able to list out all the participant information in the View summary report without lagging or crash. This shows that our program is optimized and ready to handle sizeable amount of data without having any trouble. We believe this a very impressive strength of our program to be able to handle more than five hundred enrolment information.





Work Breakdown:

**This is a picture of how we break down our work and assign the tasks to both of us.**

We split the project into two main part the first part is the enrolment module and the second part is view summary report module. The enrolment is assign to Thien and the view summary report is assign to Aiman. While we are doing our task we also split our task into many part by using function so that we can focus on one thing at a time. Throughout this project we communicate with each other using discord and whatsapp to keep everyone on the same page so that we won’t get confuse while working on the project, and when we face any problem we’ll try to find a solution together to solve the problem. We also use github to create a repository and work on the project together. Github helps us a lot when one of us change the code we are able to know which line is changed. By using github we are able to work on the project efficiently.

Reflection & Evaluation

*Aiman Hakimin Bin Ahmad Khairi*

This course has made me work for long hours into the night, mostly I end up sleeping at 2AM in the morning. However, despite my lack of sleep, I’ve managed to defeat every hurdle with great success. It’s amazing how far I’ve come from not being confident with my skills to being proficient.

There were many challenges I faced but the 2 prominent ones were file handling as well as navigating a project file of this size.

I will start with the latter, this is project is massive close to 700 lines of code, it felt pretty daunting staring at a screen trying to find the right code. I solved this by separating the two halves of the code: enrolment and summary. So, we can work on each other’s part separately.

Next, the file handling was a pain in the beginning. I had not learnt the full concepts of it and had to rely on Stack Overflow as well as YouTube. However, these did not explain more advanced topics as the goal in my mind was to search through a text file and selecting specific words to add to a count. My teammate, on the other hand, was doing well with file handling so I read his code and found a way to make my side work as well. The lecturer also gave me some tips to make my code more efficient to prevent it from reading constantly, it reads it once and updates the values as the user inputs new data.

To end of my report, I would like give a shoutout to my teammate, Thien Kian Foh for being an awesome teammate. I would also shoutout to my lecturer, who answered our questions along the way.

**Thien Kian Foh**

**Reflection**

At the beginning of the project I felt like there is a lot to take in about the project we need to do enrolment and need to make sure the program can generate a text file and able to read from the text file. But after a few days when I had a meeting with Aiman using discord we are able to break down the work into many part and that makes me felt a bit more relief. Because I am able to know what I should do in this project. Right after the meeting I keep working on project because i know that the module Aiman need to work on is depend on the module I am working on. I did not want to slow him down so I keep working on the enrolment module. After three to four days, I am able to finish my part so that Aiman can continue working on the view summary report module. In just around 2 days Aiman finished the view summary report module I am very surprised the fact that he is able to finish the module in just two days. In less than a week we are able to finish the project. Overall i felt very grateful to be in a group with Aiman, and I believe that I might not be able to finish the project without him.

**Evaluation**

Throughout the project I face a lot of problem. But I am able to learn about a lot things such as new library and also I am able to increase my critical thinking skill by finding solution to solve a problem. I had encountered a lot of problem in the duration of working on the project. one of the problem I had encountered is how do we avoid input failure I did not know about input failure until I entered an alphabet in the main menu, input failure will cause infinite loop. So I had to find a solution to solve this problem. I google search the solution for this problem and try to understand how it work. After some research finally I were able to solved this problem by implementing both cin.clear and cin.ignore functions. After we finished the project I had a sense of accomplishment because of how much work we put into to make the program work very well. All in all I am glad we are able to finish this project in time, it was a pleasure to work together with Aiman.