

GROUP ASSIGNMENT

CT038-3-2-OODJ

OBJECT ORIENTED DEVELOPMENT JAVA

HAND OUT DATE: 5th April 2022

HAND IN DATE: 31th May 2022

GROUP MEMBER:

|  |  |
| --- | --- |
| Wong Jun Yuan | TP067666 |
| Lee Woon Xun | TP067738 |

Table of Contents

[1.0 Introduction 1](#_Toc104499068)

[2.0 Diagram 2](#_Toc104499069)

[2.1 Class Diagram 2](#_Toc104499070)

[2.2 Use Case Diagram 3](#_Toc104499071)

[2.3 Use Case Specifications 4](#_Toc104499072)

[3.0 Source Code 21](#_Toc104499073)

[3.1 Header File 21](#_Toc104499074)

[3.2 Variable 21](#_Toc104499075)

[3.3 Control Structure 22](#_Toc104499076)

[3.3.1 If 22](#_Toc104499077)

[3.3.2 If-else 23](#_Toc104499078)

[3.3.3 Nested If-else 24](#_Toc104499079)

[4.0 Looping Structure 24](#_Toc104499080)

[4.1 While Loop 24](#_Toc104499081)

[4.2 For Loop 25](#_Toc104499082)

[5.0 Array 25](#_Toc104499083)

[6.0 ArrayList 26](#_Toc104499084)

[7.0 OOPs 26](#_Toc104499085)

[7.1 Class 26](#_Toc104499086)

[7.2 Object 27](#_Toc104499087)

[7.3 Encapsulation 27](#_Toc104499088)

[7.4 Inheritance 28](#_Toc104499089)

[7.5 Constructor 28](#_Toc104499090)

[7.6 Get-Set Method 29](#_Toc104499091)

[7.7 Normal Method 30](#_Toc104499092)

[7.8 Exceptional handling 30](#_Toc104499093)

[8.0 File 31](#_Toc104499094)

[8.1 Write 31](#_Toc104499095)

[8.2 Read 32](#_Toc104499096)

[8.3 Search 33](#_Toc104499097)

[8.4 Generate Report 34](#_Toc104499098)

[4.0 Output Screen Shot 36](#_Toc104499099)

[5.0 Conclusion 56](#_Toc104499100)

[6.0 References 57](#_Toc104499101)

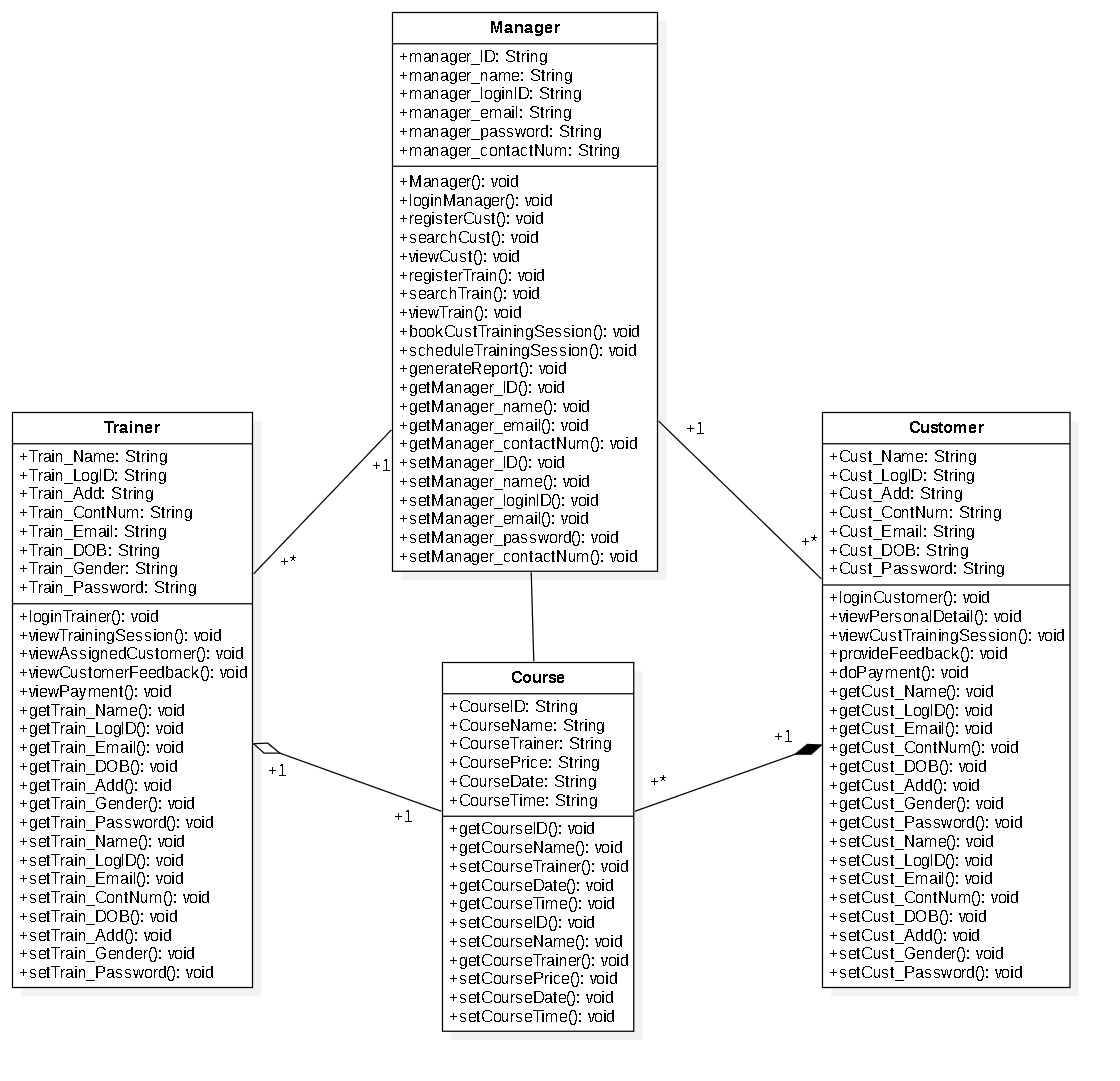
# **1.0 Introduction**

The concept of analysing object-oriented programming is used in this project to design and develop an APU Gym Centre Management System. Moreover, this system is required to design for the manager to complete the registration of the user and support customers to register their account and book their training session as well. In addition, trainer of the centre can check the individual training session through this system and also collect the payment that made by customer. They can also receive the feedback from the customer about the training session.

Therefore, this report will provide two complete Use Case and Class Diagram design diagrams and will explain object-oriented programming implementation all features on this project. The system is not only built on basic language concepts but also includes File storage capabilities. The project users include Manager, Trainer and Customer. Most of these functions rely on the Manager to operate such as scheduling new courses, creating Customer and Trainer, helping customers book courses, and generating reports.

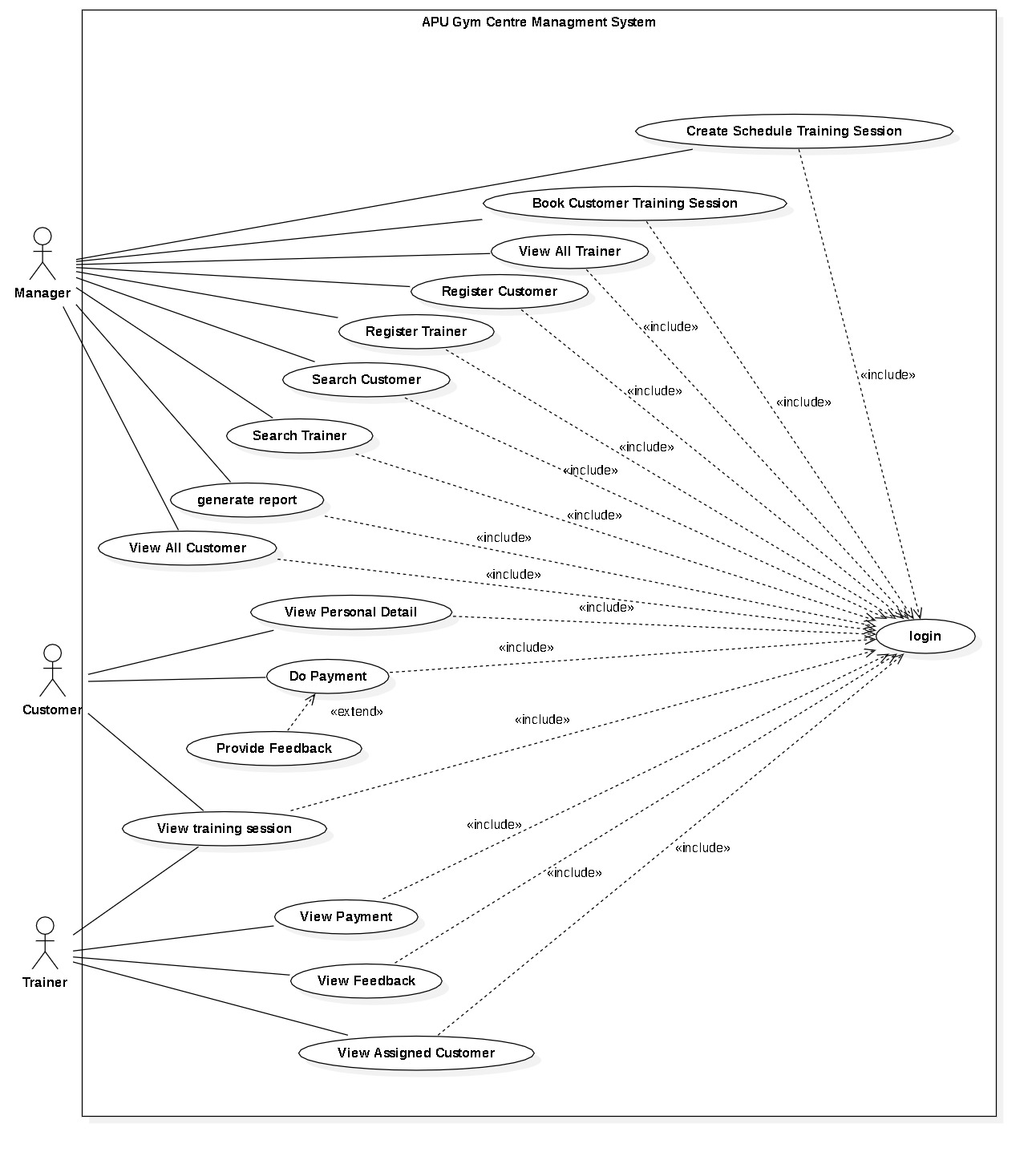
# **2.0 Diagram**

## **2.1 Class Diagram**



*Figure 1: Class Diagram of APU Gym Management System.*

## **2.2 Use Case Diagram**



*Figure 2: Use Case of APU Gym Management System.*

## **2.3 Use Case Specifications**

|  |  |
| --- | --- |
| **Use Case** | Login |
| **Brief Description** | This use case allows a user to login to the system. |
| **Actors** | Manager, Trainer, and Customer |
| **Preconditions** | User must have account and password before login the system. |
| **Main Flow** | 1. When the user needs to select the login port of the role, such as Manager, Trainer or Customer. 2. The system requires the user to enter a valid account ID and password and click the LOGIN button. 3. After the user enters the correct account ID and password, the system will run the user and enter the next step. |
| **Alternative Flows** | 1. Manager does not remember its own account ID and password; it needs to be obtained through background data. If the Customer and the Trainer do not remember, they need to find the Manager for help. 2. System login fails in the following cases  * The account ID does not exist * The account ID or password is entered incorrectly |

|  |  |
| --- | --- |
| **Use Case** | Register Customer |
| **Brief Description** | This use case allows a manager help customer to register new account into the system. |
| **Actors** | Manager |
| **Preconditions** | 1. Managers need to login to system first. 2. Customer needs provide manager him personal information included full name, address, contact number, email, date of birth and password. |
| **Main Flow** | 1. In the Manager Dashboard, the user needs to select the “Register Customer” button. 2. In the form of Register Customer, Manager can fill in the personal information provided by the customer to create an account for the customer. 3. After ensuring that the customer information is correct, click the Submit button to complete the register step. |
| **Alternative Flows** | 1. The user needs to make sure to fill in all the information, otherwise the creation will be invalid. 2. The Manager needs to pay attention to the account ID format CUST{Login ID}-{Name} automatically generated by the system after the creation is successful. For example, the Customer name is Lee, and the manager sets the Login ID to 001. Then the customer's account ID will be generated as CUST001-Lee. 3. Manager can click “Exit” button to back to dashboard. |

|  |  |
| --- | --- |
| **Use Case** | View All Customer |
| **Brief Description** | This use case allows a manager to view all customer and their information into the system. |
| **Actors** | Manager |
| **Preconditions** | Managers need to login to system first. |
| **Main Flow** | 1. In the Manager Dashboard, the user needs to select the “View Customer” button. 2. In the table will show all available customer details on this page. |
| **Alternative Flows** | 1. If the system does not have a customer account available, nothing will be displayed. 2. Managers can click the “Exit” button to return to the Dashboard. |

|  |  |
| --- | --- |
| **Use Case** | Search Customer |
| **Brief Description** | This use case allows a manager to search customer into the system. |
| **Actors** | Manager |
| **Preconditions** | 1. Managers need to login to system first. 2. Manager must be known which one customer’s account ID before search. 3. Customer’s account must be available in the system database. |
| **Main Flow** | 1. In the Manager Dashboard, the user needs to select the “Search Customer” button. 2. The manager fills in the account ID of the client who wants to search. 3. Click the “Submit” button to search after completing the form. 4. Customer details will show on the table. |
| **Alternative Flows** | 1. If show error message means cannot find the customer’s account ID, manager can register new customer. 2. Manager can click “Exit” button to back to dashboard. |

|  |  |
| --- | --- |
| **Use Case** | Register Trainer |
| **Brief Description** | This use case allows a manager to register new account for trainer into the system. |
| **Actors** | Manager |
| **Preconditions** | 1. Managers need to login to system first. 2. Trainer needs provide manager him personal information included full name, address, contact number, email, date of birth and password. |
| **Main Flow** | 1. In the Manager Dashboard, the user needs to select the “Register Trainer” button. 2. In the form of Register Trainer, Manager can fill in the personal information provided by the trainer to create an account for the trainer. 3. After ensuring that the trainer information is correct, click the Submit button to complete the register step. |
| **Alternative Flows** | 1. The user needs to make sure to fill in all the information, otherwise the creation will be invalid. 2. The Manager needs to pay attention to the account ID format TRAIN{Login ID}-{Name} automatically generated by the system after the creation is successful. For example, the trainer’s name is John, and the manager sets the Login ID to 001. Then the trainer’s account ID will be generated as TRAIN001-John. 3. Manager can click “Exit” button to back to dashboard. |

|  |  |
| --- | --- |
| **Use Case** | View All Trainer |
| **Brief Description** | This use case allows a manager to view all trainer and their information into the system. |
| **Actors** | Manager |
| **Preconditions** | Managers need to login to system first. |
| **Main Flow** | 1. In the Manager Dashboard, the user needs to select the 'View Trainer' button. 2. In the table will show all available customer details on this page. |
| **Alternative Flows** | 1. If the system does not have a trainer account available, nothing will be displayed. 2. Managers can click the “Exit” button to return to the Dashboard. |

|  |  |
| --- | --- |
| **Use Case** | Search Trainer |
| **Brief Description** | This use case allows a manager to search trainer into the system. |
| **Actors** | Manager |
| **Preconditions** | 1. Managers need to login to system first. 2. Manager must be known which one trainer’s account ID before search. 3. Trainer’s account must be available in the system database. |
| **Main Flow** | 1. In the Manager Dashboard, the user needs to select the “Search Trainer” button. 2. The manager fills in the account ID of the client who wants to search. 3. Click the “Submit” button to search after completing the form. 4. Trainer details will show on the table. |
| **Alternative Flows** | 1. If show error message means cannot find the trainer’s account ID, manager can register new trainer. 2. Manager can click “Exit” button to back to dashboard. |

|  |  |
| --- | --- |
| **Use Case** | Schedule Training Session |
| **Brief Description** | This use case allows a manager to create new training session or course to each trainer. |
| **Actors** | Manager |
| **Preconditions** | 1. Managers need to login to system first. 2. Managers need to register new trainer, or available trainer in the system. 3. Managers must list out for which trainer, course name, price, date, and duration time of the course. |
| **Main Flow** | 1. In the Manager Dashboard, the user needs to select the Schedule Training Session' button. 2. The manager must select trainer, and filled up all detail about the course, included Course name, price, date, and duration time of the course. 3. Click the “Submit” button to create the new training session. |
| **Alternative Flows** | 1. If show empty on “Trainer” drop menu, manager need create new trainer account first. 2. If show error “Please enter a numeric value” means price of the course need be number. 3. If show error “Please fill in the course and price field” means some field incomplete, manager need fill in all data field. 4. Manager can click “Exit” button to back to dashboard. |

|  |  |
| --- | --- |
| **Use Case** | Booking Customer Training Session |
| **Brief**  **Description** | This use case allows a manager to help customer book the training session. |
| **Actors** | Manager |
| **Preconditions** | 1. Managers need to login to system first. 2. The customer account and course need be available into system database. |
| **Main Flow** | 1. In the Manager Dashboard, the user needs to select the “Book Customer Training Session” button. 2. Manager selects Customer Account ID and selects Course. 3. All course details will be displayed below. 4. Click the “Submit” button to complete the book training course steps. |
| **Alternative Flows** | 1. If show empty on Customer ID or Course, manager need to create customer account or create course first. 2. Manager can click “Exit” button to back to dashboard. |

|  |  |
| --- | --- |
| **Use Case** | Generate Report |
| **Brief Description** | This use case allows a manager to easy automatic generate the pdf file report. |
| **Actors** | Manager |
| **Preconditions** | 1. Managers need to login to system first. 2. The system database needs available customer or trainer account. |
| **Main Flow** | In the Manager Dashboard, the user needs to select the “Generate Report” button system will auto generate report to the document place. |
| **Alternative Flows** | If did not show nothing means manager need to create customer account or create trainer first. |

|  |  |
| --- | --- |
| **Use Case** | View Training Session |
| **Brief Description** | This use case allows customer and trainers to check the next available training sessions. |
| **Actors** | Customer and Trainer |
| **Preconditions** | 1. Users need to login to system first. 2. Manager must have help customer book training session before. |
| **Main Flow** | 1. In the Dashboard, the user needs to select the “View Own Training Session” or “View Training Session” button. 2. All training session will show on the table. |
| **Alternative Flows** | 1. If did not show nothing means no available training session. 2. Users can click “Exit” button to back to dashboard. |

|  |  |
| --- | --- |
| **Use Case** | Do Payment |
| **Brief Description** | This use case allows customer to pay the course. |
| **Actors** | Customer |
| **Preconditions** | 1. Customer need to login to system first. 2. Manager must have help customer book training session before. |
| **Main Flow** | 1. In the Dashboard, the user needs to select the “Do Payment” button. 2. Customer needs select which one course to pay. 3. After entering the amount and click the “Pay” button to complete payment action. |
| **Alternative Flows** | 1. If did not course means customer need find manager book the training session first. 2. If show error means, customer need fill numeric value in “Enter Amount” field. 3. Customer can click “Exit” button to back to dashboard. |

|  |  |
| --- | --- |
| **Use Case** | Provide Feedback |
| **Brief Description** | This use case allows customer to write the feedback after payment and complete the course. |
| **Actors** | Customer |
| **Preconditions** | 1. Customer needs to login to system first. 2. Customer needs pay the course before provide feedback. |
| **Main Flow** | 1. In the Dashboard, the user needs to select the “Provide Feedback” button. 2. Customer needs select which one course to write feedback. 3. After write the feedback on the text area and click the “Submit” button to complete provide feedback action. |
| **Alternative Flows** | 1. If did not show course means customer need pay the course first, or customer still unbook the training session. 2. Customer can click “Exit” button to back to dashboard. |

|  |  |
| --- | --- |
| **Use Case** | View Personal Detail |
| **Brief Description** | This use case allows customer to see they personal information. |
| **Actors** | Customer |
| **Preconditions** | 1. Customer available account in the system. 2. Customer need to login to system first. |
| **Main Flow** | In the Dashboard, the user needs to select the “View Personal Details” button and all the details will be displayed below. |
| **Alternative Flows** | Customer can click “Exit” button to back to dashboard. |

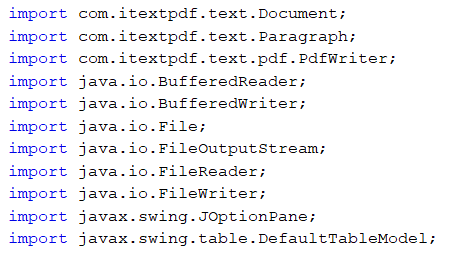
|  |  |
| --- | --- |
| **Use Case** | View Payment |
| **Brief Description** | This use case allows the trainer to see which customers have completed their payments. |
| **Actors** | Trainer |
| **Preconditions** | 1. Trainers need to login to system first. 2. Trainers must have training session or course before view payment. |
| **Main Flow** | In the dashboard, the user needs to select the “View Payment” button and all customers who have completed payments will be displayed below. |
| **Alternative Flows** | 1. If the table is empty, it means that no customer has paid yet. 2. The trainer can click the “Exit” button to return to the Dashboard. |

|  |  |
| --- | --- |
| **Use Case** | View Feedback |
| **Brief Description** | This use case allows the trainer to see which customers have submit their feedback. |
| **Actors** | Trainer |
| **Preconditions** | 1. Trainers need to login to system first. 2. Trainers must have training session or course before view feedback. |
| **Main Flow** | In the dashboard, the user needs to select the “View Feedback” button and all customers who have provide feedback will be displayed below. |
| **Alternative Flows** | 1. If the table is empty, it means that no customer submits the feedback. 2. The trainer can click the “Exit” button to return to the Dashboard. |

|  |  |
| --- | --- |
| **Use Case** | View Assigned Customers |
| **Brief Description** | This use case allows the trainer to see which customer have booked which of his courses. |
| **Actors** | Trainer |
| **Preconditions** | 1. Trainers need to login to system first. 2. Trainers must have training session or course. |
| **Main Flow** | In the dashboard, the user needs to select the “View Assigned Customer” button and all customers who have book his course will be displayed below. |
| **Alternative Flows** | 1. If the table is empty, it means that no customer book his course. 2. The trainer can click the “Exit” button to return to the Dashboard. |

# **3.0 Source Code**

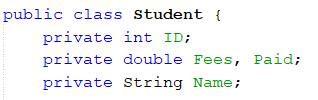
## **3.1 Header File**



*Figure 3: Header File in Manager Class*

Based on the Figure, these are the library that we had import to the java including File, BufferedReader, itextpdf and so on. The first three library that has been imported is for generating the PDF file. For this library, we have been downloading a jar for this itextpdf library. Next is File library, we need to import this library since we used text file to store our information and documentation. Furthermore, BufferedReader and FileReader has been used to read the information in the text file.

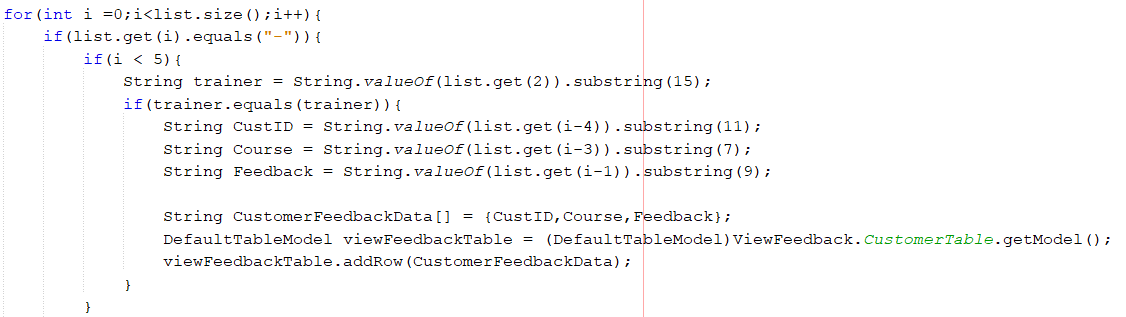
## **3.2 Variable**

  
*Figure 4: Variable in Student Class*

There are several variables that we have been used and executed in our java program. For example, the data type of java has also been used in the program such as String, Boolean, int, double, and so on. In the java program, most of the variable will use String data type since it can assign numeric or alphabet to the variable.

## **3.3 Control Structure**

### **3.3.1 If**



*Figure 5: If Control Structure in Trainer Class*

If of the control structure is very important to verify what user’s activity and take the user’s activity to the specific code and run it for fulfilling the user requirement. As a result, in Figure, it showed that if the i lower than 5, it will enter the If statement. After that, it gets the value from the ArrayList and compare the value. If the value is the same value, it will execute the If statement.

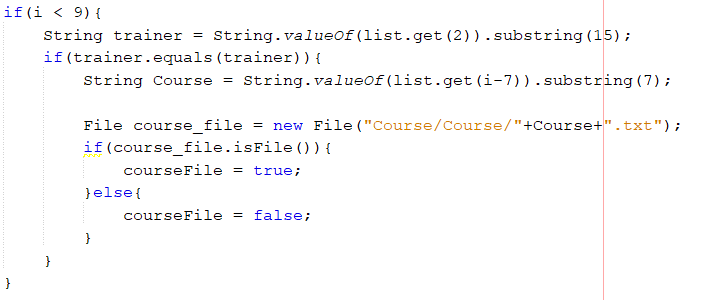
### **3.3.2 If-else**



*Figure 6: If-else in Manager Class*

If-else control structure can be used for decision making of the user’s input. For example, every input from user will be different and it will also show different output for the user. Based on this Figure, the If-else statement has been executed for checking the file in the folder. When the specific file is available in folder, it will enter the If statement, otherwise it will join the Else statement.

### **3.3.3 Nested If-else**

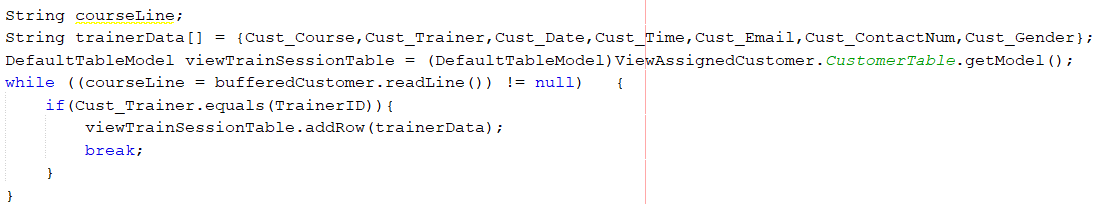


*Figure 7: Nested If-else in Customer Dashboard page*

Control structure of Nested If-else allow to have some If-else or If statement within an If statement. It helps the program to have more accurate in executing the code. In the Figure, we have been used Nested If-else in the trainer class to make a double verification within the For Loop in the program to meet the requirement.

## **4.0 Looping Structure**

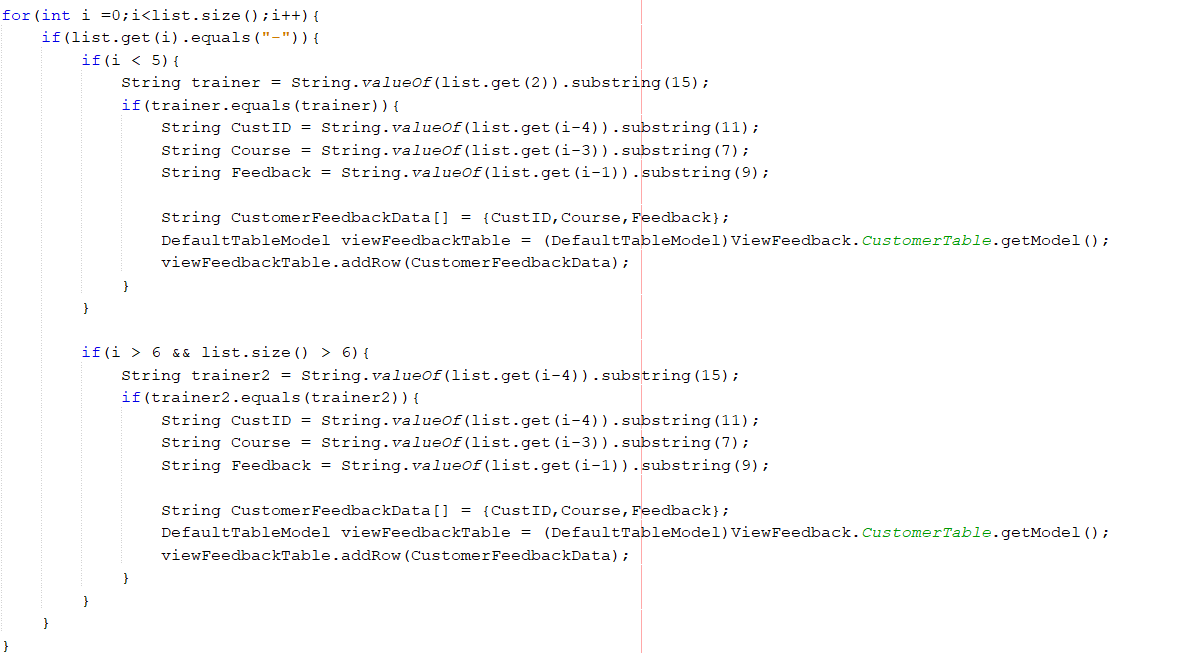
### **4.1 While Loop**



*Figure 8: While Loop in Trainer Class*

While Loop is similar to If statement where the condition is true, it will enter the While Loop and initiate the Looping until the condition become false or assign the break after the first Loop occur. Based on the Figure, we have assigned the value to an array and get the table. The condition of While Loop is the line of the text file has values, it will enter the Looping and after the first loop happen and then it will add the value to the row in the table, it will exit the While Loop since we have executed the break in the end of While Loop.

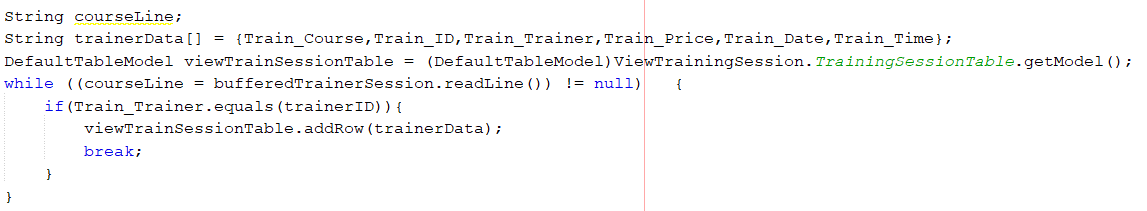
### **4.2 For Loop**



*Figure 9: For Loop in Trainer Class*

For Loop allow the looping to the specific times with the statement in the For Loop. In the Figure, the time of looping is dependent on the size of ArrayList. After the looping completed, it will exit from the For Loop.

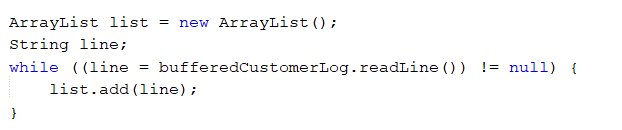
## **5.0 Array**



*Figure 10: Array in Trainer Class*

Array can be used to store several values in a same variable. In our program, we had been store the trainer information from the text file into a variable with the table and While Loop to display the trainer’s training session in a table.

## **6.0 ArrayList**

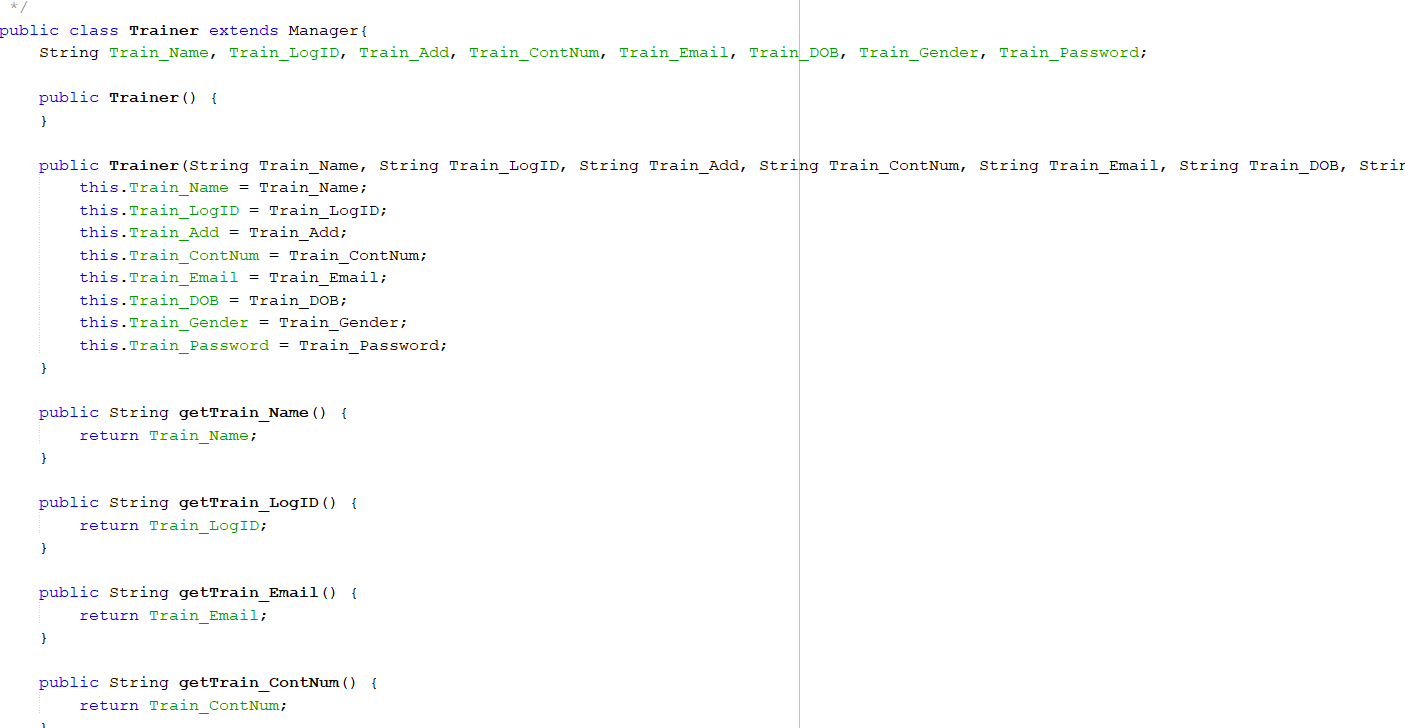


*Figure 11: ArrayList in Trainer Class*

ArrayList is different compared to Array due to ArrayList is flexible and the size is resizable, but the size of Array can’t be changed. Also, ArrayList allow us to add or remove the value anytime. In the java program, we executed the ArrayList for storing all the data from the text file.

## **7.0 OOPs**

### **7.1 Class**



*Figure 12: Trainer Class*

Class is specified by the user, and it is a combination of attributes or methods such as Get-Set method, trainer name variable and so on. Based on the Figure, it showed the trainer class and there are several attributes in this class such as trainer name, trainer login ID, and so on.

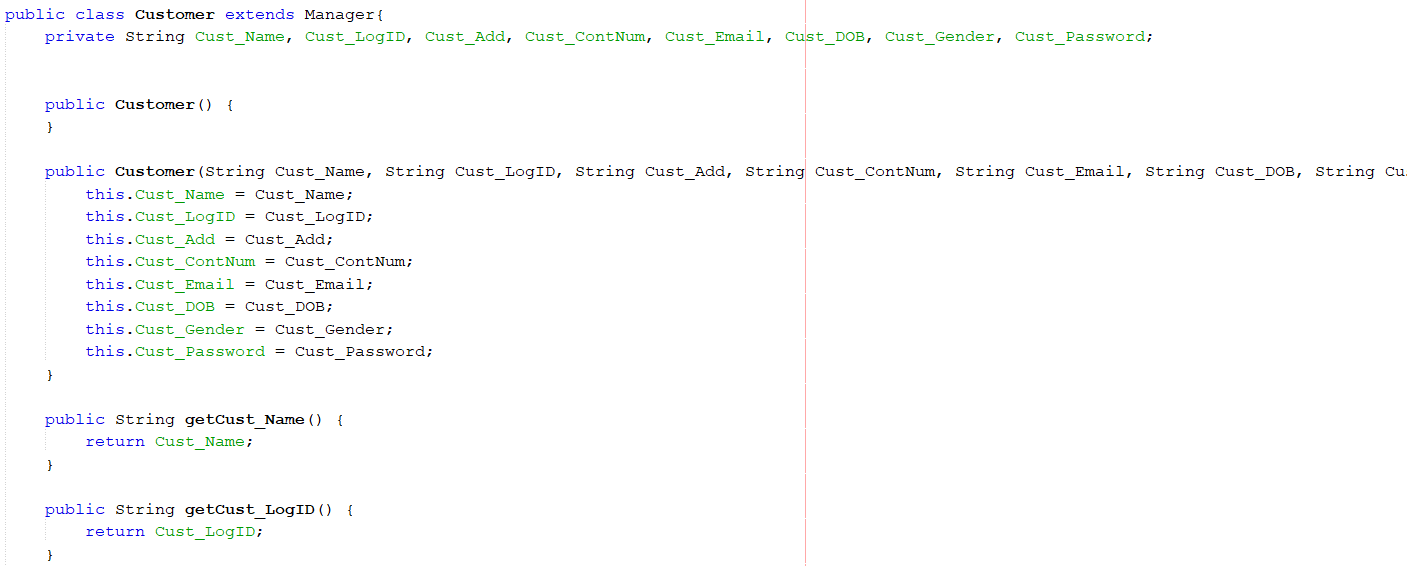
### **7.2 Object**



*Figure 13: Object in View Assigned Customer Page*

Object is executed the created class like Figure above in the page. The object will be used the default constructor method in the class (Hartman, 2020). After that, the object can also use the method in the class.

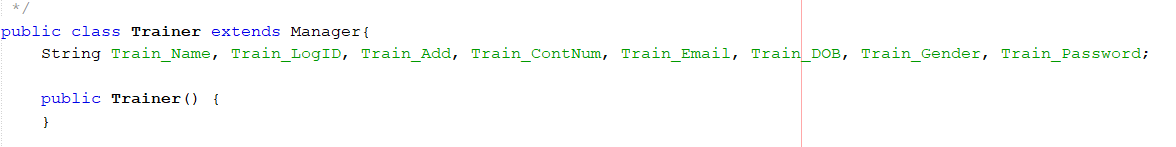
### **7.3 Encapsulation**



*Figure 14: Encapsulation in Customer Class*

Encapsulation is to combine the variables in one class and only the class object can access to the variables and methods (Java - Encapsulation, n.d.). In the Figure, the object that has been assigned can only access to the methods that are in their class since the method is public.

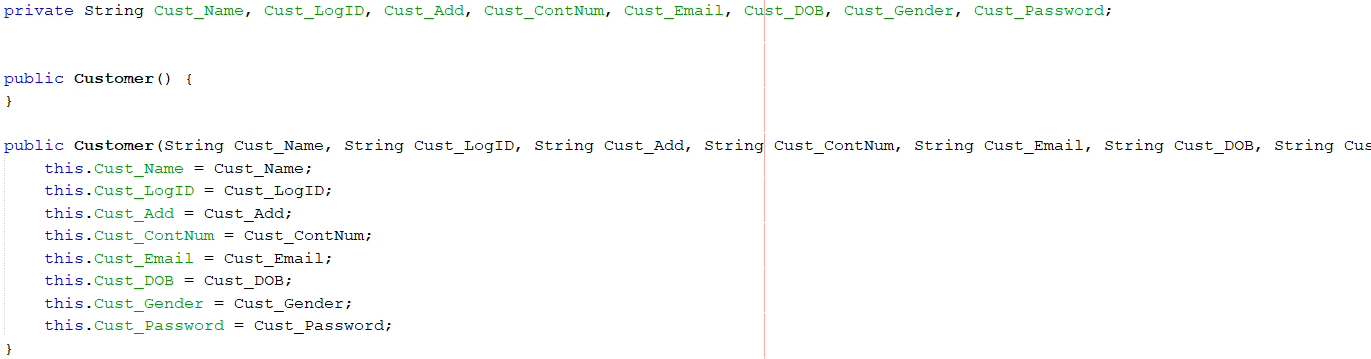
### **7.4 Inheritance**



*Figure 15: Inheritance in Trainer Class*

Inheritance is allowing to create a new class from the existing class (Java Inheritance, n.d.). It created as subclass which also called child class and superclass which called parent class. Based on the Figure, the trainer class is child class and manager class are parent class. For example, trainer object can execute the method in manager class since trainer class is the subclass.

### **7.5 Constructor**



*Figure 16: Constructor Method in Customer Class*

Constructor methods allow user to initiate the object. In the Figure, there are two constructor methods in customer class which is default constructor method and constructor parameter method. Constructor parameter method allow user to assign the value to the specific attributes.

### **7.6 Get-Set Method**



*Figure 17: Get-Set Method in Customer Class*

Get-Set method is very important in OOPs concept because it allow user to assign the value to the attributes by using setter method and allow user gets the value from the attributes via getter method for the object. User can also update the value of the attributes by using setter method.

### **7.7 Normal Method**



*Figure 18: View Trainer Method in Manager Class*

Based on the Figure 1, it showed one of the class methods in the manager class which is view trainer. This method is allowed manager to view all the trainer in the system and the trainers’ information will be display in the table.

### **7.8 Exceptional handling**

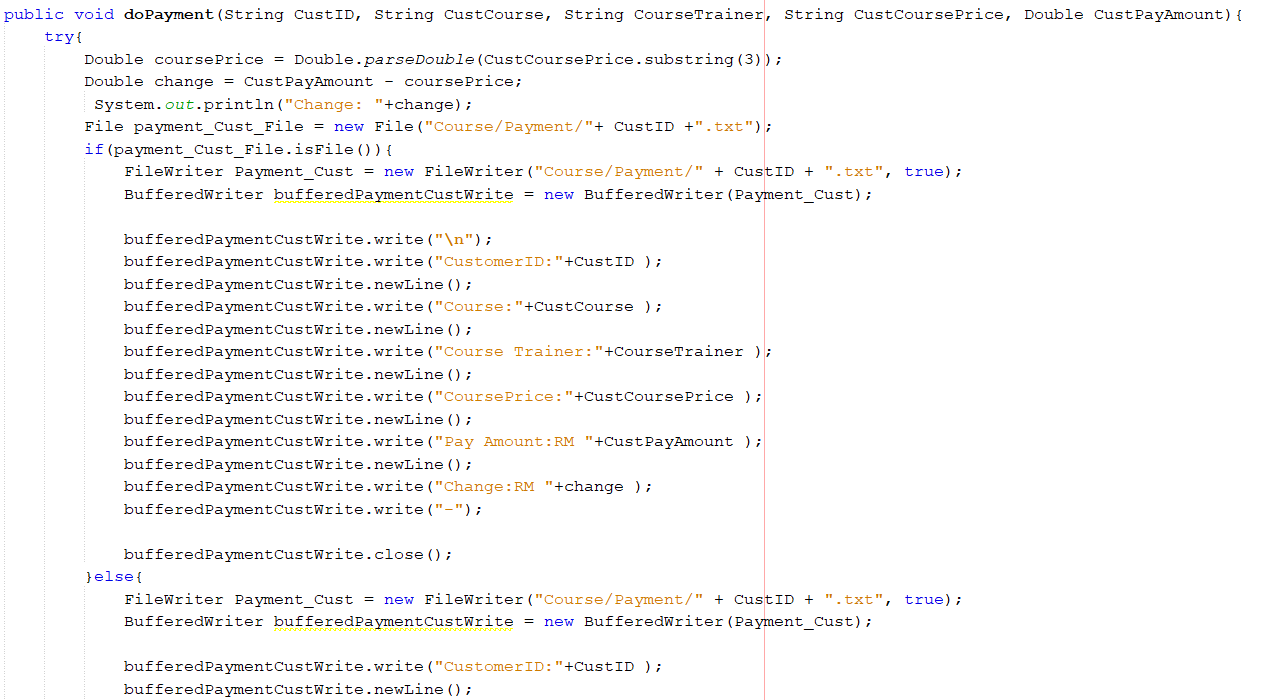


*Figure 19: Try-catch in Manager Class*

Exceptional handling is very important to avoid user’s input error and prevent the error input store into the text file. It also can capture the error and stop the operation of the coding. In the Figure, the search function has been used by the Try-catch method to avoid the error by the user.

## **8.0 File**

### **8.1 Write**



*Figure 20: Write File Method in Customer Class*



*Figure 21: Write File Method in Customer Class*

Based on these Figure above, the FileWriter and BufferedWriter method has been executed to the do payment for the customer to do the payment. It showed that the payment did by customer will be stored in the text file.

### **8.2 Read**

****

*Figure 22: Read File Method in Manager Class*

In this Figure, it showed that the FileReader and BufferedReader method has been executed for the login function to allow user to login to their account. These methods are used to read the data from the text file. After that, the value will be assigned to the variable and assigned to the object.

### **8.3 Search**

****

*Figure 23: Search Function in Manager Class*

Based on the Figure, the search customer and search trainer function has been completed to allow manager to search the account of customer and trainer. In this function will used FileReader and BufferReader to get the data from the text file and store the data into an array. After storing the data, it will be displayed in the table.

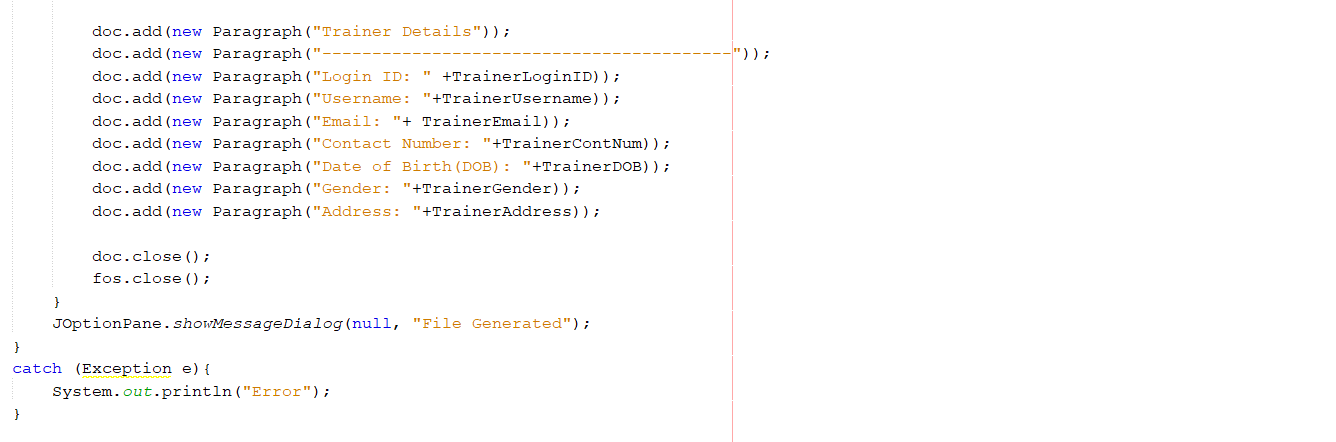
### **8.4 Generate Report**



*Figure 24: Generate Report Function in Manager*



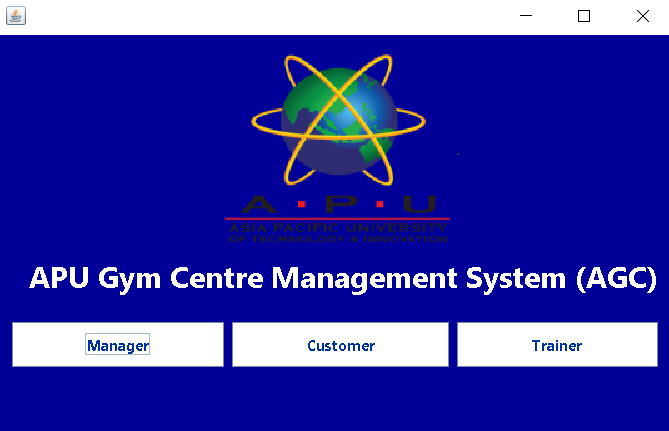
*Figure 25: Generate Report Function in Manager*



*Figure 26: Generate Report Function in Manager*

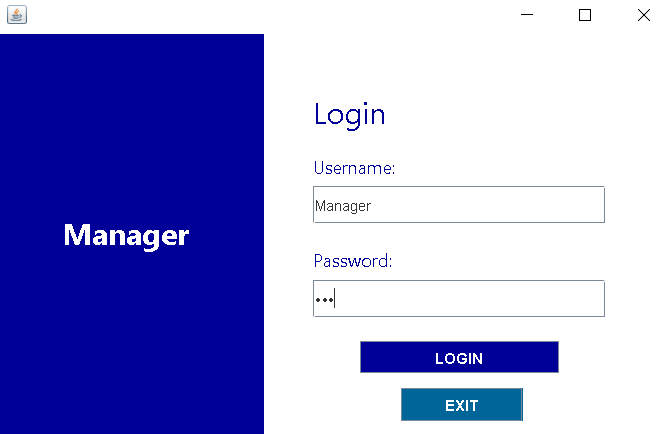
This function is to allow manager to export the customer and trainer account information to the PDF file. In this function, it will executed FileReader method, BufferedReader method, For Loop method, FileOutputStream method, Document method, and so on to generate a PDF file and store the data into the PDF file.

# **4.0 Output Screen Shot**



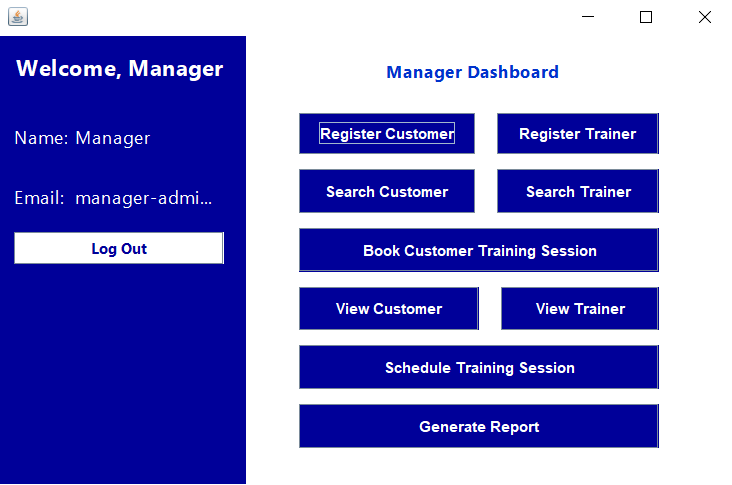
*Figure 27: Select Role Page of the APU Gym Centre Management System (AGC).*

From this page, a customer or trainer can find a manager to create an account within the system. Only the manager account needs to find the developer to create it. Each role can select their role from this login page to log in to the system. Anything or any action will be logged in before it is allowed to start.



*Figure 28: Manager login page of the APU Gym Centre Management System (AGC).*

From this figure, manager need type in correct and available username and password on the field. After that click the LOGIN button to go in system. If want to exit back to Select Role Page can click EXIT button.



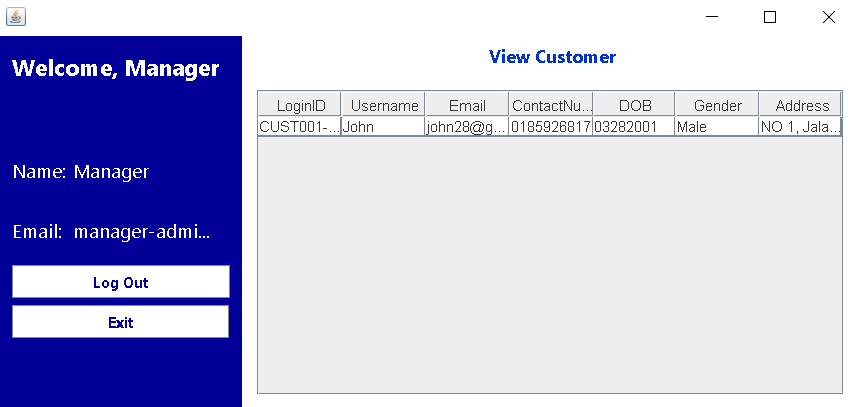
*Figure 29: Manager Dashboard of the APU Gym Centre Management System (AGC).*

After the manager login successful, he will see the interface same with figure the manager dashboard. He can choose any button from right hand side if he wants. If he wants to close the system just click Log Out button to closed.



*Figure 30: Register Customer of the APU Gym Centre Management System (AGC).*

When manager choose “Register Customer”, system will bring manage come to this page. This page allows manager help customer to create an account or register a APU gym member. Manager just need to fill up all information about customer and click “Submit” button to complete register step. If manager does not want to create can click "Exit" button from left hand side to go back Manager Dashboard.



*Figure 31: View All Customer of the APU Gym Centre Management System (AGC).*

After manager register new customer account, can click “View Customer” button from manager dashboard. From this view customer page, manager can see overall available member customer in the system. If manager want exit back to the manager dashboard can click “Exit” button from left hand side.



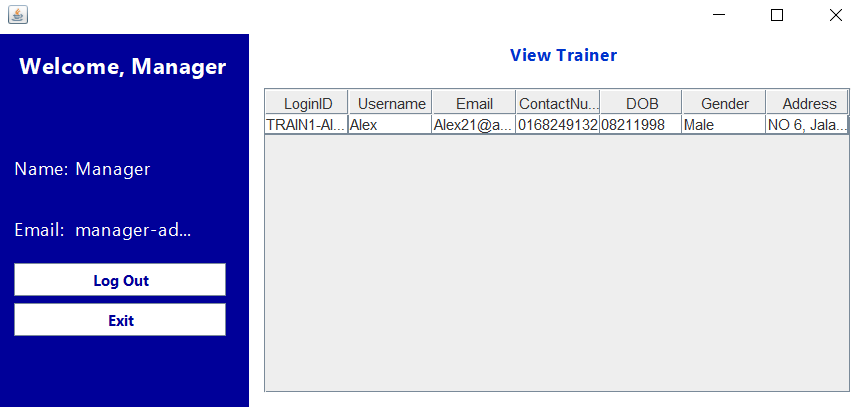
*Figure 32: Search Customer of the APU Gym Centre Management System (AGC).*

If manager want to search some member can select “Search Customer” button from manager dashboard. Then just key in the customer member Login ID on the search bar and click “Submit” button. If found the Login ID it will appear below table, if cannot found it means the Login ID are not available.



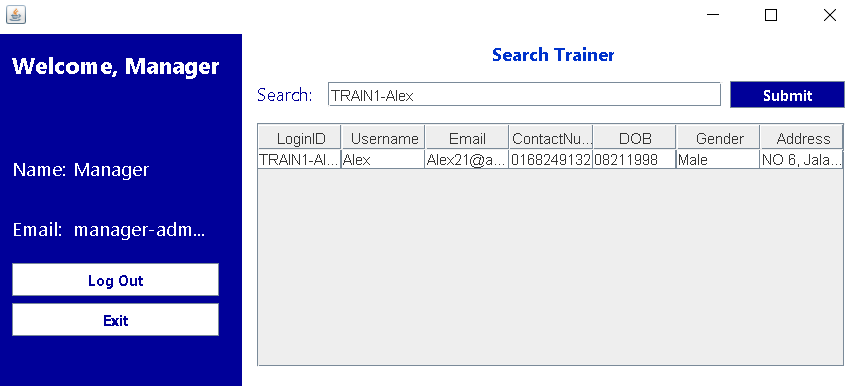
*Figure 33: Register Trainer of APU Gym Centre Management System (AGC).*

If manager choose “Register Trainer”, system will bring manage come to register trainer page. This page allows manager to create an account for APU gym trainer. Manager just need to fill up all information about trainer and click “Submit” button to complete register step. If manager does not want to create can click “Exit” button from left hand side to go back Manager Dashboard.



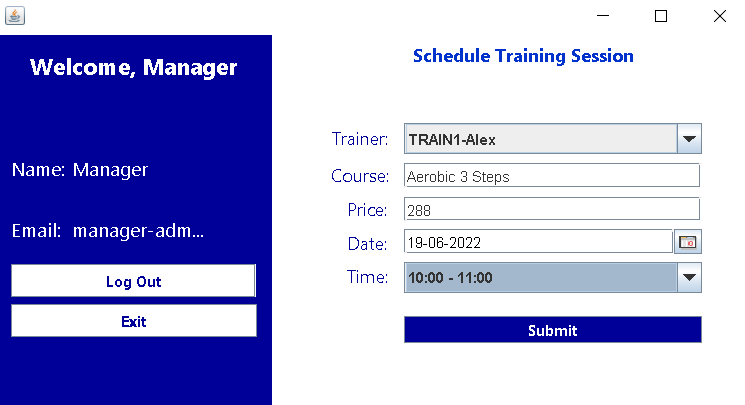
*Figure 34: View All Trainer of APU Gym Centre Management System (AGC).*

Managers can simply click the “View Trainers” button in the manager dashboard and link to this View All Trainers page. From this page, managers can see their profiles of all the trainers in the APU Gym Centre. If the manager wants to exit the manager dashboard, they can click the “Exit” button on the left.



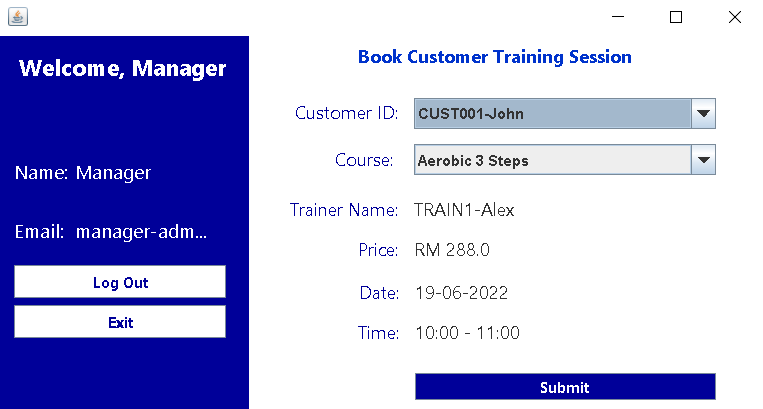
*Figure 35: Search Trainer of the APU Gym Centre Management System (AGC).*

If a manager wants to search for a trainer, he can select the “Search Trainer” button from the manager dashboard. Then just enter the trainer's login ID in the search bar and click the submit button. If a login ID is found, all information about the trainer is displayed in the table below, if not, either the login ID is unavailable or the wrong type of login ID.



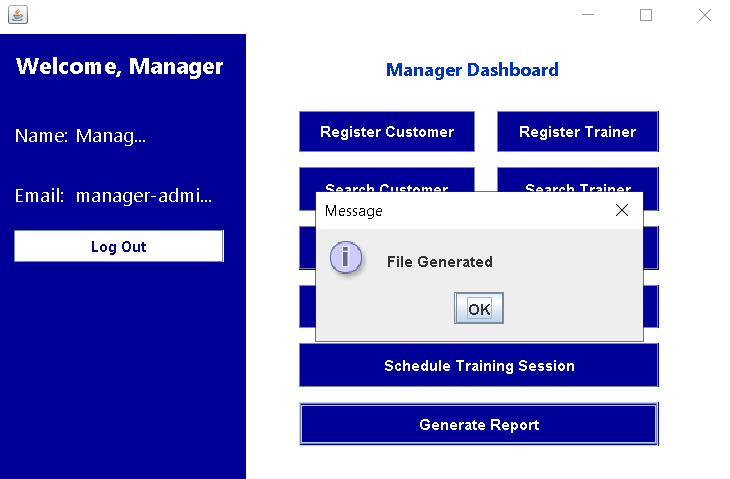
*Figure 36: Schedule Training Session of APU Gym Centre Management System (AGC).*

Suppose the manager wants to help the trainer schedule or create a new course, he will help the trainer through this “Schedule Training Session”. After entering this page, the manager needs to choose which trainer to create for, the name of the course, the price, and the time and duration.



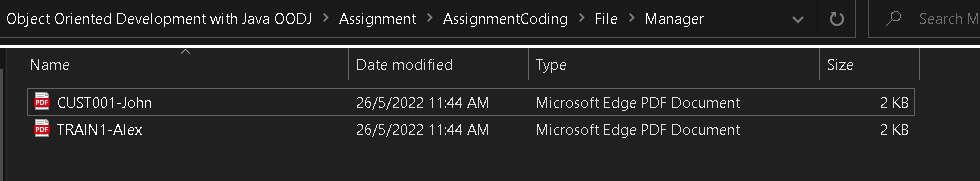
*Figure 37: Book Customer Training Session of APU Gym Centre Management System (AGC).*

If a customer member wants to book Trainer Alex's aerobic 3-step course, the manager needs to book through “Book Customer Training Session”. From this page, just select the customer and the course required by the customer, and then all the information about the course will be displayed below.



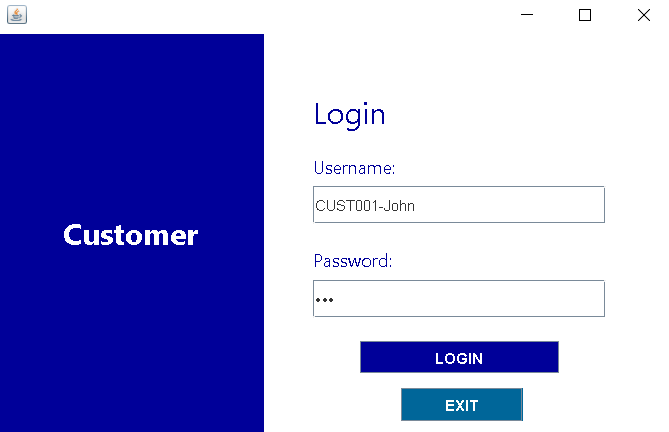
*Figure 38: Generate Report of the APU Gym Centre Management System (AGC).*

The manager clicks “Generate Report”, and the system will automatically generate PDF files of all customers and trainers. The PDF file will be generated with the customer and trainer account information.



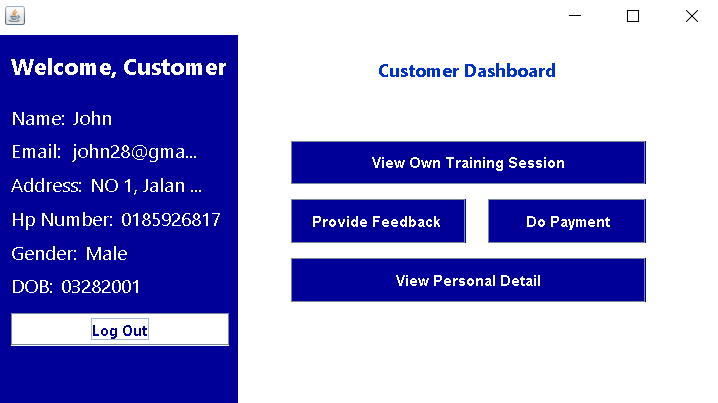
*Figure 39: PDF report location where report generation functions are stored.*

These PDFs will automatically be stored in the Manager folder of the File folder.



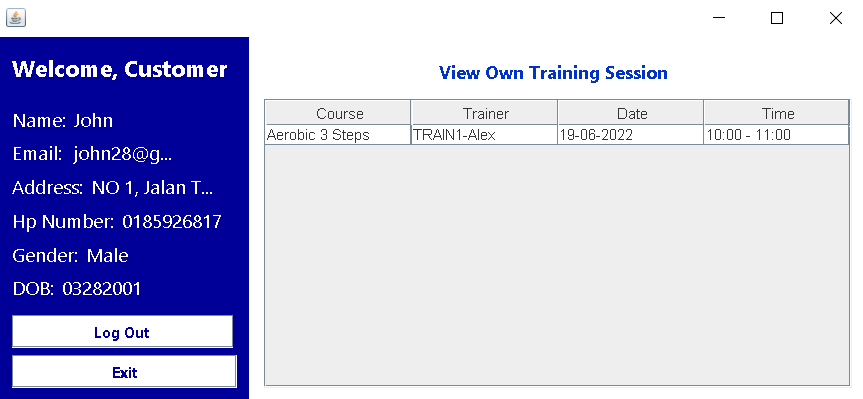
*Figure 40: Customer login page of the APU Gym Centre Management System (AGC).*

This page is allowed customer type in correct and available username and password on the field. After that click the LOGIN button to go in system. If customer still have not account can find manager to help register an account. If want to exit back to Select Role Page can click EXIT button.



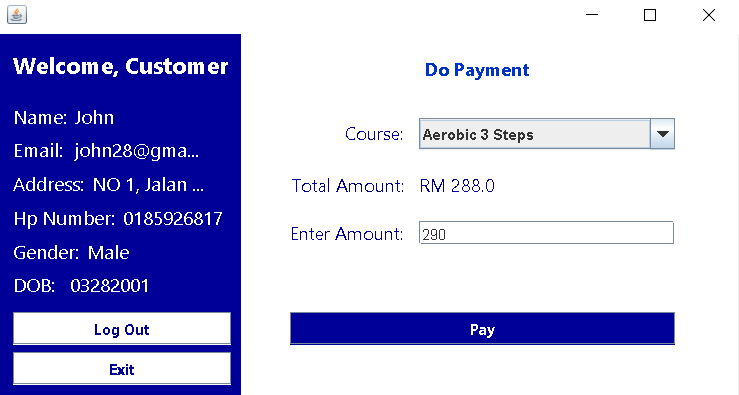
*Figure 41: Customer dashboard of the APU Gym Centre Management System (AGC).*

After login customer can see the interface left hand side is his personal information and log out button, right hand side will show some buttons allow to some action. Customer can choose any button from right hand side if he wants.



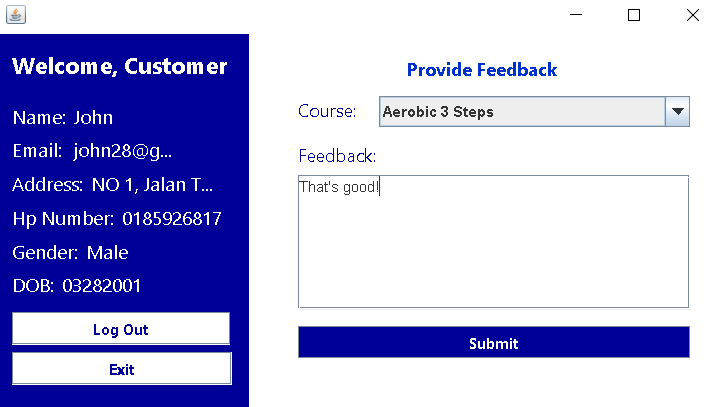
*Figure 42: View Own Training Session of APU Gym Centre Management System (AGC).*

If the customer clicks the “View Own Training Session” button, this page will display what courses the customer booked and the time on the table. If customer want to go back to dashboard can click “Exit” button from left hand side.



*Figure 43: Do Payment of the APU Gym Centre Management System (AGC).*

The system provides payment function, customers only need to select “Do Payment” from the customer dashboard. Then choose which course you want to pay for and add the payment amount. After confirmation, you can click the “Pay” button to complete the payment. It should be noted that the customer must first help the Manager to book the course before the available courses will be displayed.



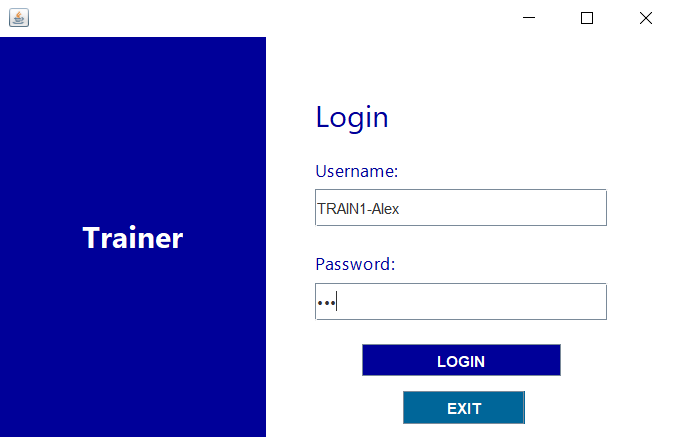
*Figure 44: Provide feedback of APU Gym Centre Management System (AGC).*

After completing the payment course, customers can go to “Provide Feedback” to provide feedback. Just choose which course to fill in the feedback, and then submit.



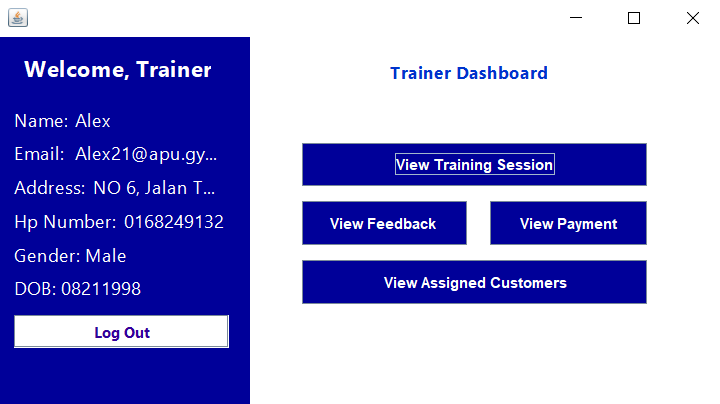
*Figure 45: View Personal Detail of APU Gym Centre Management System (AGC).*

Customers can also click the “View Personal Detail” button, through this page can see their personal information.



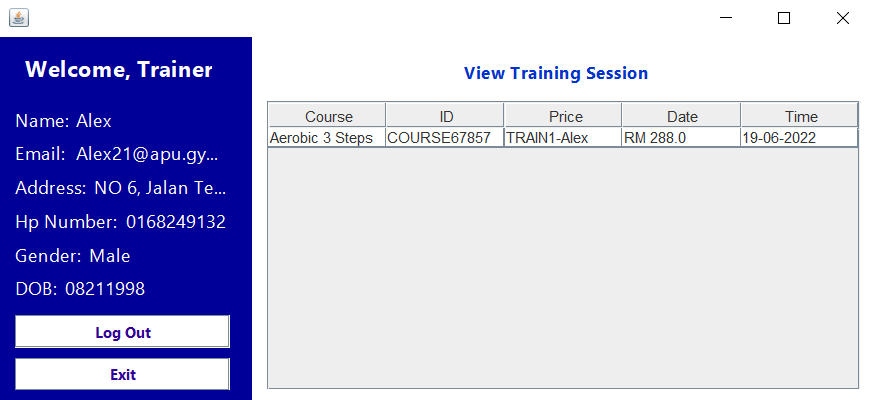
*Figure 46: Trainer Login Page of the APU Gym Centre Management System (AGC).*

This page is allowed trainer to enter in correct and available username and password on the field. After that click the LOGIN button to go in system. If trainer still have not account can find manager to help register an account. If want to exit back to Select Role Page can click EXIT button.



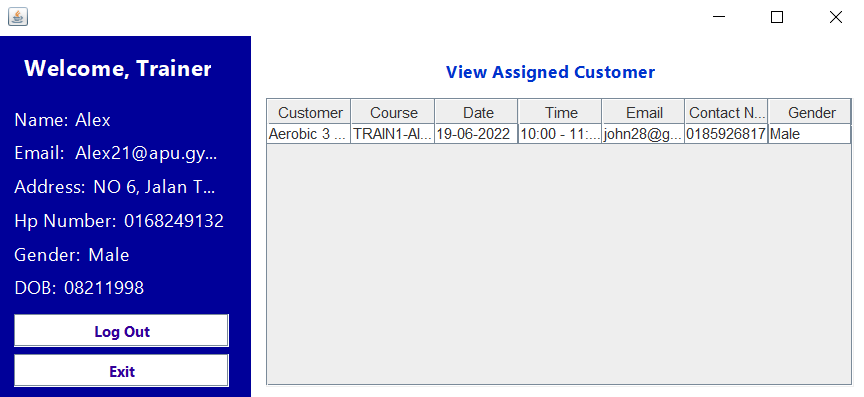
*Figure 47: Trainer Dashboard of the APU Gym Centre Management System (AGC).*

After trainer login can see the interface left hand side is his personal information and log out button, right hand side will show some buttons allow to some action. Trainers use any buttons to manage they are training.



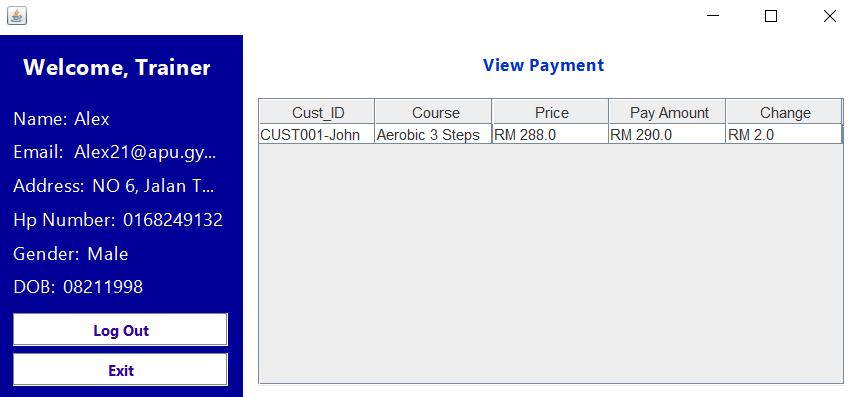
*Figure 48: View Training Session of APU Gym Centre Management System (AGC).*

From the View Training Session page, the trainer can see which courses and specific information the manager has created for the trainer.



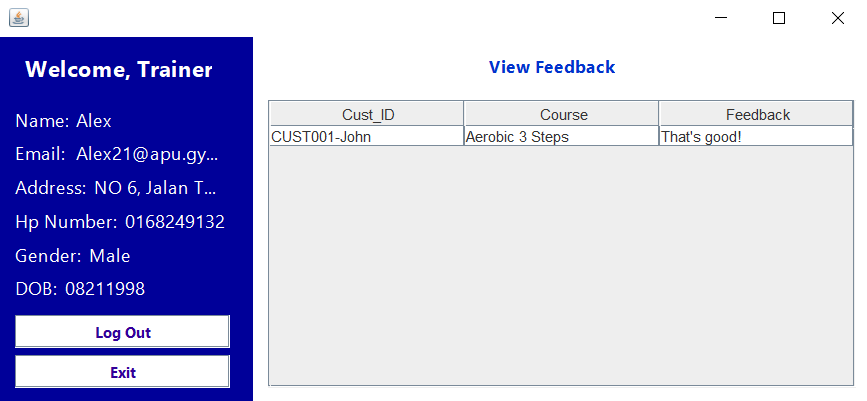
*Figure 49: View Assigned Customer of the APU Gym Centre Management System (AGC).*

After the trainer clicks the “View Assigned Customer” button, this page will display the customers who have booked his class and the customer's contact information.



*Figure 50: View Payment of the APU Gym Centre Management System (AGC).*

Then the Trainer can select “View Payment” to see which customers have paid back, and these customers will be displayed in the table on the page.



*Figure 50: View Feedback of the APU Gym Centre Management System (AGC).*

After that, if a customer fills in the feedback, the trainer can check which customer fills in the feedback and the content of the feedback through “View Feedback”.

# **5.0 Conclusion**

In this APU Gym Centre Management System project, we have learned how to plan and arrange the functions that are needed and should exist in the system through Use Case and Class Diagram. The use Case and Class Diagram clearly allow us to understand what function we should do and add in this project. Furthermore, we also learned to use Java's JFrame functionality and the usage of some techniques or features including Control Structure, Looping Structure, ArrayList, Exceptional Handling, File usage, the project's most important object-oriented programming concepts, and so on to operate in the project. Finally, we also learned to prevent some common user problems through programs, whether it is user error, not numeric value and so on. In this assignment, we have learned problem-solving skill and teamwork skill when we are doing this assignment’s program and report. We are having a conversation and have a well communication to finish this assignment on time and ensure that the program didn’t have any bugs as well. When we are facing the problem, we will try to solve the problem together.

# **6.0 References**

1. Hartman, J. (2020). What is Class and Object in Java OOPS. Www.guru99.com. <https://www.guru99.com/java-oops-class-objects.html#:~:text=An%20object%20in%20OOPS%20is>
2. Java - Encapsulation. (n.d.). Www.tutorialspoint.com. <https://www.tutorialspoint.com/java/java_encapsulation.htm#:~:text=Encapsulation%20in%20Java%20is%20a>
3. Java Inheritance. (n.d.). Www.programiz.com. Retrieved May 26, 2022, from <https://www.programiz.com/java-programming/inheritance#:~:text=Inheritance%20is%20one%20of%20the>