Tunku Abdul Rahman University College

AACS2204
Object-Oriented Programming Techniques

Assignment
2021/2022

Programme : Diploma in Information Technology

Tutorial Group : G5

Date Submitted to Tutor : 17/09/2021

Team Members:

No	Student Name	Student ID
1.	Lim En Xi	20PMD05947
2.	Tan Lin Yi	20PMD05909
3.	Alvin Chan Ee Aun	20PMD11351
4.	Vinnie Chin Loh Xin	20PMD05307

No.	Team Member	Task(s) Allocated
1.	Lim En Xi	flight + ticket (super class) + reservation + transaction
2.	Tan Lin Yi	flight + ticket FirstClass + report
3.	Alvin Chan Ee Aun	flight + ticket EconomyClass + payment
4.	Vinnie Chin Loh Xin	flight + ticket BusinessClass + customer

Coursework Declaration

We confirm that we have read and shall comply with all the terms and conditions of TAR University College's plagiarism policy.

We declare that this assignment is free from all forms of plagiarism and for all intents and purposes is our own properly derived work.

Signature :				
Name :	Lim En Xi	Tan Lin Yi	Alvin Chan Ee Aun	Vinnie Chin Loh Xin
Date :	17 / 09 / 2021	17 / 09 / 2021	17 / 09 / 2021	17 / 09 / 2021

Table of Contents

UML diagram

Flight

- _flightNo: String
- _departLocation: String
- _arriveLocation: String
- _tickets: ArrayList<Ticket>
- _firstClassQty: int
- _businessClassQty: int
- _economyClassQty: int
- _firstClassPrice: double
- _businessClassPrice: double
- _economyClassPrice: double
- _isCancelled: boolean

+ Flight(flightNo: String, departLoc: String, arriveLoc: String, firstClassQty: int, firstClassPrice: double, businessClassQty: int, businessClassPrice: double, economyClassQty: int, economyClassPrice: double)
+ getDepartLocation(): String
+ getArriveLocation(): String
+ getFirstClassQty(): int
+ getBusinessClassQty(): int
+ getEconomyClassQty(): int
+ getFlightNo(): String
+ getFirstClassPrice(): double
+ getBusinessClassPrice(): double
+ getEconomyClassPrice(): double
+ getTickets(): ArrayList<Ticket>
+ getBusinessClassTickets(): ArrayList<Ticket>
+ getFirstClassTickets(): ArrayList<Ticket>
+ getEconomyClassTickets(): ArrayList<Ticket>
+ AddTicket(ticket: Ticket): void
+ Cancel(): void
+ IsCancelled(): boolean
+ DrawFlight(): void
+ DrawFlightWithDetail(): void

Customer

- _customerName: String
- _customerPassportNo: String
- _customerPhone: String
- _tickets: ArrayList<Ticket>

+ Customer(customerName: String , customerPassportNo:String, customerPhone: String)
+ getCustomerName(): String
+ getCustomerPassportNo(): String
+ getCustomerPhone(): String
+ Tickets: ArrayList<Ticket>
+ AddTicket(Ticket ticket): void
+ setCustomerName(customer Name: String): void
+ setCustomerPassportNo(customerPassportNum: String): void
+ setCustomerPhone(customerPhone: String): void
+ DrawCustomer(): void
+ DrawCustomerWithDetail(): void

Transaction

- _customer: Customer
- _ticket: Ticket
- _payment: Payment
- _reservation: Reservation

+ Transaction(customer: Customer, ticket: Ticket, payment: Payment, reservation: Reservation)
+ getCustomer(): Customer
+ getTicket(): Ticket
+ getPayment(): Payment
+ getReservation(): Reservation

Reservation

- _payment: Payment
- _ticket: ArrayList<Ticket>
- _flight: Flight
- _customer: Customer
- _reservationNo: String
- _reservationNoCounter: int

+ Reservation(customer: Customer , payment: Payment, flight: Flight, tickets: ArrayList<Ticket>)
+ NextReservationNo(): String
+ getReservationNo(): String
+ getPayment(): Payment
+ getFlight(): Flight
+ getCustomer(): Customer
+ getTickets(): ArrayList<Ticket>
+ setReservationNo(reservationNo: String): void
+ DrawReservation(): void
+ DrawReservationWithDetail(): void

Payment

- _price: double
- _paid: double

+ Payment(price: double, paid: double)
+ getPrice(): double
+ getPaid(): double
+ Change(): double

Ticket

_ticketNo: String
_flight: Flight
- _isSold: boolean = false
- _isCancelled: boolean = false

+ Ticket(flight: Flight, ticketCounter: int)
+ TicketType(): String
+ TicketDescription(): String
# TicketPrefix(): String
# TicketPrice(): double
+ AddToFlight(): void
+ getTicketNo(): String
+ getFlight(): Flight
+ AssignToCustomer(customer: Customer): void
+ IsCancelled(): boolean

TicketFirstClass

+ TicketFirstClass(flight: Flight, ticketCounter: int)
+ TicketType(): String
+ TicketDescription(): String
# TicketPrefix(): String
+ TicketPrice(): double

TicketBusinessClass

+ TicketBusinessClass(flight: Flight, ticketCounter: int)
+ TicketType(): String
+ TicketDescription(): String
# TicketPrefix(): String
+ TicketPrice(): double

TicketEconomyClass

+ TicketEconomyClass(flight: Flight, ticketCounter: int)
+ TicketType(): String
+ TicketDescription(): String
# TicketPrefix(): String
+ TicketPrice(): double

Figure 1 shows UML diagram of flight reservation system.

Description of Ideas

- ✓ This system user is the **staff** that help customers to reserve and purchase tickets.

- ✓ Our system allows the receptionists to add, view, and delete flight details.

- ✓ The receptionists can add, view, update and delete customer details.

- ✓ A flight's tickets have different classes like first class, business class, and economy class.

- ✓ Each class of tickets has a different price, different style of ticket number.

- ✓ A customer can reserve and purchase many flight tickets at once.

- ✓ The customer must make the payment after the reservation of flight tickets.

- ✓ The customer is not allowed to cancel the reservation and payment after paying.

- ✓ There is a ticket status showing whether the tickets of a flight are still available or have been sold.

- ✓ If the flight is deleted, the bought tickets status will change to "cancelled".

Screenshots of system

Main Menu

Figure 2: shows the main menu validation 2
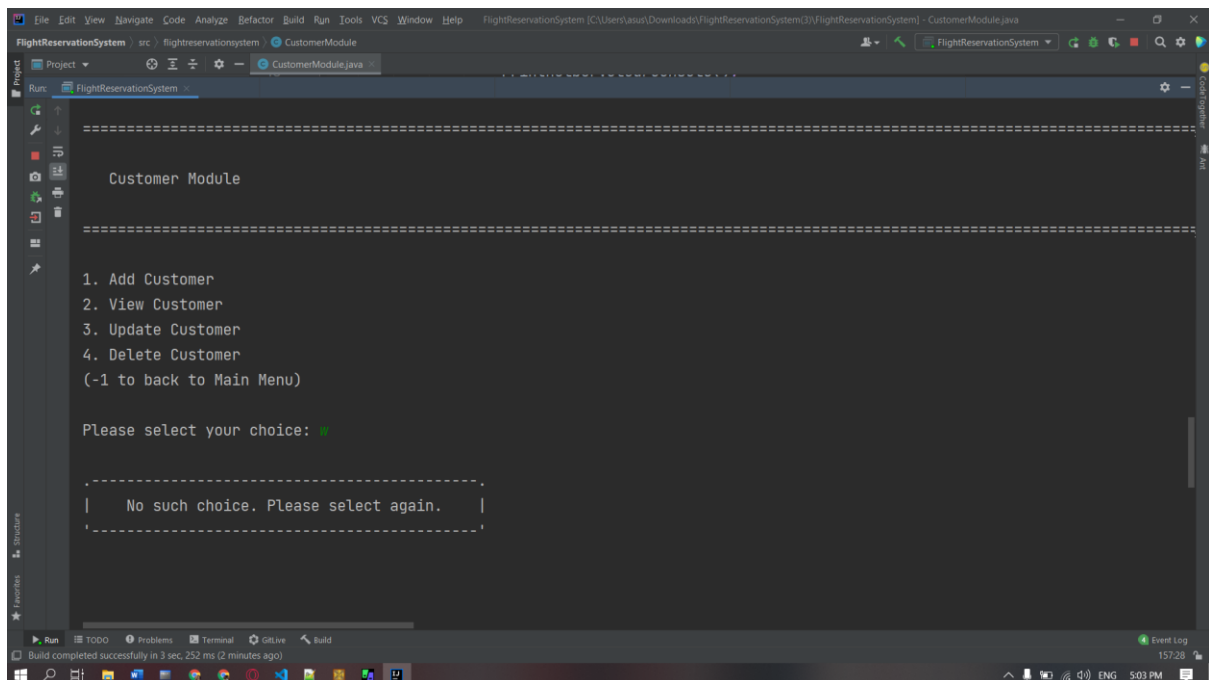
Figure 3: shows the main menu validation 2

Customer Module

Figure 4: customer menu with actions can be made

Figure 5: customer menu with actions can be made

The user can choose to add, view, update and delete customer details, if the user has entered a choice that is not included in the menu, they will be prompted and asked to choose again.
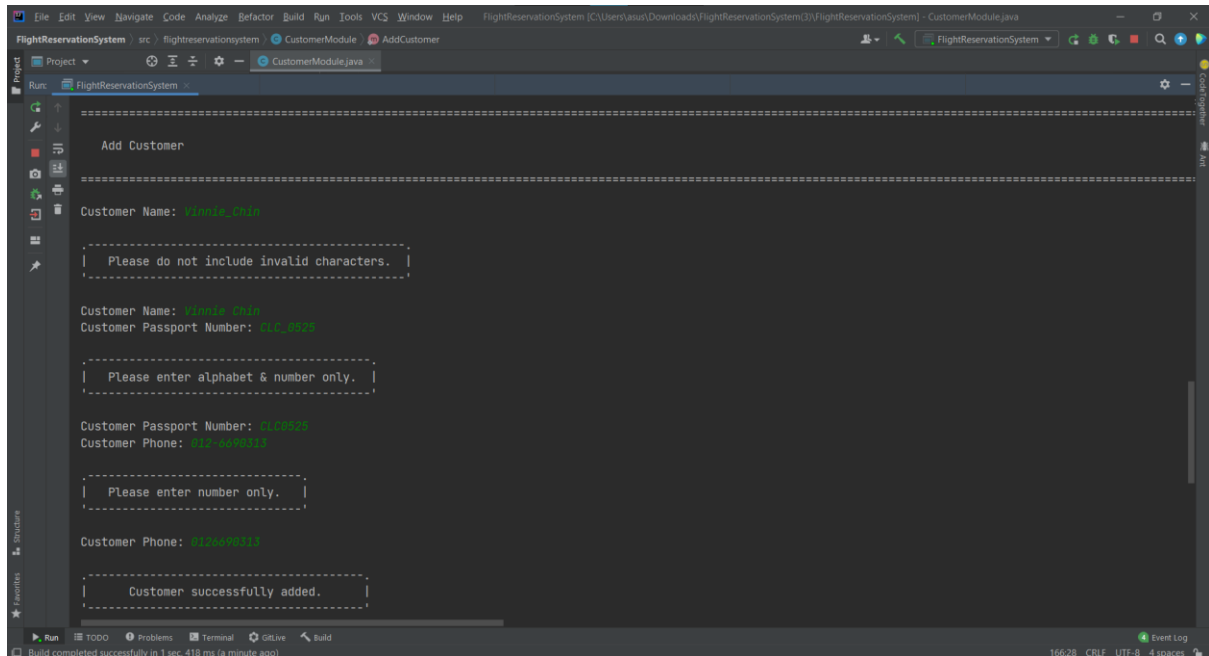
Figure 6: shows add customer details

There are validation checks for every field of customer details. For customer names, it is not allowed to have invalid characters. Only alphabet and numbers are allowed in passport numbers, and for phone numbers everything is not allowed except digit.
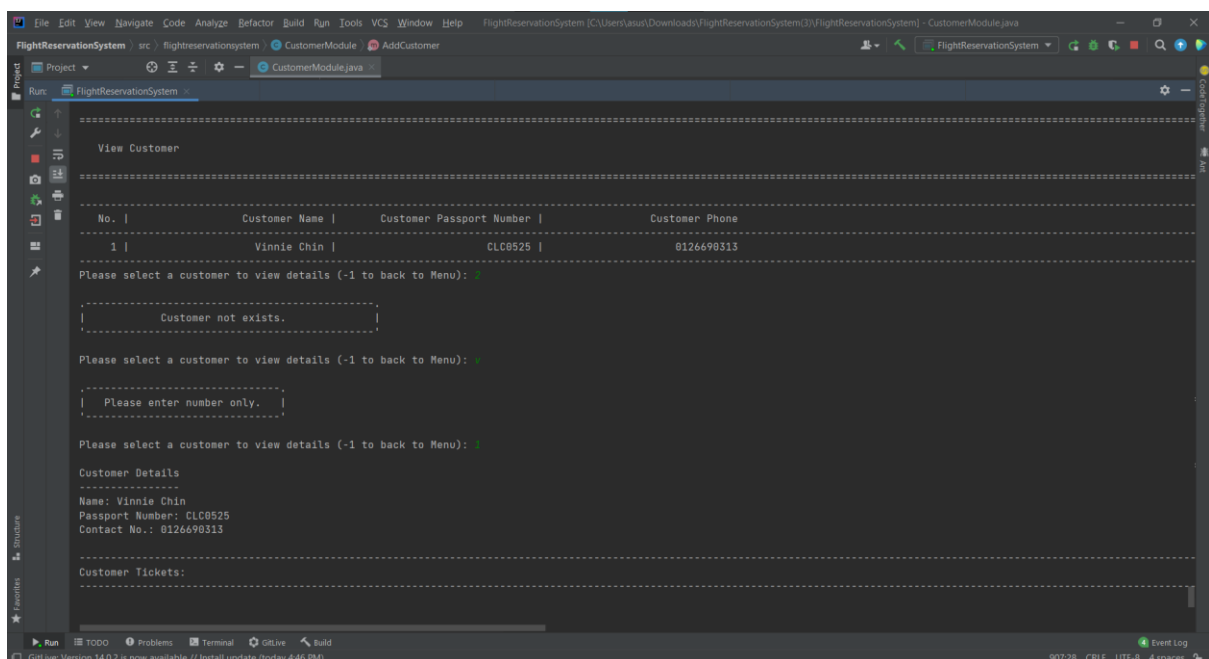
Figure 6: shows view customer details

When the user wants to view customer details, he or she can select the customer with a number that is listed in the list to view. If the user selects the user that is not in the list, the user will be prompted that the customer selected does not exist. Also, when the user has entered the alphabet there will be a prompt telling the user to enter a number (number in the customer list) only. Customer tickets are not showing anything yet as the customer didn't make any flight reservation yet.

Figure 7: shows update customer details

When the user wants to update customer details, he or she can select the customer with a number that is listed in the list and make changes to the customer details. The validation checking on this part is similar to the validation checking of the add customer section.

Figure 8: shows new updated customer details

After updating the customer details, the user can go to view customer details section to view the changes made previously. However, there will be validation checking such as making sure that the inputted values are numerical, and the value entered exists.

```
================================================================================

  View Customer

================================================================================


--------------------------------------------------------------------------------
  No. |                  Customer Name |     Customer Passport Number |              Customer Phone
--------------------------------------------------------------------------------
   1 |            Vinnie Chin Loh Xin |                    LCVC0719 |                0123456789
--------------------------------------------------------------------------------
Please select a customer to view details (-1 to back to Menu): 1

Customer Details
----------------
Name: Vinnie Chin Loh Xin
Passport Number: LCVC0719
Contact No.: 0123456789


--------------------------------------------------------------------------------
Customer Tickets:
Flight No: MH333    Ticket Type: Economy Class    Ticket No.: MH333_EC015    Ticket Status: Available
Flight No: MH333    Ticket Type: Economy Class    Ticket No.: MH333_EC014    Ticket Status: Available
Flight No: MH333    Ticket Type: Economy Class    Ticket No.: MH333_EC004    Ticket Status: Available
Flight No: MH333    Ticket Type: Economy Class    Ticket No.: MH333_EC013    Ticket Status: Available
Flight No: NUI094   Ticket Type: First Class      Ticket No.: NUI094_FC001   Ticket Status: Available
Flight No: NUI094   Ticket Type: First Class      Ticket No.: NUI094_FC002   Ticket Status: Available
Flight No: NUI094   Ticket Type: First Class      Ticket No.: NUI094_FC003   Ticket Status: Available
--------------------------------------------------------------------------------
```

Figure 9: shows customer details with tickets reserved after making flight reservation

After the user finishes the booking payment of flight reservation for the customer, the tickets purchased will be shown in the customer tickets under customer details. As you can see from the screenshot above, there are few ticket bookings for two different flights showing each of the ticket details recorded in the customer record.

```
========================================================================================

   View Customer

========================================================================================


-----------------------------------------------------------------------------------------
   No. |              Customer Name |     Customer Passport Number |            Customer Phone
-----------------------------------------------------------------------------------------
    1 |          Vinnie Chin Loh Xin |                    LCVC0719 |                0123456789
-----------------------------------------------------------------------------------------
Please select a customer to view details (-1 to back to Menu): 1

Customer Details
----------------
Name: Vinnie Chin Loh Xin
Passport Number: LCVC0719
Contact No.: 0123456789


-----------------------------------------------------------------------------------------
Customer Tickets:
Flight No: MH333   Ticket Type: Economy Class   Ticket No.: MH333_EC015   Ticket Status: Available
Flight No: MH333   Ticket Type: Economy Class   Ticket No.: MH333_EC014   Ticket Status: Available
Flight No: MH333   Ticket Type: Economy Class   Ticket No.: MH333_EC004   Ticket Status: Available
Flight No: MH333   Ticket Type: Economy Class   Ticket No.: MH333_EC013   Ticket Status: Available
Flight No: NUI094  Ticket Type: First Class     Ticket No.: NUI094_FC001  Ticket Status: Cancelled
Flight No: NUI094  Ticket Type: First Class     Ticket No.: NUI094_FC002  Ticket Status: Cancelled
Flight No: NUI094  Ticket Type: First Class     Ticket No.: NUI094_FC003  Ticket Status: Cancelled
-----------------------------------------------------------------------------------------
```

Figure 10: shows ticket status of a flight that has been deleted

As you can see from the screenshot above, the flight ticket for the flight number of NIU094, the ticket status for it is cancelled. This is because the flight is deleted, the ticket status of that flight will be changed to "cancelled".

```
=============================================================================================9==|=P=a=g=e====

   Delete Customer

=============================================================================================================

-------------------------------------------------------------------------------------------------------------
   No. |                    Customer Name |     Customer Passport Number |               Customer Phone
-------------------------------------------------------------------------------------------------------------
     1 |             Vinnie Chin Loh Xin |                      LCVC0719 |                  0123456789
-------------------------------------------------------------------------------------------------------------
Please select a customer to delete (-1 to back to Menu): 3

.----------------------------------------.
|          Customer not exists.          |
'----------------------------------------'

Please select a customer to delete (-1 to back to Menu): vinnie

.-----------------------------.
|   Please enter number only. |
'-----------------------------'

Please select a customer to delete (-1 to back to Menu): 1

.--------------------------------------.
|      Customer successfully deleted.  |
'--------------------------------------'
```

Figure 11: shows delete customer

The user is able to delete the customer record at the delete customer section. From there, the user is required to select the customer number to delete their record. There will be validation checking to make sure that the number entered for the customer record exists and also the value entered is numerical. After choosing the right customer to delete, the system will prompt a success message showing that the customer record has been successfully deleted.

```
=============================================================================================================

   View Customer

=============================================================================================================

-------------------------------------------------------------------------------------------------------------
   No. |                    Customer Name |     Customer Passport Number |               Customer Phone
-------------------------------------------------------------------------------------------------------------
Please select a customer to view details (-1 to back to Menu): 1

.----------------------------------------.
|          Customer not exists.          |
'----------------------------------------'

Please select a customer to view details (-1 to back to Menu): -1
```

Figure 12: shows deleted customer details has been removed from the customer list

And from here, the user is able to go to the view customer record to see the changes made. And as you can see from the screenshot above, the customer that existed before is no longer there as it has been deleted earlier from the function above.

Flight Module

Figure 13: shows Flight Menu Validation 1

Figure 14: Flight menu validation 2

The flight menu has the validation restrict the not in listed character (1, 2, 3, -1).

Figure 15: Add Flight 1

Figure 16: Add Flight 2

The user needs to key in the flight number, departure and arrival location. As every flight does have first class, business class, and economy class type, The user needs to key in the quantity and price of each class type.
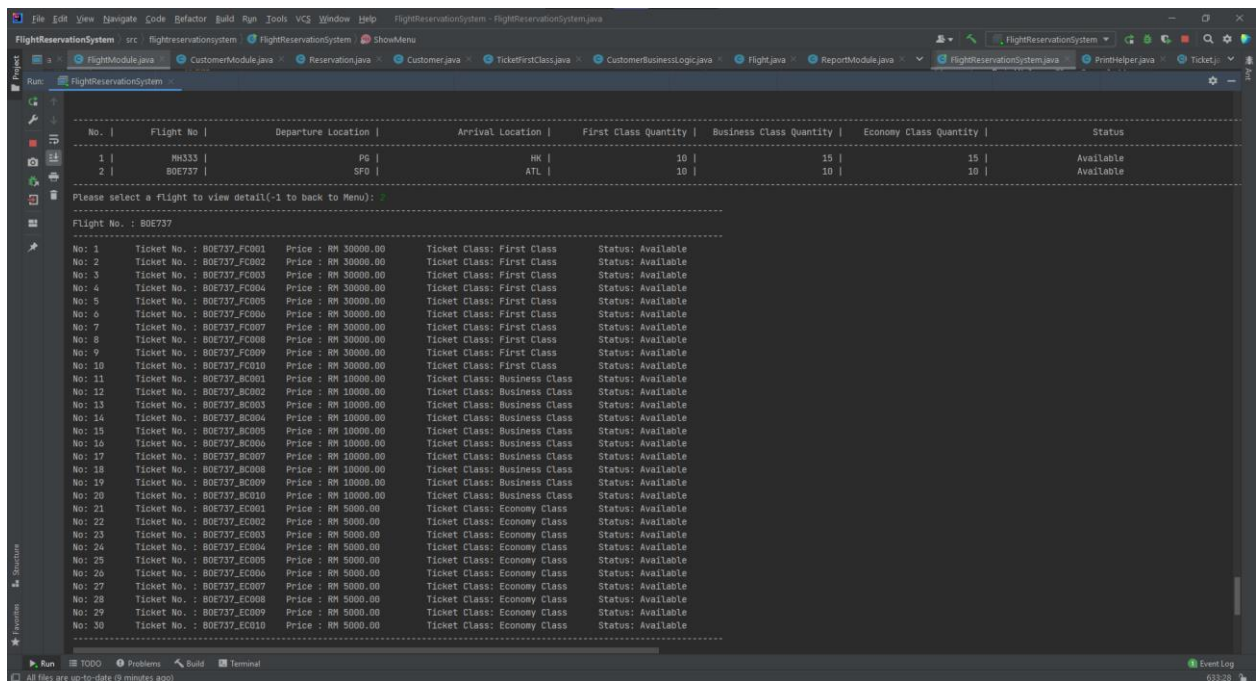
Figure 17: View flight and flight details

When the user enters view flight, the user can choose the number to view the flight detail(tickets).
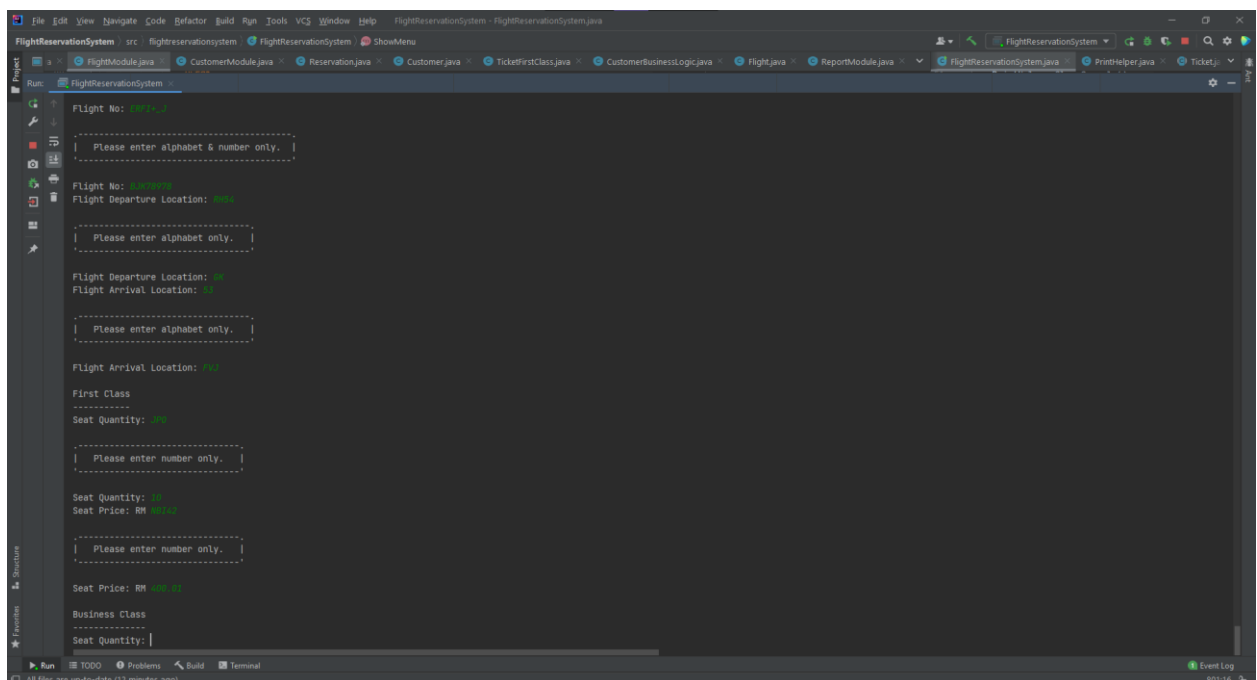
Figure 18: Flight validation

Here is the add flight validation. The flight number can only contain the alphabet and number, no symbol allowed. The location only allows the alphabet. The seat and price are not allowed for any alphabet and symbol.

Figure 19: Delete flight

The user can choose the number to delete the flight.

Figure 20: Choose Flight validation

The validation of checking flight exists, and no number is applied on every process of choosing a flight.

Reservation Module

Figure 21: Reservation module validation for number

Figure 22: Reservation module validation for alphabet

Figure 23: Reservation module validation for symbol

After the user enters "3" in the main menu, the reservation module menu will be displayed. Then users can choose to add reservation or view reservation by entering "1" for add and "2" for view. "-1" is for the user to go back to the main menu. If the user enters an

alphabet, symbol, or another number other than "1", "2" or "-1", a validation message will be displayed to let the user know there is no such choice.
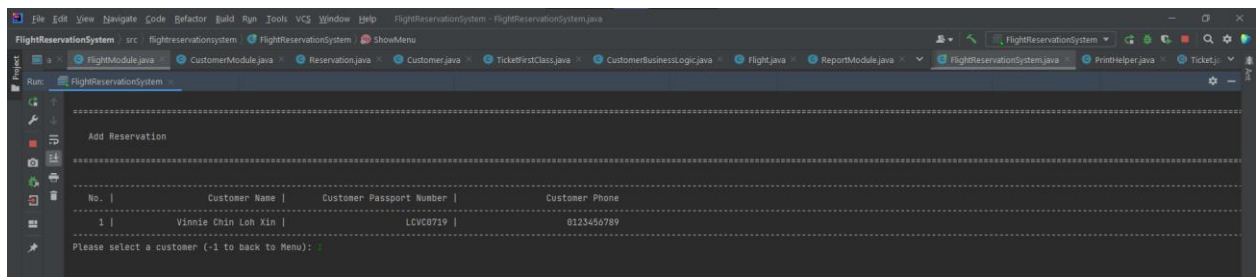
Figure 24: Add reservation
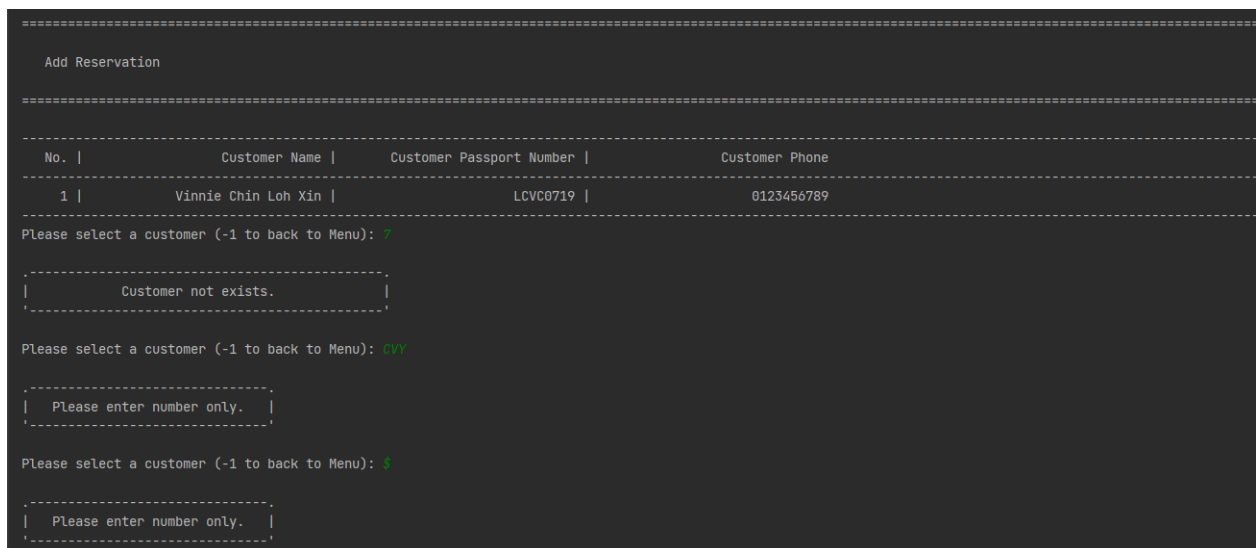
Figure 25: Validation for select customer

After the user enters "1" to add the reservation, the customer list will be displayed to let the user select to add the reservation. If a user enters a customer that does not exist, a validation message "Customer not exists" will be displayed. Other than this, if the user enters an alphabet or symbol, a validation message "Please enter number only" will be displayed.
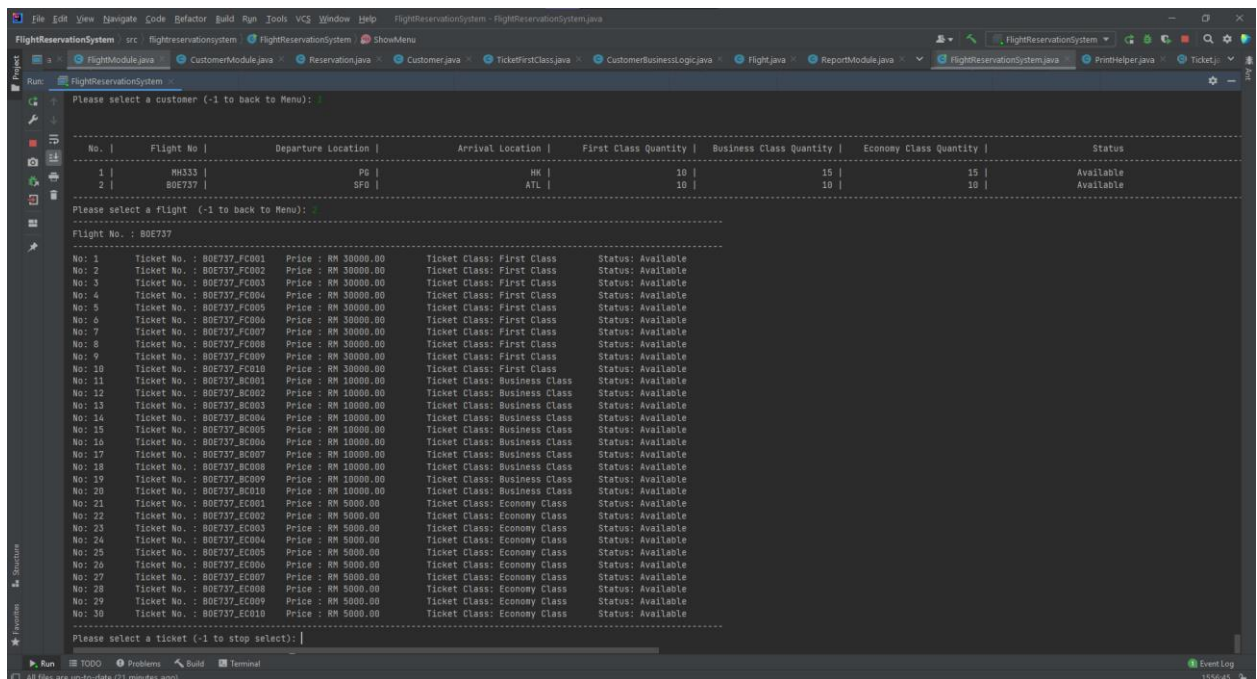
Figure 26: Reservation module (choose the flight & show ticket)

After selecting the customer, the user needs to select the flight. For this select flight, it will also have the same validation with the selected customer. If a user enters a flight that does not exist, a validation message "Flight not exists" will be displayed. Other than this, if a user enters an alphabet or symbol, a validation message "Please enter number only" will be displayed. When the user selects a flight and enters the correct input, all tickets for that flight will be shown.
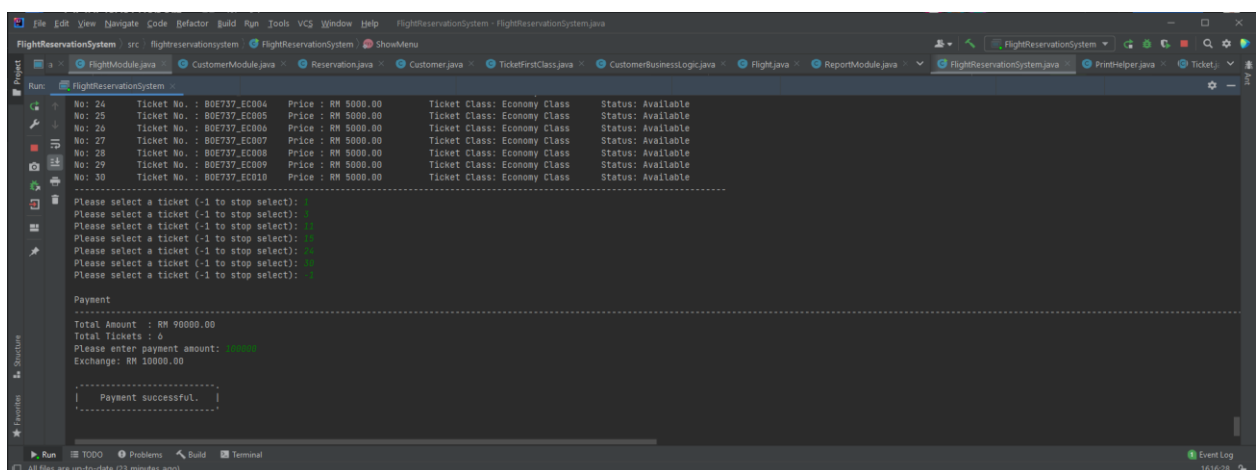
Figure 27 : Reservation module (choose ticket & make payment)

Furthermore, the user will need to select the tickets they want. After finishing selecting the tickets, enter "-1" to stop selecting and proceed to payment. Payment will show the amount that the user needs to pay, the user will need to enter the payment amount. If the amount paid by the user exceeds the amount that is provided, the exchange will be shown to the user. A message "Payment successful" will be shown to the user.

```
No: 31     Ticket No. : MH333_EC006    Price : RM 400.00        Ticket Class: Economy Class    Status: Available
No: 32     Ticket No. : MH333_EC007    Price : RM 400.00        Ticket Class: Economy Class    Status: Available
No: 33     Ticket No. : MH333_EC008    Price : RM 400.00        Ticket Class: Economy Class    Status: Available
No: 34     Ticket No. : MH333_EC009    Price : RM 400.00        Ticket Class: Economy Class    Status: Available
No: 35     Ticket No. : MH333_EC010    Price : RM 400.00        Ticket Class: Economy Class    Status: Available
No: 36     Ticket No. : MH333_EC011    Price : RM 400.00        Ticket Class: Economy Class    Status: Available
No: 37     Ticket No. : MH333_EC012    Price : RM 400.00        Ticket Class: Economy Class    Status: Available
No: 38     Ticket No. : MH333_EC013    Price : RM 400.00        Ticket Class: Economy Class    Status: Available
No: 39     Ticket No. : MH333_EC014    Price : RM 400.00        Ticket Class: Economy Class    Status: Available
No: 40     Ticket No. : MH333_EC015    Price : RM 400.00        Ticket Class: Economy Class    Status: Available
------------------------------------------------------------------------------------------------------------------
Please select a ticket (-1 to stop select): 40
Please select a ticket (-1 to stop select): 39
Please select a ticket (-1 to stop select): 39

 .-----------------------------------------------------.
 |   The ticket is chosen. Please select another ticket  |
 '-----------------------------------------------------'

Please select a ticket (-1 to stop select): 29
Please select a ticket (-1 to stop select): -1

Payment
------------------------------------------------------------------------------------------------------------------
Total Amount  : RM 1200.00
Total Tickets : 3
Please enter payment amount: 1000

 .-------------------------------.
 |    Please pay enough amount.    |
 '-------------------------------'

Please enter payment amount: 1200
Exchange: RM 0.00

 .------------------------.
 |    Payment successful.    |
 '------------------------'
```

Figure 28: Validation for selecting ticket and payment

When the user repeatedly selects the tickets that have been chosen, a validation message "The ticket is chosen. Please select another ticket"will be displayed. Then, if the amount paid by the user is less than the amount that is provided, a validation message "Please pay enough amount" will be displayed.

```
No: 32      Ticket No. : MH333_EC007    Price : RM 400.00       Ticket Class: Economy Class    Status: Available
No: 33      Ticket No. : MH333_EC008    Price : RM 400.00       Ticket Class: Economy Class    Status: Available
No: 34      Ticket No. : MH333_EC009    Price : RM 400.00       Ticket Class: Economy Class    Status: Available
No: 35      Ticket No. : MH333_EC010    Price : RM 400.00       Ticket Class: Economy Class    Status: Available
No: 36      Ticket No. : MH333_EC011    Price : RM 400.00       Ticket Class: Economy Class    Status: Available
No: 37      Ticket No. : MH333_EC012    Price : RM 400.00       Ticket Class: Economy Class    Status: Available
No: 38      Ticket No. : MH333_EC013    Price : RM 400.00       Ticket Class: Economy Class    Status: Available
No: 39      Ticket No. : MH333_EC014    Price : RM 400.00       Ticket Class: Economy Class    Status: Sold
No: 40      Ticket No. : MH333_EC015    Price : RM 400.00       Ticket Class: Economy Class    Status: Sold
---------------------------------------------------------------------------------------------------------------
Please select a ticket (-1 to stop select): 40


.----------------------------------------------------.
|   The ticket is sold. Please select another ticket.   |
'----------------------------------------------------'


Please select a ticket (-1 to stop select): 38
Please select a ticket (-1 to stop select): -1

Payment
---------------------------------------------------------------------------------------------------------------
Total Amount  : RM 400.00
Total Tickets : 1
Please enter payment amount: 500
Exchange: RM 100.00


.------------------------.
|    Payment successful.  |
'------------------------'
```

Figure 29: Validation for selecting ticket

When the user selects the tickets that have been sold, a validation message "The ticket is sold. Please select another ticket" will be displayed.
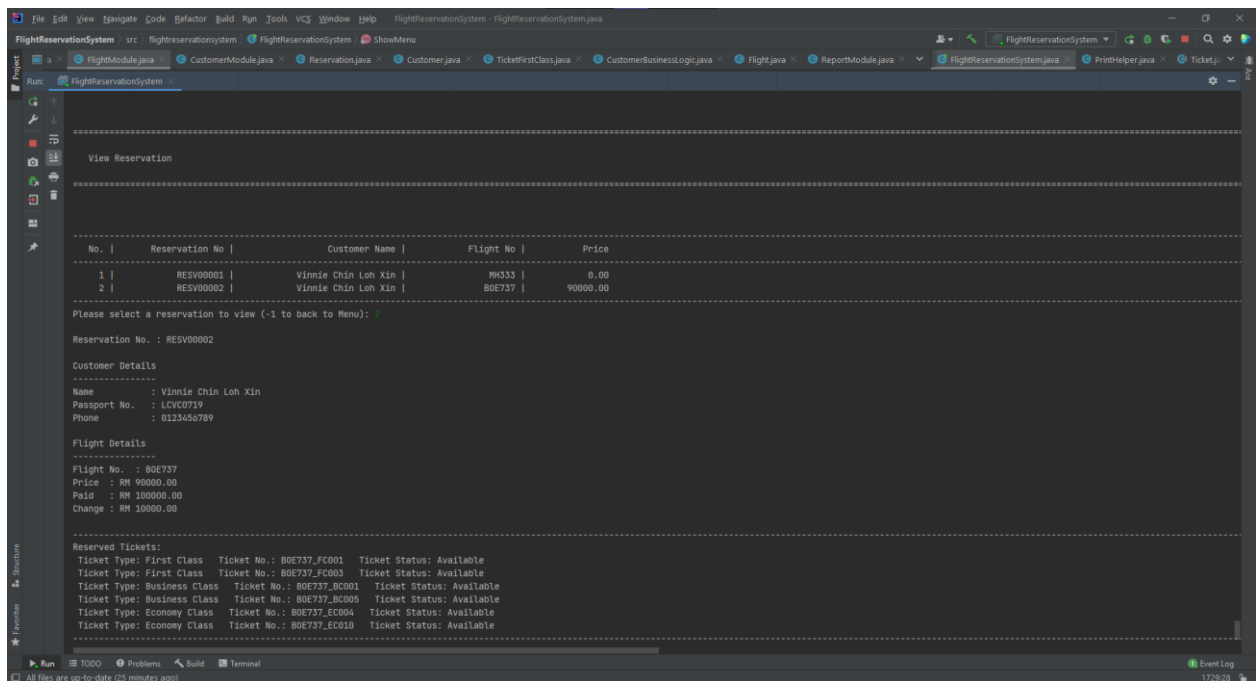
Figure 30: View reservation

After the user enters "2" to view the reservation, the customer reservation list will be displayed to let the user select to view the reservation. When the user chooses, the customer details, flight details and the ticket the customer reserve will be displayed.
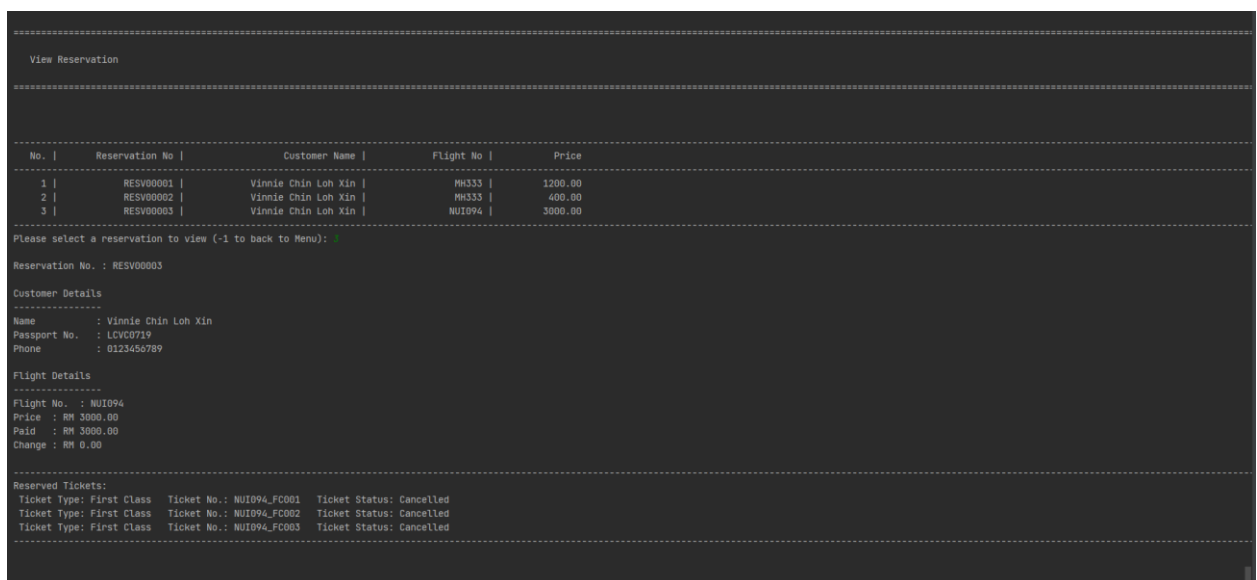
Figure 31: Ticket status

When the flight is cancelled, the ticket status also will change from available to cancelled.
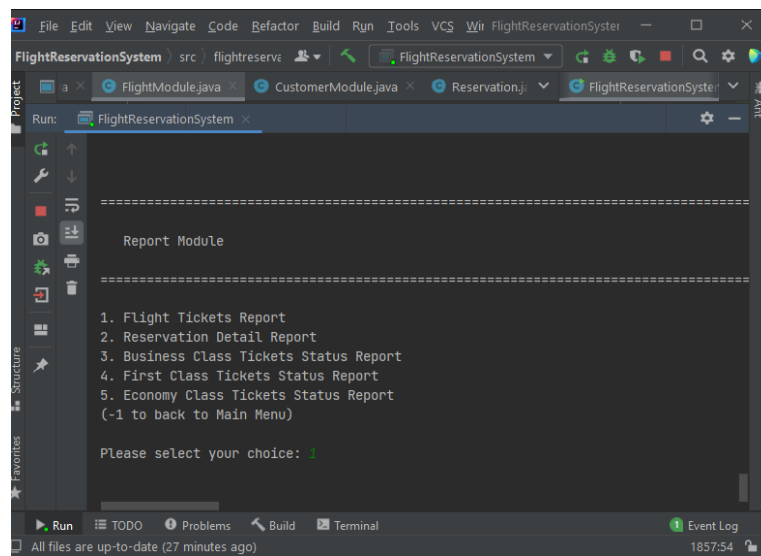
Figure 32: Report module (choose flight tickets report)

Figure 33: Flight tickets report(choose flight)

Figure 34: Flight tickets report

When the user needs to see the flight ticket report, then they just need to enter '1'. The flight ticket report will display after the user enters. Then, it will display all the flights in the report to let the user choose what flight ticket they need to see. If the user enters 2 to view the selected flight, all the tickets, prices, status, etc. will be displayed.
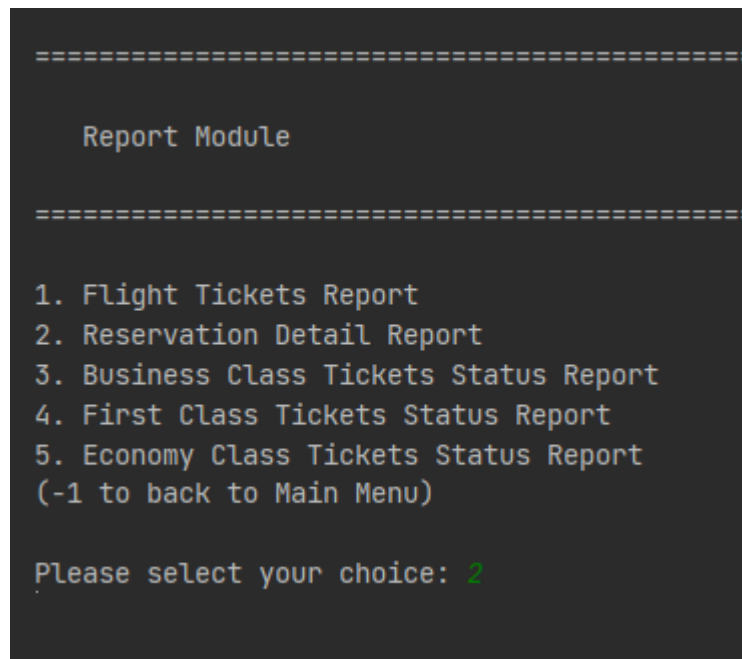
Figure 35:Report module (choose Reservation Detail Report)

Figure 36: Reservation Detail Report

The user needs to see the second report which is a reservation detail report, then they just need to enter '2'. The report will also display after the user enters. Then, it will display all the reservation details in the report.
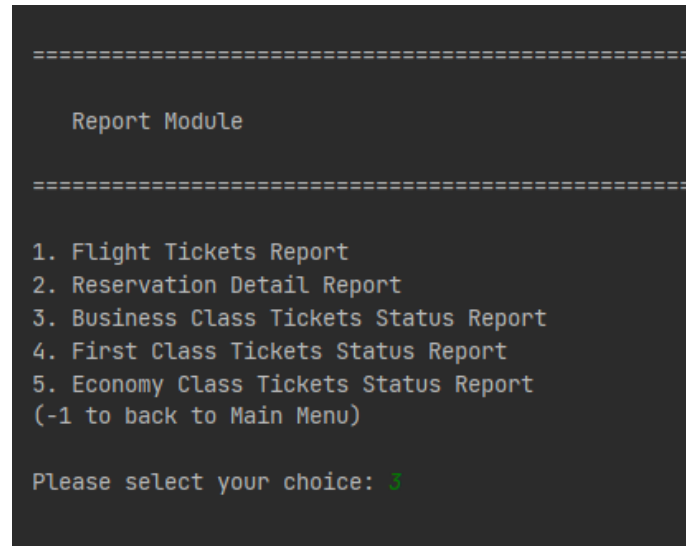
Figure 37:Report module (choose Business Class Tickets Status Report)

Figure 38: Business Class Tickets Status Report

The user needs to see the third report which is a businessclass tickets status report, then they just need to enter '3'. The report will also display after the user enters. Then, it will display all the business class ticket status like fight no, price, status, etc in the report.
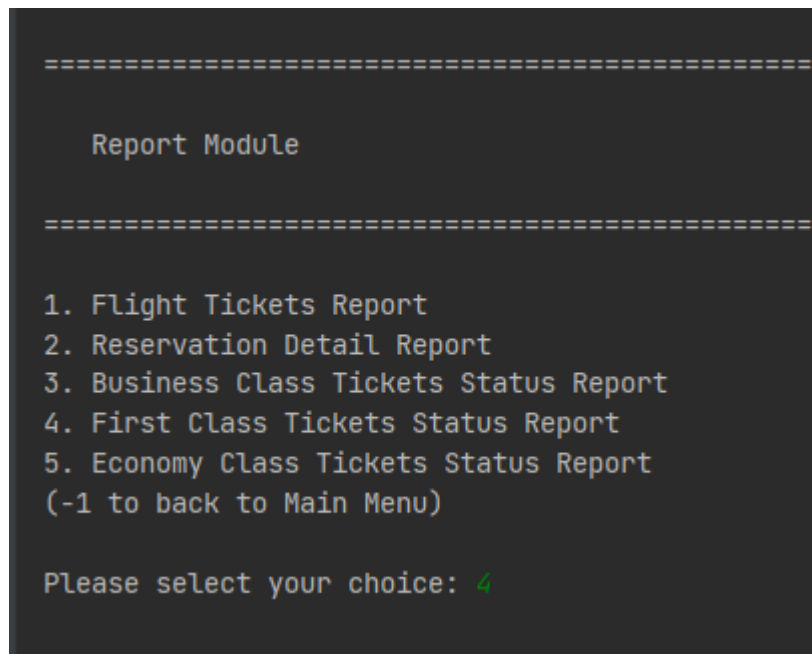
Figure 39: Report module (choose First Class Tickets Status Report)

Figure 40: First Class Tickets Status Report

The user needs to see the fourth report which is a firstclass tickets status report, then they just need to enter '4'. The report will also display after the user enters. Then, it will display all the first class ticket status like fight no, price, status, etc in the report.
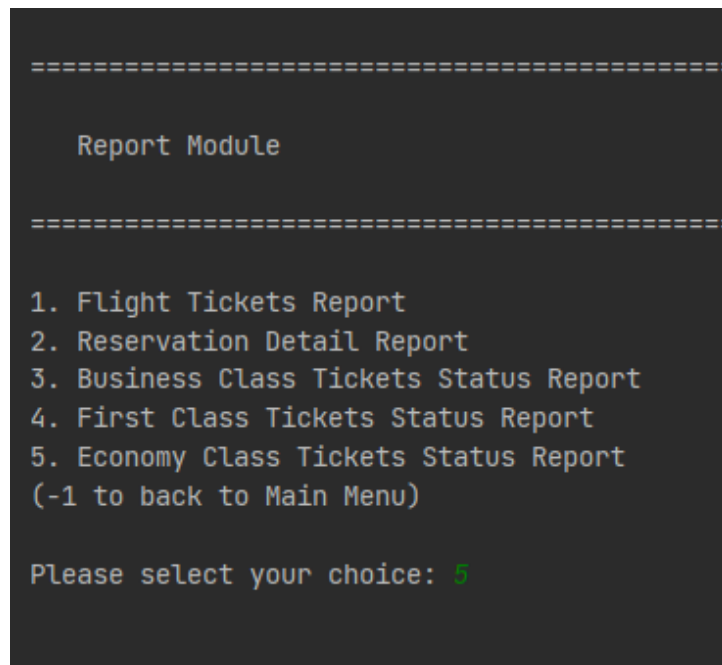
Figure 41: Report module (choose Economy Class Tickets Status Report)

Figure 42: Economy Class Tickets Status Report

The user needs to see the last report, which is an economy class tickets status report, then they just need to enter '5'. The report will also display after the user enters. Then, it will display all the economy class ticket status like fight no, price, status, etc in the report.

Figure 43: Report module validation for alphabet

Figure 44: Report module validation for number

Figure 45: Report module validation for symbol

When the user enters the report module, the report menu will be displayed. In the report module, it will be validated, in order to prevent the user from entering the invalid number, because there are only digits from 1 to 5. Therefore, when the user enters an invalid number, it will pop out the invalid message to let the user know they are entering the invalid number. In

this module, the user cannot press the alphabet, symbol or other digits to continue the process to another page.


Figure 46: Validation in each of the report

In all the reports, it also has validation, and their validation the same. But in the screenshot, we only show the validation of a report. In this module, the user cannot enter the alphabet, symbol or other digits to continue. When the user enters the invalid number, then it will display an invalid message to let the user know they are entering the invalid number.