



UTM
UNIVERSITI TEKNOLOGI MALAYSIA

OBJECT ORIENTED PROGRAMMING (SECJ2154)

SECTION 1

SEMESTER 1 2023/2024

GROUP PROJECT AIRLINE RESERVATION SYSTEM

NAME	MATRIC ID
MUHAMMAD IZMEER BIN ZULKIFLEE	B23CS0058
MUHAMMAD FIRDAUS BIN MOHIDEEN ABDUL CADER	B23CS0056
MUHAMMAD DENNISH BIN MOHD AZMI	B23CS0053
AHMAD AFIF AISY BIN AHMAD RIZAL	B23CS0018

LECTURER:
DR. LIZAWATI MI YUSUF

TABLE OF CONTENT

1.0 PROJECT DESCRIPTION.....	3
1.1 Introduction.....	3
1.2 Objectives and Scopes of the System.....	4
1.3 Purposes or Significance of the System.....	5
2.0 Object Oriented Concepts.....	6
2.1 Inheritance.....	6
2.2 Polymorphism.....	7
2.3 Encapsulation and Data Hiding.....	8
2.4 Association: Composition.....	11
2.5 Association: Aggregation.....	12
3.0 FLOW CHART.....	13
3.1 Flow Chart for AirAsia and MAS where origin is from Senai.....	13
3.2 Flow Chart for AirAsia where origin is from KLIA.....	14
3.3 Flow Chart for AirAsia where origin is from TAWAU.....	15
3.4 Flow Chart for MAS where origin is from KLIA.....	16
3.5 Flow Chart for MAS where origin is from TAWAU.....	17
4.0 UML CLASS DIAGRAM.....	18
4.1 Class SkyInfo.....	18
4.2 Class Booking.....	19
4.3 Class Reservation.....	19
4.4 Class Ticket.....	20
4.5 Class Passenger.....	20
4.6 Class Airline.....	21
4.7 Class Flight.....	21
4.8 Class Seat.....	22
4.9 Class SkyReserve.....	22
5.0 SOURCE CODE.....	23
5.1 SkyInfo.java.....	23
5.1 Booking.java.....	23
5.1 Reservation.java.....	24
5.1 Ticket.java.....	25
5.1 Passenger.java.....	25
5.1 Airline.java.....	26
5.1 Flight.java.....	26
5.1 Seat.java.....	27
5.1 SkyReserve.java.....	27
6.0 USER MANUAL.....	33
7.0 TASK DISTRIBUTION.....	39
8.0 CONCLUSION.....	39

1.0 PROJECT DESCRIPTION

1.1 Introduction

This easy to use Airline Reservation System (ARS) where it helps people to book flight reservations. It's a platform filled with splendid features. It's changed the way we book flights. It's easy to use, right on the web or an app. Travelers can hunt for flights fitting their needs like origin, destination or service. Updated flight details are at their fingertips. This includes schedules, open spots, and convenience. It helps others to plan better.

Choosing flights, seats, and additional services is effortless through ARS's straightforward reservation process. It instantly reserves the seat and flight based on your selections and provides a secure booking process to ensure your personal data stays private and secure. After successfully making a reservation, passengers gain complete control over their booking. You can review your chosen airlines, ticket detail, adjust your seat selections, and even book many flights.

Moreover, the ARS spotlights customer care with helplines and online chat tools. These resources assist passengers with questions, adaptations, and challenges, enhancing the whole customer journey. Basically, the Airline Reservation System is a comprehensive tool that covers the full travel process, showing the air travel sector's dedication to tech advancement, effectiveness, and client contentment.

1.2 Objectives and Scopes of the System

1.2.1 Quick Reservations: The ARS aims to offer travelers a streamlined booking journey. It provides a simple user interface, list of flight details, and a versatile option of seat to make this happen.

1.2.2 Great Resource Management: The system assists airlines in handling their seat allotment efficiently. The ARS helps boost profits and smartly uses resources for airlines. It changes seat accessibility and adjusts prices as per need.

1.2.3 Better Experience for Customers: The main goal here is to make customer experiences better. With the ARS, travelers can reserve flights that match what they need. They can handle their flight plans, reach out to support with ease. This all leads to gain customers happiness and satisfaction.

1.2.4 Booking and Taking Care of Flights: For the booking of flight, ARS has it covered from start to end. It helps people find flights. It helps them book. You can also manage your flights there - like picking your seat, tweaking the travel plan, even multi-passenger booking.

1.2.5 Live Info: The system dishes out instant airline updates on schedules, flight availability, and seat availability. With the latest trustworthy info, passengers can make smart choices.

.

1.3 Purposes or Significance of the System

1.3.1 Positive Passenger Establishment: The ARS keeps meaningful passenger details. It helps in bonding with customers. Data-driven services, better airline-passenger relations are possible.

1.3.2 Strategic Planning and Adjustment: Along with airline and flight seats, ARS assists in strategic planning and adjustments. It works for effective apportioning, flights up-keep, ground amenities are optimized.

1.3.3 Safety and Backup Measures: The ARS system comes with dependable disaster recovery and business endurance features. This backup process ensures valuable data is preserved. Recovery plans are ready for unexpected system breakdowns or interruptions.

1.3.4 Market Research and Strategy: Airlines use the ARS for reviewing the market and planning routes. This system provides an understanding of route popularity. It aids travelers in choosing the best destination.

1.3.5 Fast Reserves and Booking: The ARS enhances the booking process with quick and convenient reservation and booking. It strengthens the satisfaction of travelers to use ARS again later. This results in shortening the time needed for booking flights.

2.0 Object Oriented Concepts

2.1 Inheritance

The SkyInfo class is abstract, and Airline, Ticket, Flight, and Seat inherit from it.

```
1  ↴  abstract class SkyInfo {  
2      private String info;  
3  
4      public SkyInfo(String info) {  
5          this.info = info;  
6      }  
7  
8      public String getName() {  
9          return info;  
10     }  
11  
12     public void setName(String info) {  
13         this.info = info;  
14     }  
15 }
```

Figure 2.1.1 Inheritance Concept in SkyInfo Class

```
1  ↴  class Airline extends SkyInfo {  
2      public Airline(String info) {  
3          super(info);  
4      }  
5  }
```

Figure 2.1.2 Inheritance Concept in Airline Class

```
1  ↴  public class Flight extends SkyInfo{  
2      private String origin;  
3      private String destination;  
4      private String boardingTime;  
5  }
```

Figure 2.1.3 Inheritance Concept in Flight Class

```
1  ↴  public class Ticket extends SkyInfo {
2      private int ticketNumber;
3
4      public Ticket(int ticketNumber) {
5          super("Ticket");
6          this.ticketNumber = ticketNumber;
7      }
8
9      public int getTicketNumber() {
10         return ticketNumber;
11     }
12
13     public void setTicketNumber(int ticketNumber) {
14         this.ticketNumber = ticketNumber;
15     }
16 }
```

Figure 2.1.4 Inheritance Concept in Ticket Class

```
1  ↴  public class Seat extends SkyInfo {
2      private String seatName;
3
4      public Seat(String seatName) {
5          super("Seat");
6          this.seatName = seatName;
7      }
8 }
```

Figure 2.1.5 Inheritance Concept in Seat Class

2.2 Polymorphism

Polymorphism is used in the List<SkyInfo> in the SkyReserve class, which can hold objects of any subclass of SkyInfo.

```
1      import java.util.*;
2
3  ↴  public class SkyReserve {
4  ↴      public static void main(String[] args) {
5          Scanner scanner = new Scanner(System.in);
6
7          List<Airline> airlines = new ArrayList<>();
8          airlines.add(new Airline("AirAsia"));
9          airlines.add(new Airline("MAS"));
```

Figure 2.2.1 Polymorphism Concept in SkyReserve Class

2.3 Encapsulation and Data Hiding

- Achieved through private fields and public getter/setter methods in classes like SkyInfo, Airline, Ticket, Reservation, Passenger, Flight, and Seat.
- Data hiding is maintained by making fields private and controlling access through getter and setter methods.

```
4     public SkyInfo(String info) {
5         this.info = info;
6     }
7
8     public String getName() {
9         return info;
10    }
11
12    public void setName(String info) {
13        this.info = info;
14    }
15 }
```

Figure 2.3.1 Encapsulation Concept in SkyInfo Class

```
1  class Airline extends SkyInfo {
2      public Airline(String info) {
3          super(info);
4      }
5  }
```

Figure 2.3.2 Encapsulation Concept in Airline Class

```
4     public Ticket(int ticketNumber) {
5         super("Ticket");
6         this.ticketNumber = ticketNumber;
7     }
8
9     public int getTicketNumber() {
10    return ticketNumber;
11 }
12
13    public void setTicketNumber(int ticketNumber) {
14        this.ticketNumber = ticketNumber;
15    }
16 }
```

Figure 2.3.3 Encapsulation Concept in Ticket Class

```
6     public Reservation() {
7         this.tickets = new ArrayList<>();
8     }
9
10    public List<Ticket> getTickets() {
11        return tickets;
12    }
13
14    public void setTickets(List<Ticket> tickets) {
15        this.tickets = tickets;
16    }
17
18    public void addTicket(Ticket ticket) {
19        this.tickets.add(ticket);
20    }
21 }
```

Figure 2.3.4 Encapsulation Concept in Reservation Class

```
5     public Passenger(String name, String contactInfo) {
6         this.name = name;
7         this.contactInfo = contactInfo;
8     }
9
10    public String getName() {
11        return name;
12    }
13
14    public void setName(String name) {
15        this.name = name;
16    }
17
18    public String getContactInfo() {
19        return contactInfo;
20    }
21
22    public void setContactInfo(String contactInfo) {
23        this.contactInfo = contactInfo;
24    }
25}
```

Figure 2.3.5 Encapsulation Concept in Passenger Class

```
4     public Seat(String seatName) {
5         super("Seat");
6         this.seatName = seatName;
7     }
8
9     public String getSeatName() {
10        return seatName;
11    }
```

Figure 2.3.6 Encapsulation Concept in Seat Class

```
6     public Flight(String origin, String destination, String boardingTime) {
7         super("Flight");
8         this.origin = origin;
9         this.destination = destination;
10        this.boardingTime = boardingTime;
11    }
12
13    public String getOrigin() {
14        return origin;
15    }
16
17    public void setOrigin(String origin) {
18        this.origin = origin;
19    }
20
21    public String getDestination() {
22        return destination;
23    }
24
25    public void setDestination(String destination) {
26        this.destination = destination;
27    }
28
29    public String getBoardingTime() {
30        return boardingTime;
31    }
```

Figure 2.3.7 Encapsulation Concept in Flight Class

2.4 Association: Composition

- The Booking class has composition relationships with Passenger, Flight, Seat, and Airline classes, as it contains instances of these classes as its fields.
- The Reservation class has a composition relationship with the Ticket class, as it contains a list of Ticket instances.

```
7     public Booking(Passenger passenger, Flight flight, Seat seat, Airline airline) {
8         this.passenger = passenger;
9         this.flight = flight;
10        this.seat = seat;
11        this.airline = airline;
12    }
```

Figure 2.4.1 Composition Concept in Booking Class

```
3  public class Reservation {  
4      private List<Ticket> tickets;  
5  
6      public Reservation() {  
7          this.tickets = new ArrayList<>();  
8      }  
9  
10     public List<Ticket> getTickets() {  
11         return tickets;  
12     }
```

Figure 2.4.2 Composition Concept in Reservation Class

2.5 Association: Aggregation

- The SkyReserve class has an aggregation relationship with a List<Booking> bookings, as it holds a collection of Booking instances.

```
39         List<Booking> bookings = new ArrayList<>();  
40  
41         boolean continueBooking = true;  
42         while (continueBooking) {  
43             try {  
44                 System.out.print("Enter passenger name: ");  
45                 String passengerName = scanner.nextLine();  
46  
47                 System.out.print("Enter contact information: ");  
48                 String contactInfo = scanner.nextLine();  
49  
50                 Passenger passenger = new Passenger(passengerName, contactInfo);  
51  
52                 System.out.println("\nChoose Airline:");  
53                 for (int i = 0; i < airlines.size(); i++) {  
54                     System.out.println((i + 1) + ". " + airlines.get(i).getName());  
55                 }
```

Figure 2.5.2 Composition Concept in Reservation Class

3.0 FLOW CHART

3.1 Flow Chart for AirAsia and MAS where origin is from Senai

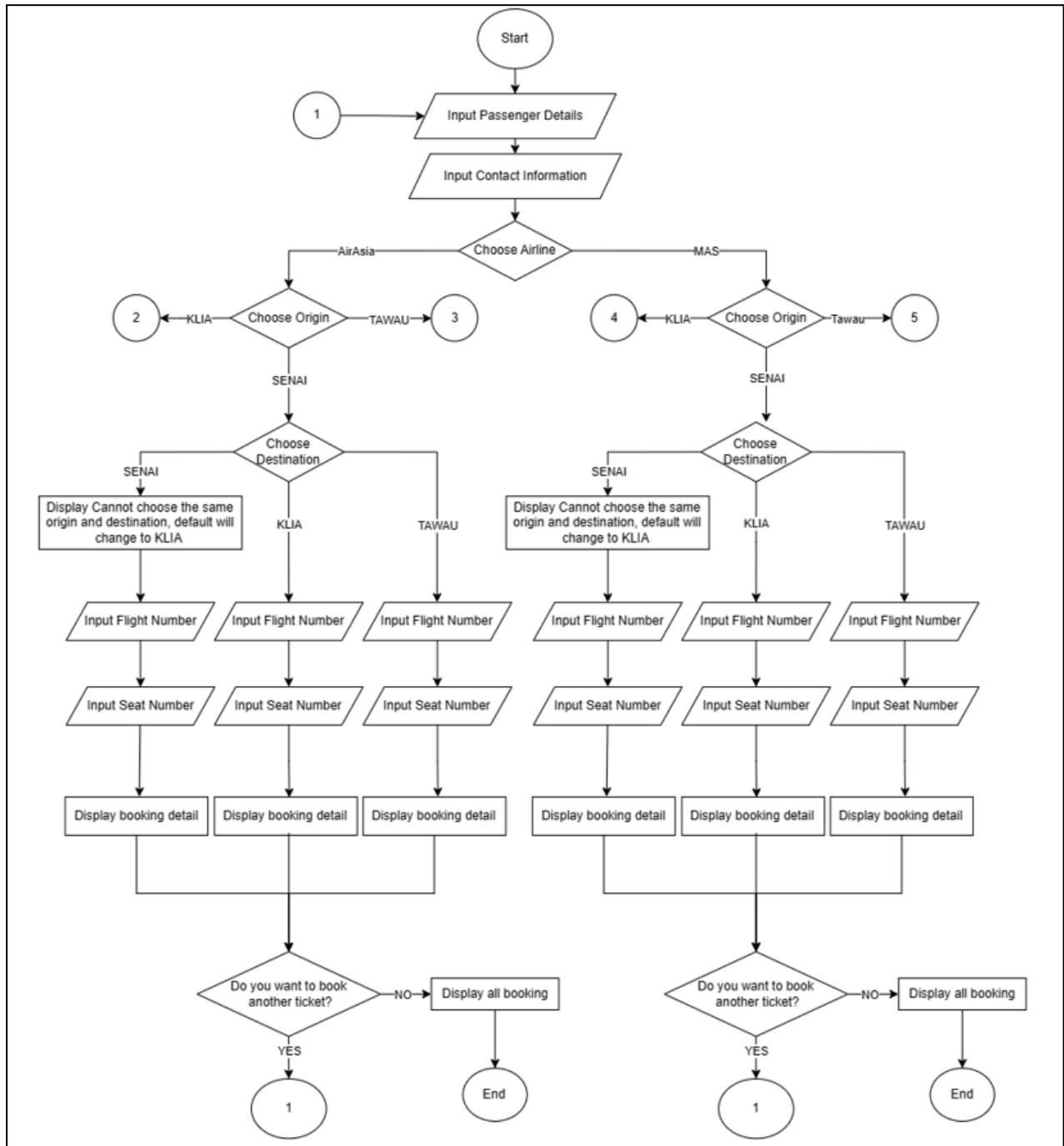


Figure 3.1 Flow Chart for AirAsia and MAS where origin is from Senai

3.2 Flow Chart for AirAsia where origin is from KLIA

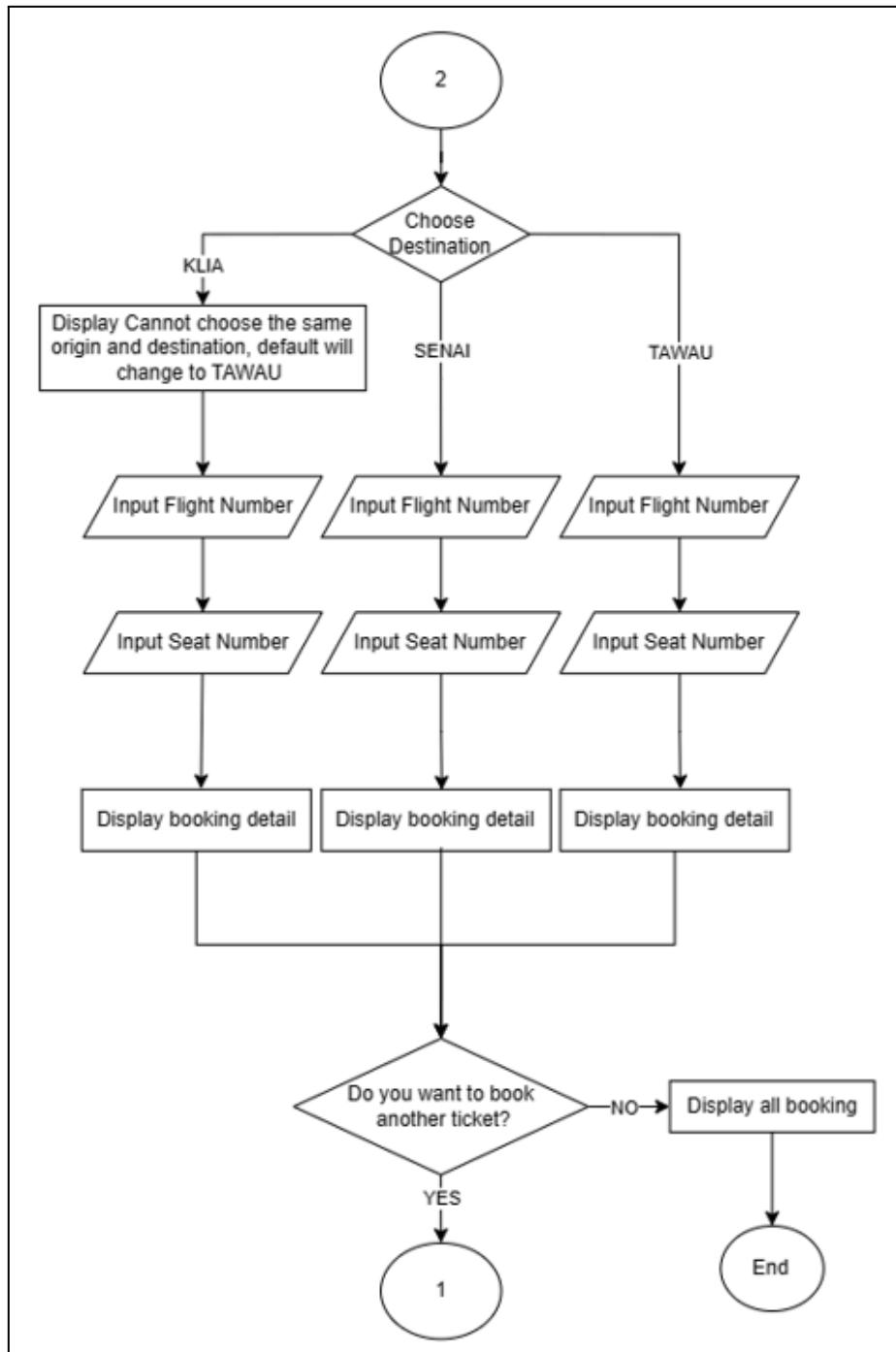


Figure 3.2 Flow Chart for AirAsia where origin is from KLIA

3.3 Flow Chart for AirAsia where origin is from TAWAU

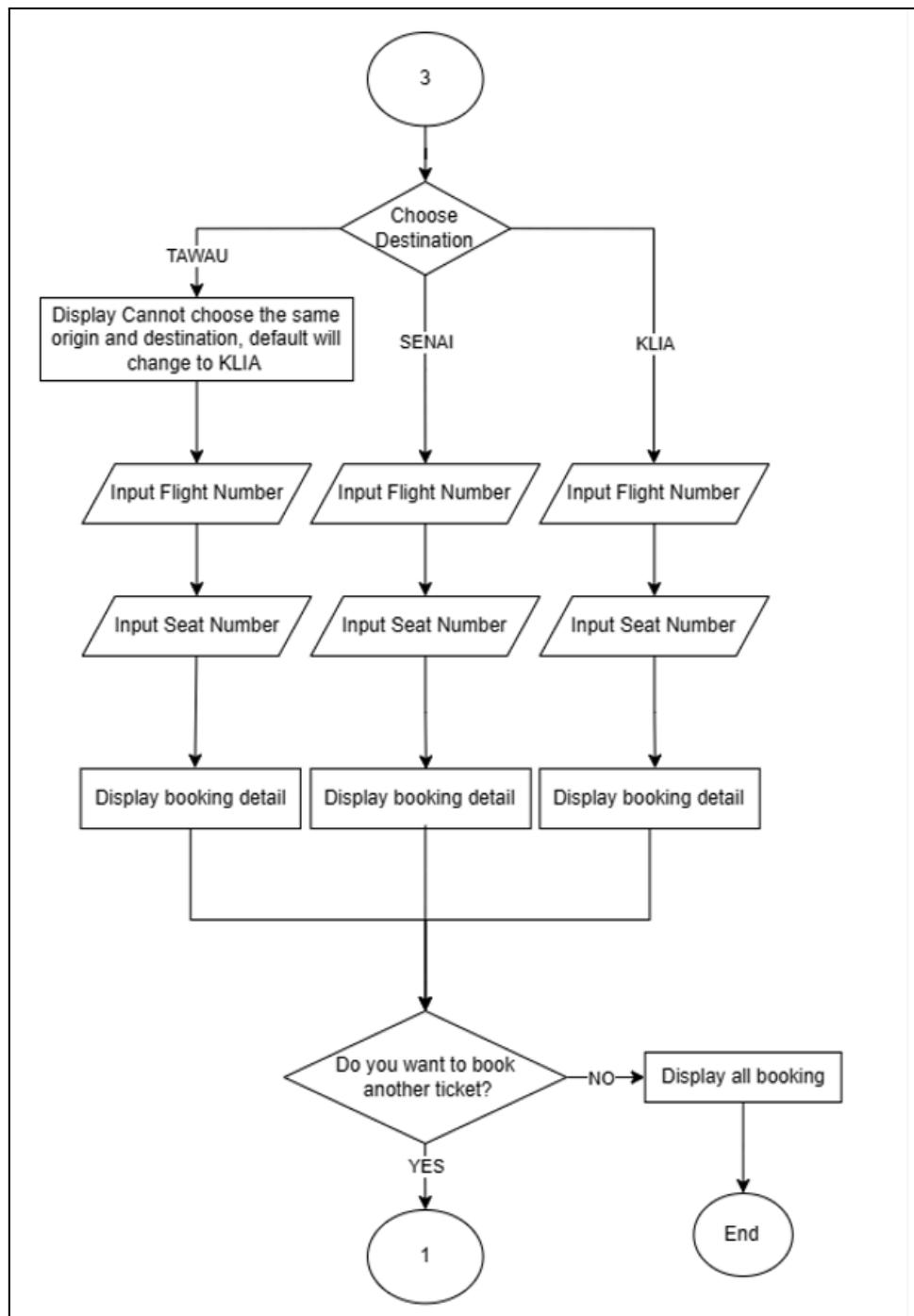


Figure 3.3 Flow Chart for AirAsia where origin is from TAWAU

3.4 Flow Chart for MAS where origin is from KLIA

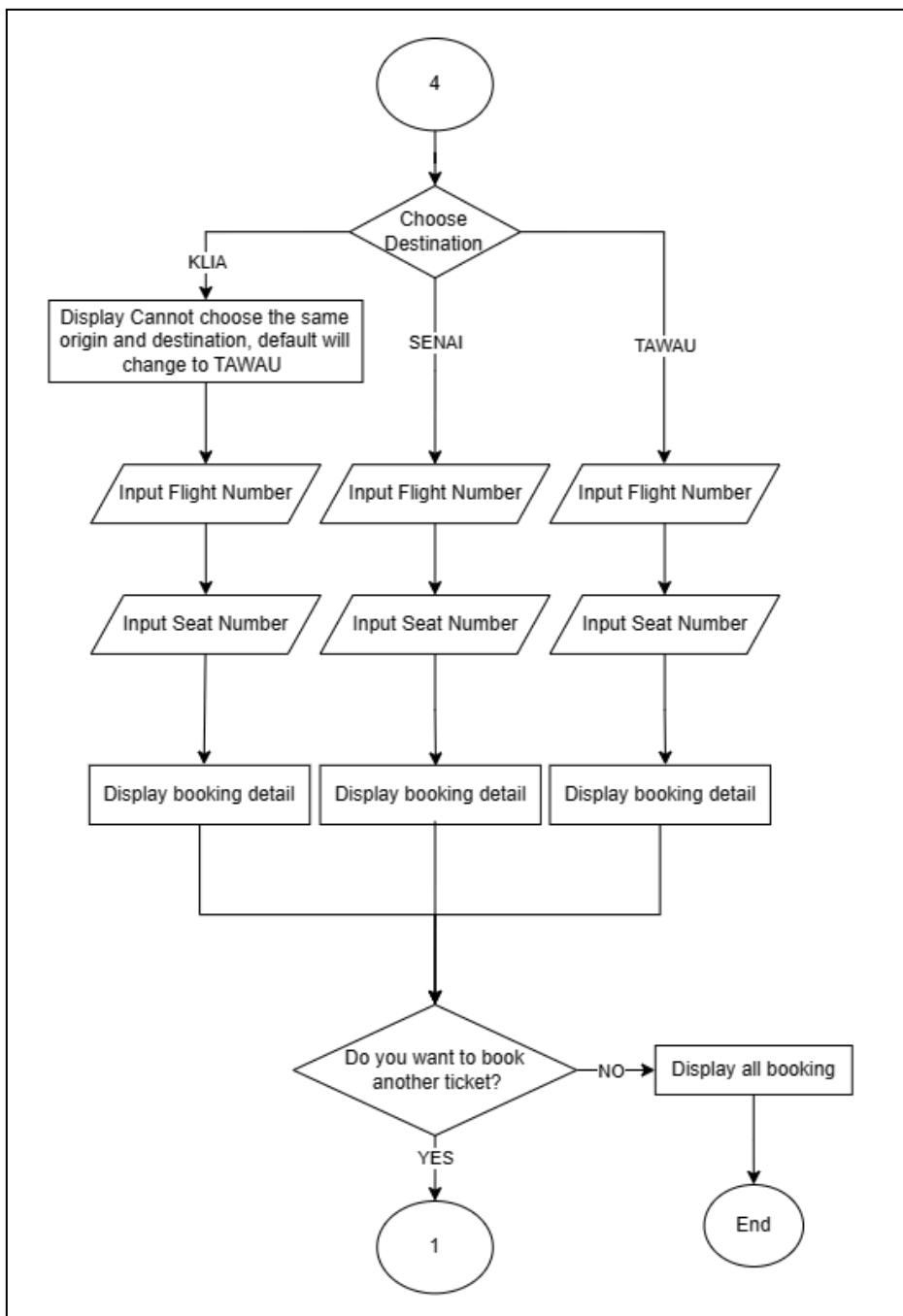


Figure 3.4 Flow Chart for MAS where origin is from KLIA

3.5 Flow Chart for MAS where origin is from TAWAU

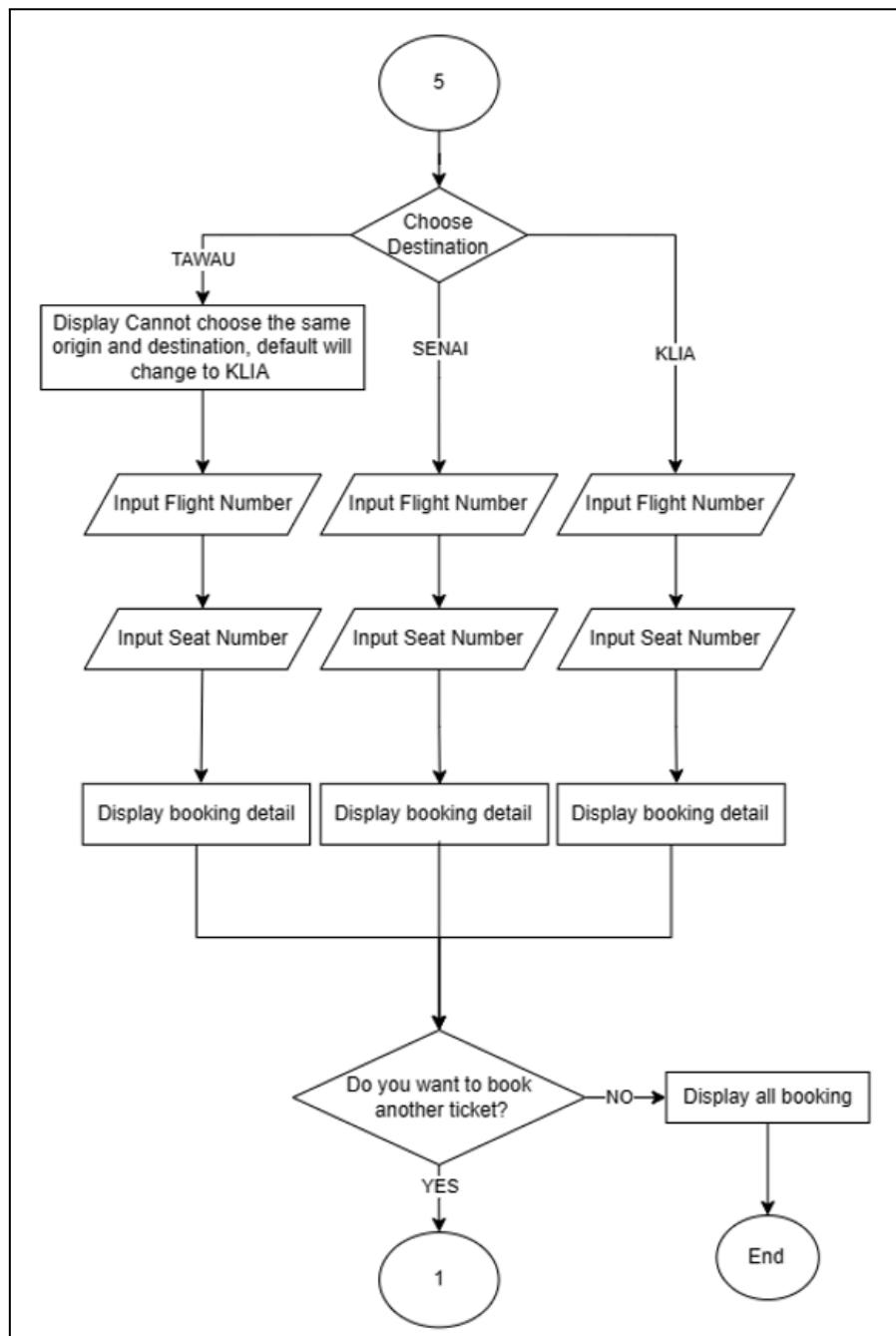


Figure 3.5 Flow Chart for MAS where origin is from TAWAU

4.0 UML CLASS DIAGRAM

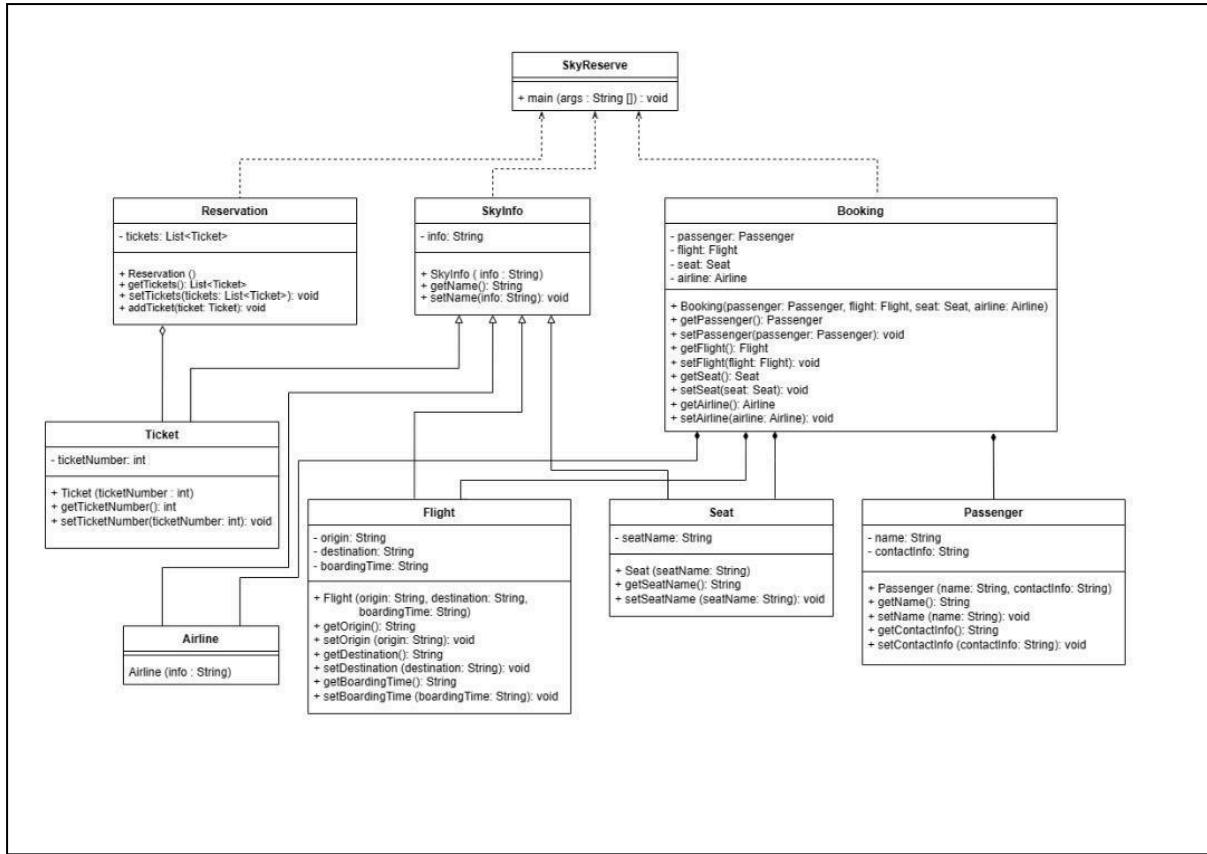


Figure 4.1 UML Class Diagram of the Airline Reservation System

4.1 Class SkyInfo

Attributes	Description
info	The information associated with instances of classes that extend SkyInfo
Methods	Description
SkyInfo	Initializes the info attribute with the provided information
getName	Return the value of info attribute
setName	Sets the value of the name attribute

4.2 Class Booking

Attributes	Description
passenger	The passenger associated with the booking
flight	The flight associated with the booking
seat	The seat associated with the booking
airline	The airline associated with the booking
Methods	Description
Booking	Various getters and setters for ‘passenger’, ‘seat’ and ‘airline’
getPassenger	Returns the value of the passenger attribute
getFlight	Returns the value of the flight attribute
getSeat	Returns the value of the seat attribute
getAirline	Returns the value of the airline attribute
setPassenger	Sets the value of passenger attribute
setFlight	Sets the value of flight attribute
setSeat	Sets the value of seat attribute
setAirline	Sets the value of airline attribute

4.3 Class Reservation

Attributes	Description
tickets	A list containing tickets
Methods	Description
getTickets	Returns the value of the ticket attribute
setTickets	Sets the value of ticket attribute
addTicket	Adds a ticket to the list

4.4 Class Ticket

Attributes	Description
ticketNumber	The ticket number of the ticket
Methods	Description
Ticket	Initializes the 'info' attribute with "Ticket" and sets the 'ticketNumber'.
getTicketNumber	Returns the value of the ticket number attribute
setTicketNumber	Sets the value of ticket number attribute

4.5 Class Passenger

Attributes	Description
name	The name of the passenger
contactInfo	The contact information
Methods	Description
Passenger	Initializes the 'name' and 'contactInfo' attributes with the provided information
getName	Returns the value of the name attribute
getContactInfo	Returns the value of contact info attribute
setName	Sets the value of name attribute
setContactInfo	Sets the value of contact info attribute

4.6 Class Airline

Attributes	Description
Methods	Description
Airline	Initializes the 'info' attribute with the provided information.

4.7 Class Flight

Attributes	Description
origin	The origin airport of the flight
destination	The destination airport of the flight
boardingTime	The boarding time of the flight
Methods	Description
Flight	Initializes the 'info' attribute with "Flight" and sets the 'origin', 'destination', and 'boardingTime'.
getOrigin	Returns the value of origin attribute
getDestination	Returns the value of destination attribute
getBoardingTime	Returns the value of boarding time attribute
setOrigin	Sets the value of origin attribute
setDestination	Sets the value of destination attribute
setBoardingTime	Sets the value of boarding time attribute

4.8 Class Seat

Attributes	Description
seatName	The name of the seat
Methods	Description
Seat	Initializes the 'info' attribute with "Seat" and sets the 'seatName'.
getSeatName	Returns the value of seat name attribute
setSeatName	Sets the value of seat name attribute

4.9 Class SkyReserve

Attributes	Description
Methods	Description
main	The main method where the program execution starts. Handles the user interaction for booking flights and displays the booked flights at the end.

5.0 SOURCE CODE

5.1 SkyInfo.java

```
abstract class SkyInfo {  
    private String info;  
  
    public SkyInfo(String info) {  
        this.info = info;  
    }  
  
    public String getName() {  
        return info;  
    }  
  
    public void setName(String info) {  
        this.info = info;  
    }  
}
```

5.1 Booking.java

```
public class Booking {  
    private Passenger passenger;  
    private Flight flight;  
    private Seat seat;  
    private Airline airline;  
  
    public Booking(Passenger passenger, Flight flight, Seat seat, Airline airline) {  
        this.passenger = passenger;  
        this.flight = flight;  
        this.seat = seat;  
        this.airline = airline;  
    }  
  
    public Passenger getPassenger() {  
        return passenger;  
    }  
  
    public void setPassenger(Passenger passenger) {  
        this.passenger = passenger;  
    }  
  
    public Flight getFlight() {  
        return flight;  
    }
```

```

    }

    public void setFlight(Flight flight) {
        this.flight = flight;
    }

    public Seat getSeat() {
        return seat;
    }

    public void setSeat(Seat seat) {
        this.seat = seat;
    }

    public Airline getAirline() {
        return airline;
    }

    public void setAirline(Airline airline) {
        this.airline = airline;
    }
}

```

5.1 Reservation.java

```

import java.util.*;

public class Reservation {
    private List<Ticket> tickets;

    public Reservation() {
        this.tickets = new ArrayList<>();
    }

    public List<Ticket> getTickets() {
        return tickets;
    }

    public void setTickets(List<Ticket> tickets) {
        this.tickets = tickets;
    }

    public void addTicket(Ticket ticket) {
        this.tickets.add(ticket);
    }
}

```

5.1 Ticket.java

```
public class Ticket extends SkyInfo {  
    private int ticketNumber;  
  
    public Ticket(int ticketNumber) {  
        super("Ticket");  
        this.ticketNumber = ticketNumber;  
    }  
  
    public int getTicketNumber() {  
        return ticketNumber;  
    }  
  
    public void setTicketNumber(int ticketNumber) {  
        this.ticketNumber = ticketNumber;  
    }  
}
```

5.1 Passenger.java

```
public class Passenger {  
    private String name;  
    private String contactInfo;  
  
    public Passenger(String name, String contactInfo) {  
        this.name = name;  
        this.contactInfo = contactInfo;  
    }  
  
    public String getName() {  
        return name;  
    }  
  
    public void setName(String name) {  
        this.name = name;  
    }  
  
    public String getContactInfo() {  
        return contactInfo;  
    }  
    public void setContactInfo(String contactInfo) {  
        this.contactInfo = contactInfo;  
    }  
}
```

5.1 Airline.java

```
class Airline extends SkyInfo {  
    public Airline(String info) {  
        super(info);  
    }  
}
```

5.1 Flight.java

```
public class Flight extends SkyInfo {  
    private String origin;  
    private String destination;  
    private String boardingTime;  
  
    public Flight(String origin, String destination, String boardingTime) {  
        super("Flight");  
        this.origin = origin;  
        this.destination = destination;  
        this.boardingTime = boardingTime;  
    }  
    public String getOrigin() {  
        return origin;  
    }  
    public void setOrigin(String origin) {  
        this.origin = origin;  
    }  
  
    public String getDestination() {  
        return destination;  
    }  
  
    public void setDestination(String destination) {  
        this.destination = destination;  
    }  
    public String getBoardingTime() {  
        return boardingTime;  
    }  
  
    public void setBoardingTime(String boardingTime) {  
        this.boardingTime = boardingTime;  
    }  
}
```

5.1 Seat.java

```
public class Seat extends SkyInfo {
    private String seatName;

    public Seat(String seatName) {
        super("Seat");
        this.seatName = seatName;
    }

    public String getSeatName() {
        return seatName;
    }

    public void setSeatName(String seatName) {
        this.seatName = seatName;
    }
}
```

5.1 SkyReserve.java

```
import java.util.*;

public class SkyReserve{
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        List<Airline> airlines = new ArrayList<>();
        airlines.add(new Airline("AirAsia"));
        airlines.add(new Airline("MAS"));

        List<Flight> availableFlightsAirAsia = new ArrayList<>();
        availableFlightsAirAsia.add(new Flight("KLIA", "SENAI", "10:00 AM"));
        availableFlightsAirAsia.add(new Flight("KLIA", "TAWAU", "12:00 PM"));
        availableFlightsAirAsia.add(new Flight("SENAI", "KLIA", "2:00 PM"));
        availableFlightsAirAsia.add(new Flight("SENAI", "TAWAU", "4:00 PM"));

        availableFlightsAirAsia.add(new Flight("TAWAU", "KLIA", "4:00 PM"));
        availableFlightsAirAsia.add(new Flight("TAWAU", "SENAI", "6:00 PM"));

        List<Seat> availableSeatsAirAsia = new ArrayList<>();
        for (char seatChar = 'A'; seatChar <= 'J'; seatChar++) {
            availableSeatsAirAsia.add(new Seat(String.valueOf(seatChar)));
        }
    }
}
```

```

List<Flight> availableFlightsMAS = new ArrayList<>();
availableFlightsMAS.add(new Flight("KLIA", "SENAI", "11:00 AM"));
availableFlightsMAS.add(new Flight("KLIA", "TAWAU", "1:00 PM"));
availableFlightsMAS.add(new Flight("SENAI", "KLIA", "3:00 PM"));
availableFlightsMAS.add(new Flight("SENAI", "TAWAU", "6:00 PM"));
availableFlightsMAS.add(new Flight("TAWAU", "SENAI", "5:00 PM"));
availableFlightsMAS.add(new Flight("TAWAU", "KLIA", "8:00 PM"));

List<Seat> availableSeatsMAS = new ArrayList<>();
for (char seatChar = 'Q'; seatChar <= 'Z'; seatChar++) {
    availableSeatsMAS.add(new Seat(String.valueOf(seatChar)));
}

List<Booking> bookings = new ArrayList<>();

boolean continueBooking = true;
while (continueBooking) {
    try {
        System.out.print("Enter passenger name: ");
        String passengerName = scanner.nextLine();

        System.out.print("Enter contact information: ");
        String contactInfo = scanner.nextLine();

        Passenger passenger = new Passenger(passengerName, contactInfo);

        System.out.println("\nChoose Airline:");
        for (int i = 0; i < airlines.size(); i++) {
            System.out.println((i + 1) + ". " + airlines.get(i).getName());
        }
        System.out.print("Enter the number of your chosen airline: ");
        int airlineChoice = scanner.nextInt();
        scanner.nextLine();

        if (airlineChoice < 1 || airlineChoice > airlines.size()) {
            System.out.println("Invalid choice. Defaulting to AirAsia.");
            airlineChoice = 1;
        }
    }

    Airline selectedAirline = airlines.get(airlineChoice - 1);

    List<Flight> availableFlights;
    List<Seat> availableSeats;
    if (selectedAirline.getName().equals("AirAsia")) {
        availableFlights = availableFlightsAirAsia;
        availableSeats = availableSeatsAirAsia;
    } else {
        availableFlights = availableFlightsMAS;
    }
}

```

```

        availableSeats = availableSeatsMAS;
    }

    int originChoice = 0;
    String origin = null;
    while (true) {
        System.out.println("\nOrigin:");
        System.out.println("1. KLIA");
        System.out.println("2. SENAI");
        System.out.println("3. TAWAU");
        System.out.print("Choose your origin: ");
        originChoice = scanner.nextInt();
        scanner.nextLine(); // consume the newline character

        //String origin;
        switch (originChoice) {
            case 1:
                origin = "KLIA";
                break;
            case 2:
                origin = "SENAI";
                break;
            case 3:
                origin = "TAWAU";
                break;
            default:
                System.out.println("Invalid choice. Please choose a valid origin.");
                continue;
        }
        break;
    }

    int destinationChoice = 0;
    String destination = null;
    while (true) {
        System.out.println("\nDestination:");
        System.out.println("1. KLIA");
        System.out.println("2. SENAI");
        System.out.println("3. TAWAU");
        System.out.print("Choose your destination: ");
        destinationChoice = scanner.nextInt();

        switch (destinationChoice) {
            case 1:
                if(originChoice == 1){
                    System.out.print("\nCannot choose the same origin and destination,
default will change to TAWAU");
                    destination = "TAWAU";
                }else{

```

```

        destination = "KLIA";
    }
    break;
case 2:
    if(originChoice == 2){
        System.out.print("\nCannot choose the same origin and destination,
default will change to KLIA");
        destination = "KLIA";
    }else{
        destination = "SENAI";
    }
    break;
case 3:
    if(originChoice == 3){
        System.out.print("\nCannot choose the same origin and destination,
default will change to KLIA");
        destination = "KLIA";
    }else{
        destination = "TAWAU";
    }
    break;
default:
    System.out.println("Invalid choice. Please choose a valid destination.");
    continue;
}
break;
}

System.out.println("\nAvailable flights from " + origin + " to " + destination + "
with " + selectedAirline.getName() + ":");

for (int i = 0; i < availableFlights.size(); i++) {
    Flight flight = availableFlights.get(i);
    if (flight.getOrigin().equals(origin) &&
flight.getDestination().equals(destination)) {
        System.out.println((i + 1) + ". Origin: " + flight.getOrigin() +
", Destination: " + flight.getDestination() +
", Boarding Time: " + flight.getBoardingTime());
    }
}

System.out.print("Choose a flight number: ");
int flightNumber = scanner.nextInt();
scanner.nextLine();

Flight selectedFlight;
if (selectedAirline.getName().equals("AirAsia")) {
    selectedFlight = availableFlightsAirAsia.get(flightNumber - 1);
} else {
    selectedFlight = availableFlightsMAS.get(flightNumber - 1);
}

```

```

    }

List<Seat> availableSeatsForFlight = new ArrayList<>(availableSeats);

    for (Booking booked : bookings) {
        if (booked.getFlight().equals(selectedFlight) &&
booked.getAirline().equals(selectedAirline)) {
            availableSeatsForFlight.remove(booked.getSeat());
        }
    }

System.out.println("\nAvailable seats:");
for (int i = 0; i < availableSeatsForFlight.size(); i++) {
    Seat seat = availableSeatsForFlight.get(i);
    System.out.println((i + 1) + ". Seat: " + seat.getSeatName());
}

System.out.print("Choose a seat number: ");
int seatNumber = scanner.nextInt();
scanner.nextLine();
Seat selectedSeat = availableSeatsForFlight.remove(seatNumber - 1);

Booking booking = new Booking(passenger, selectedFlight, selectedSeat,
selectedAirline);
bookings.add(booking);

System.out.println("\nBooking details:");
System.out.println("Passenger: " + booking.getPassenger().getName());
System.out.println("Contact Info: " + booking.getPassenger().getContactInfo());
System.out.println("Airline: " + booking.getAirline().getName());
System.out.println("Flight: Flight " + flightNumber + " - Origin: " +
booking.getFlight().getOrigin() +
", Destination: " + booking.getFlight().getDestination() +
", Boarding Time: " + booking.getFlight().getBoardingTime());
System.out.println("Seat: " + booking.getSeat().getSeatName());

System.out.print("\nDo you want to book another ticket? (yes/no): ");
String choice = scanner.nextLine();
continueBooking = choice.equalsIgnoreCase("yes");
} catch (Exception e) {
    System.out.println("An error occurred. Please try again.");
    scanner.nextLine();
}
}

System.out.println("\nAll Bookings:");
for (int i = 0; i < bookings.size(); i++) {
    Booking booking = bookings.get(i);
    System.out.println("Booking " + (i + 1));
}

```

```
System.out.println("Passenger: " + booking.getPassenger().getName());
System.out.println("Contact Info: " + booking.getPassenger().getContactInfo());
System.out.println("Airline: " + booking.getAirline().getName());
System.out.println("Flight: Flight " + (i + 1) + " - Origin: " +
booking.getFlight().getOrigin() +
", Destination: " + booking.getFlight().getDestination() +
", Boarding Time: " + booking.getFlight().getBoardingTime());
System.out.println("Seat: " + booking.getSeat().getSeatName());
System.out.println("-----");
}
scanner.close();
}
```

6.0 USER MANUAL

Choice	Output
1	<pre>Enter passenger name: Daus Enter contact information: 0198764567 Choose Airline: 1. AirAsia 2. MAS Enter the number of your chosen airline: 1 Origin: 1. KLIA 2. SENAI 3. TAWAU Choose your origin: 1 Destination: 1. KLIA 2. SENAI 3. TAWAU Choose your destination: 3 Available flights from KLIA to TAWAU with AirAsia: 2. Origin: KLIA, Destination: TAWAU, Boarding Time: 12:00 PM Choose a flight number: 2 Available seats: 1. Seat: A 2. Seat: B 3. Seat: C 4. Seat: D 5. Seat: E 6. Seat: F 7. Seat: G 8. Seat: H 9. Seat: I 10. Seat: J Choose a seat number: 5 Booking details: Passenger: Daus Contact Info: 0198764567 Airline: AirAsia Flight: Flight 2 - Origin: KLIA, Destination: TAWAU, Boarding Time: 12:00 PM Seat: E Do you want to book another ticket? (yes/no): no All Bookings: Booking 1 Passenger: Daus Contact Info: 0198764567 Airline: AirAsia Flight: Flight 1 - Origin: KLIA, Destination: TAWAU, Boarding Time: 12:00 PM Seat: E -----</pre>

2

Enter passenger name: Dennish
Enter contact information: 0134258907

Choose Airline:

1. AirAsia
2. MAS

Enter the number of your chosen airline: 1

Origin:

1. KLIA
2. SENAI
3. TAWAU

Choose your origin: 1

Destination:

1. KLIA
2. SENAI
3. TAWAU

Choose your destination: 2

Available flights from KLIA to SENAI with AirAsia:

1. Origin: KLIA, Destination: SENAI, Boarding Time: 10:00 AM

Choose a flight number: 1

Available seats:

1. Seat: A
2. Seat: B
3. Seat: C
4. Seat: D
5. Seat: E
6. Seat: F
7. Seat: G
8. Seat: H
9. Seat: I
10. Seat: J

Choose a seat number: 3

Booking details:

Passenger: Dennish

Contact Info: 0134258907

Airline: AirAsia

Flight: Flight 1 - Origin: KLIA, Destination: SENAI, Boarding Time: 10:00 AM

Seat: C

Do you want to book another ticket? (yes/no): no

All Bookings:

Booking 1

Passenger: Dennish

Contact Info: 0134258907

Airline: AirAsia

Flight: Flight 1 - Origin: KLIA, Destination: SENAI, Boarding Time: 10:00 AM

Seat: C

3

```
Enter passenger name: Izmeer
Enter contact information: 0124789920

Choose Airline:
1. AirAsia
2. MAS
Enter the number of your chosen airline: 1

Origin:
1. KLIA
2. SENAI
3. TAWAU
Choose your origin: 2

Destination:
1. KLIA
2. SENAI
3. TAWAU
Choose your destination: 3

Available flights from SENAI to TAWAU with AirAsia:
4. Origin: SENAI, Destination: TAWAU, Boarding Time: 4:00 PM
Choose a flight number: 4

Available seats:
1. Seat: A
2. Seat: B
3. Seat: C
4. Seat: D
5. Seat: E
6. Seat: F
7. Seat: G
8. Seat: H
9. Seat: I
10. Seat: J
Choose a seat number: 7

Booking details:
Passenger: Izmeer
Contact Info: 0124789920
Airline: AirAsia
Flight: Flight 4 - Origin: SENAI, Destination: TAWAU, Boarding Time: 4:00 PM
Seat: G

Do you want to book another ticket? (yes/no): no

All Bookings:
Booking 1
Passenger: Izmeer
Contact Info: 0124789920
Airline: AirAsia
Flight: Flight 1 - Origin: SENAI, Destination: TAWAU, Boarding Time: 4:00 PM
Seat: G
-----
```

4

```
Enter passenger name: Aisy
Enter contact information: 0189901245

Choose Airline:
1. AirAsia
2. MAS
Enter the number of your chosen airline: 2

Origin:
1. KLIA
2. SENAI
3. TAWAU
Choose your origin: 1

Destination:
1. KLIA
2. SENAI
3. TAWAU
Choose your destination: 2

Available flights from KLIA to SENAI with MAS:
1. Origin: KLIA, Destination: SENAI, Boarding Time: 11:00 AM
Choose a flight number: 1

Available seats:
1. Seat: Q
2. Seat: R
3. Seat: S
4. Seat: T
5. Seat: U
6. Seat: V
7. Seat: W
8. Seat: X
9. Seat: Y
10. Seat: Z
Choose a seat number: 8

Booking details:
Passenger: Aisy
Contact Info: 0189901245
Airline: MAS
Flight: Flight 1 - Origin: KLIA, Destination: SENAI, Boarding Time: 11:00 AM
Seat: X

Do you want to book another ticket? (yes/no): no

All Bookings:
Booking 1
Passenger: Aisy
Contact Info: 0189901245
Airline: MAS
Flight: Flight 1 - Origin: KLIA, Destination: SENAI, Boarding Time: 11:00 AM
Seat: X
-----
```

5

```
Enter passenger name: Hitler
Enter contact information: 0112347869

Choose Airline:
1. AirAsia
2. MAS
Enter the number of your chosen airline: 2

Origin:
1. KLIA
2. SENAI
3. TAWAU
Choose your origin: 3

Destination:
1. KLIA
2. SENAI
3. TAWAU
Choose your destination: 1

Available flights from TAWAU to KLIA with MAS:
6. Origin: TAWAU, Destination: KLIA, Boarding Time: 8:00 PM
Choose a flight number: 6

Available seats:
1. Seat: Q
2. Seat: R
3. Seat: S
4. Seat: T
5. Seat: U
6. Seat: V
7. Seat: W
8. Seat: X
9. Seat: Y
10. Seat: Z
Choose a seat number: 3

Booking details:
Passenger: Hitler
Contact Info: 0112347869
Airline: MAS
Flight: Flight 6 - Origin: TAWAU, Destination: KLIA, Boarding Time: 8:00 PM
Seat: S

Do you want to book another ticket? (yes/no): no

All Bookings:
Booking 1
Passenger: Hitler
Contact Info: 0112347869
Airline: MAS
Flight: Flight 1 - Origin: TAWAU, Destination: KLIA, Boarding Time: 8:00 PM
Seat: S
-----
```

6

```
Enter passenger name: Putih
Enter contact information: 0142469986

Choose Airline:
1. AirAsia
2. MAS
Enter the number of your chosen airline: 2

Origin:
1. KLIA
2. SENAI
3. TAWAU
Choose your origin: 2

Destination:
1. KLIA
2. SENAI
3. TAWAU
Choose your destination: 3

Available flights from SENAI to TAWAU with MAS:
4. Origin: SENAI, Destination: TAWAU, Boarding Time: 6:00 PM
Choose a flight number: 4

Available seats:
1. Seat: Q
2. Seat: R
3. Seat: S
4. Seat: T
5. Seat: U
6. Seat: V
7. Seat: W
8. Seat: X
9. Seat: Y
10. Seat: Z
Choose a seat number: 10

Booking details:
Passenger: Putih
Contact Info: 0142469986
Airline: MAS
Flight: Flight 4 - Origin: SENAI, Destination: TAWAU, Boarding Time: 6:00 PM
Seat: Z

Do you want to book another ticket? (yes/no): no

All Bookings:
Booking 1
Passenger: Putih
Contact Info: 0142469986
Airline: MAS
Flight: Flight 1 - Origin: SENAI, Destination: TAWAU, Boarding Time: 6:00 PM
Seat: Z
-----
```

7.0 TASK DISTRIBUTION

Category	Task	Person In-Charge
Coding Implementation	SkyInfo Class	Muhammad Izmeer Muhammad Firdaus Muhammad Dennish Afif Aisy
	Booking Class	
	Reservation Class	
	Ticket Class	
	Passenger Class	
	Airline Class	
	Flight Class	
	Seat Class	
	SkyReserve	
Others (Diagrams, Reports & Slide)	Project Description and Conclusion	Muhammad Dennish
	Object Oriented Concepts	Muhammad Dennish
	Flow Chart	Muhammad Firdaus
	Class Diagram	Muhammad Firdaus
	User Manual	Muhammad Izmeer
	Presentation Slide	Afif Aisy

8.0 CONCLUSION

In conclusion, the Airline Reservation System (ARS) is now the best method for travelers to reserve and book flights to enjoy the effectiveness and swiftness when booking a flight reservation. ARS is not just any reservation system, it ensures travelers to feel relief about their privacy and safety about their personal data. With this, ARS also helps travelers to plan a good and smooth travel so that travelers do not strain themselves when booking flights. All these factors combine to drive advancement for ARS to evolve much better in the future.

The Airline Reservation System (ARS) is still in the process of needing technological advancements and development, adapting to evolving issues in the future and opportunities in the airline industry. Its lasting significance arises from its ability to equip airlines with the necessary resources to surpass customer expectations, streamline operations, and navigate the complexities of an unpredictable high demand market. Essentially, the ARS is not just being a mere reservation platform; it serves as a strategic system that brings efficiency, ingenuity, and improved customer satisfaction, consequently reshaping the course of the aviation sector in this era of digital transformation.