UNIVERSITY OF MALAYSIA TERENGGANU

FACULTY OF COMPUTER SCIENCE AND MATHEMATICS

COURSE : SYSTEM THINKING AND LOGIC CSF3023

SESSION : SEMESTER 1 2024/2025

PROGRAMME : BANCHELOR OF COMPUTER SCIENCE (SOFTWARE ENGINEERING)

LECTURER NAME : PROFESSOR. TS. DATO' DR AZIZ DERAMAN

GROUP 3 MEMBERS :

NAME	NO. MATRICS
DANIELIN ALPHONSE A/P MARIAPPAN	S78561
TUAN WAFA HANANI BINTI TUAN ROHAIZAT	S78902
DAMIA YASARAH BINTI AMRI	S78568
HAZIQ IZUDDIN BIN RUSLI	S78693

TABLE OF CONTENTS

1. INTRODUCTION …….3
    - 1.1 Online flight booking system

2. FUNCTIONAL REQUIREMENT…….4
    - 2.1 User registration & login……..4-5
    - 2.2 Search available flights…….6-7
    - 2.3 Select flights & seats……8-9
    - 2.4 Payment processing……...10-11
    - 2.5 Booking confirmation……12-13

3. CONCLUSION…….……14

4. REFERENCES ……..……..15

1. INTRODUCTION

1.1 ONLINE FLIGHT BOOKING SYSTEM

In today's highly competitive travel industry, online flight booking systems are essential for streamlining flight-related operations. These platforms have become indispensable, offering travelers comprehensive tools that simplify planning, provide personalized options, and enhance the overall travel experience (Bharti et al., 2024). This system was chosen as the target for this assignment in order to invoke system thinking and logical problem solving which are critical for this course. Historically, booking a flight necessitated the painstaking manual efforts of travel agents or direct visits to airline offices, a method that was not only laborious but also prone to inefficiencies and human errors. The journey to develop a web-based service that would automate data transmission and storage, as well as facilitate real-time interaction between airlines and passengers during flights, required some time.

The system's primary objective is to ensure an effortless interaction between the users and the airline by assisting them in searching and booking airline tickets. By replacing manual record-keeping with computerized databases, the system ensures that passenger details and flight schedules are accurately stored and easily retrieved. The system enables customers to book flights remotely without the need to visit the airline offices. It also allows airline administrators to organize and manage available resources efficiently. Therefore, the system seeks to refine the operational efficiency of service providers in the aviation industry.

Within this setup, two primary categories of users interact with the system: the travelers (end-users) and the airline administrators (back-end users). Travelers use the front-end interface to compare prices and check availability, while administrators access the back-end system to oversee schedule updates. In accordance with the project guidelines, the following sections of this report will examine this system (after breaking it down into its functional components) and display the logic in flowcharts and pseudocode.

2. FUNCTIONAL REQUIREMENTS

2.1 USER REGISTRATION & LOGIN

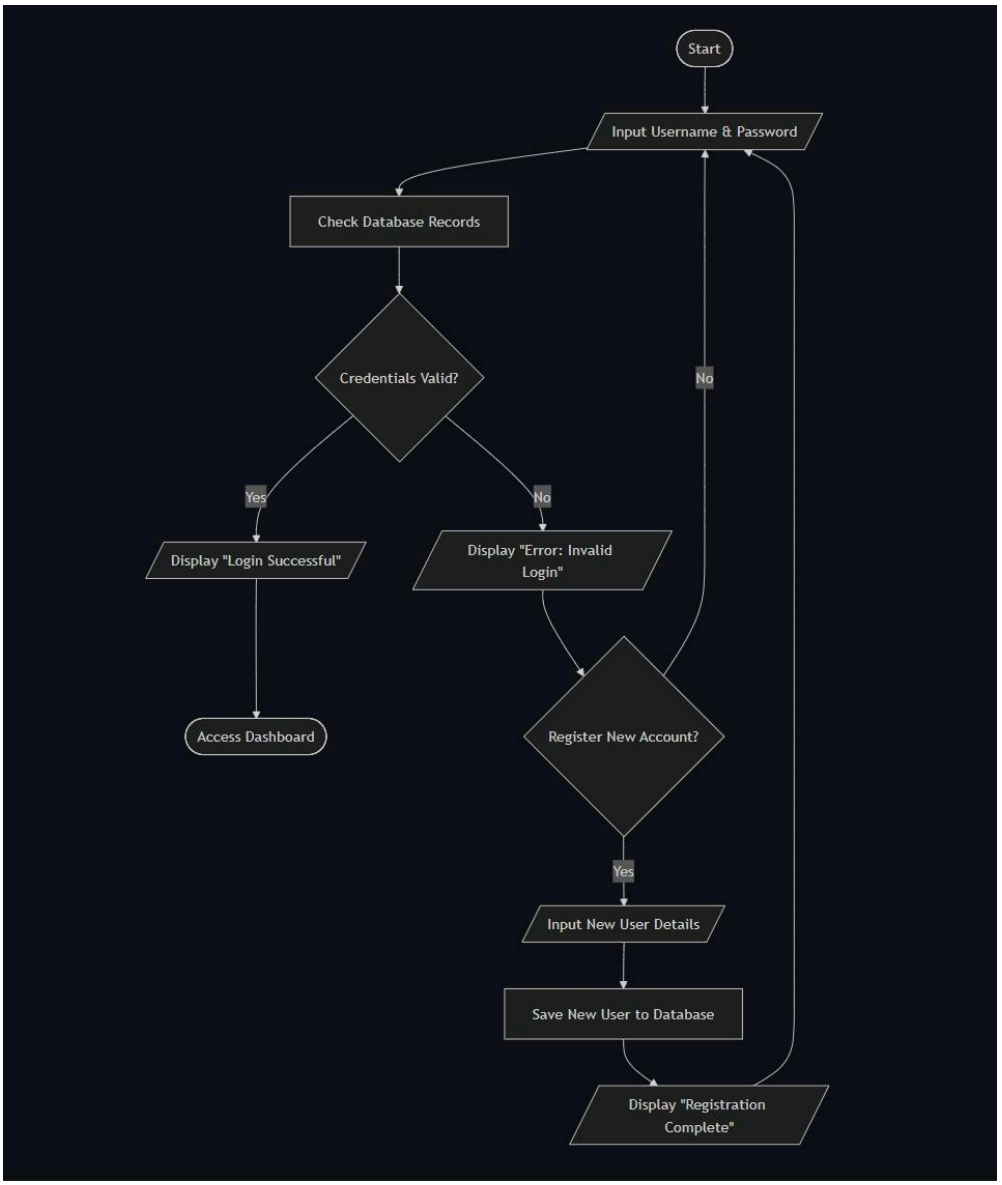
The Online Flight Booking System can be accessed using user registration & login. By providing personal information like name, email address, and password, this will then be kept secure in the system. During the register, users will have to use credentials and checks to ensure no unwanted access happens. This will be verified by the system, and secure the user's account access. This feature permits tracking of user activities and guarantees data security.

PSEUDOCODE :

```
START
    PRINT "1. Login | 2. Register"
    INPUT choice
    IF choice EQUALS "2" THEN
        PRINT "Enter New Username & Password:"
        INPUT new_user, new_pass
        IF new_user EXISTS IN database THEN
            PRINT "Error: Username taken"
        ELSE
            SAVE new_user, new_pass TO database
            PRINT "Registration Successful"
        ENDIF
    ELSE IF choice EQUALS "1" THEN
        PRINT "Enter Username & Password:"
        INPUT user, pass
        IF user EXISTS IN database AND pass MATCHES stored_pass THEN
            PRINT "Login Successful"
            CALL Search_Flights_Function()
        ELSE
            PRINT "Error: Invalid Credentials"
        ENDIF
    ENDIF
```

23. END

FLOWCHART :



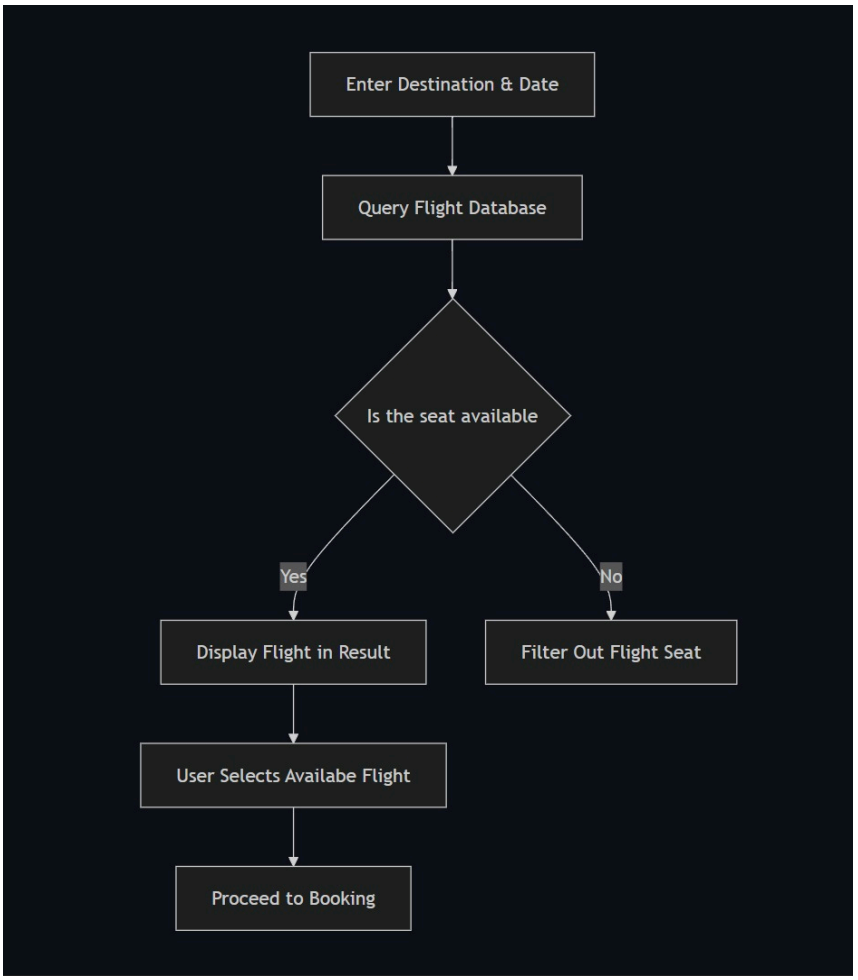
2.2 SEARCH AVAILABLE FLIGHTS

Users then proceed with searching available flights by using keywords such as their travel destination and date to book. The system then displays matching flights including the necessary information, such as flight schedule, cost and seat availability. Other than that, for fully booked flights, flight availability, and seat availability will be shown with appropriate messages such as “Full Flight”, "No flights available for these criteria." and “Available”. This feature ensures users easy browsing during their search for the suitable flights to travel with.

PSEUDOCODE:

```
1. BEGIN
2. INPUT travelDestination
3. INPUT travelDate
4. SET matchingFlights = Query flightDatabase WHERE destination = travelDestination
   AND date = travelDate
5. FOR flight IN
6.       matchingFlights DO
7.           IF flight.availableSeats > 0 THEN
8.               DISPLAY flight.details
9.               DISPLAY "Status: Available"
10.       ELSE
11.              DISPLAY "Status: Full Flight"
12.          END IF
13. END FOR
14. IF userResult IS NOT EMPTY THEN
15.     WAIT for UserSelection
16.       IF UserSelectsFlight THEN
17.          PROCEED to BookingProcess
18.       END IF
19.    ELSE
20.       DISPLAY "No flights available for these criteria."
21.    END IF
22. END
```

FLOWCHART:


Enter Destination & Date
Query Flight Database
Is the seat available
Yes
No
Display Flight in Result
Filter Out Flight Seat
User Selects Availabe Flight
Proceed to Booking

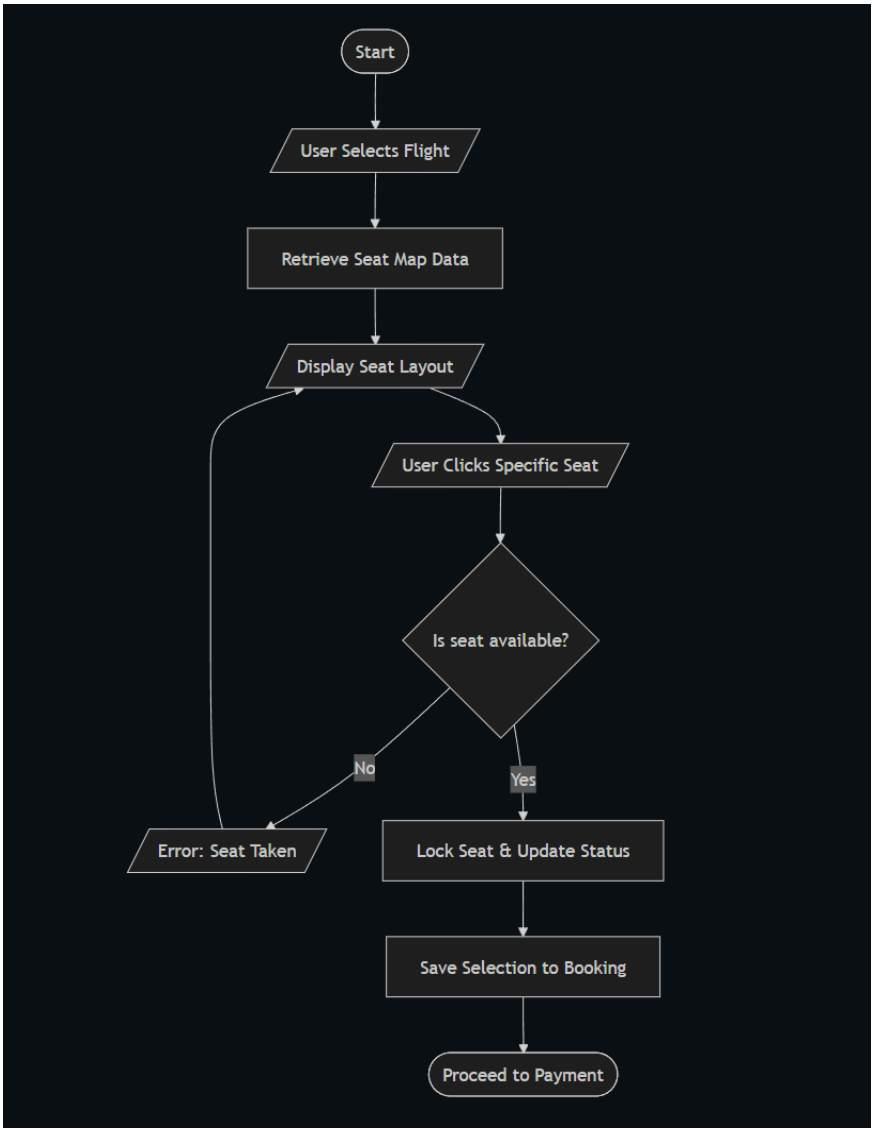
2.3 SELECT FLIGHTS & SEATS

As available flights are found, users will be prompted to select a specific flight from the results. Upon selection, the system retrieves the current seat map from the database and displays the layout. The user can view which seats are "Available" or "Occupied". When a user clicks a seat, the system verifies its status in real-time. If available, the seat is locked to prevent double-booking; if not, an error is shown.

PSEUDOCODE:

```
1.  START Function SelectSeat
2.      INPUT selectedFlight_ID
3.      CALL GetSeatMap(selectedFlight_ID)
4.      DISPLAY SeatMap
5.      INPUT userSeatSelection
6.      IF SeatMap[userSeatSelection] == "Available" THEN
7.          UPDATE SeatMap SET status = "Reserved"
8.          ADD selection TO CurrentBooking
9.          PROCEED to Payment
10.     ELSE
11.         DISPLAY "Seat taken, please choose another."
12.         RETURN to SeatMap
13.     END IF
14. END Function
```

FLOWCHART:


Start
User Selects Flight
Retrieve Seat Map Data
Display Seat Layout
User Clicks Specific Seat
Is seat available?
No
Yes
Error: Seat Taken
Lock Seat & Update Status
Save Selection to Booking
Proceed to Payment

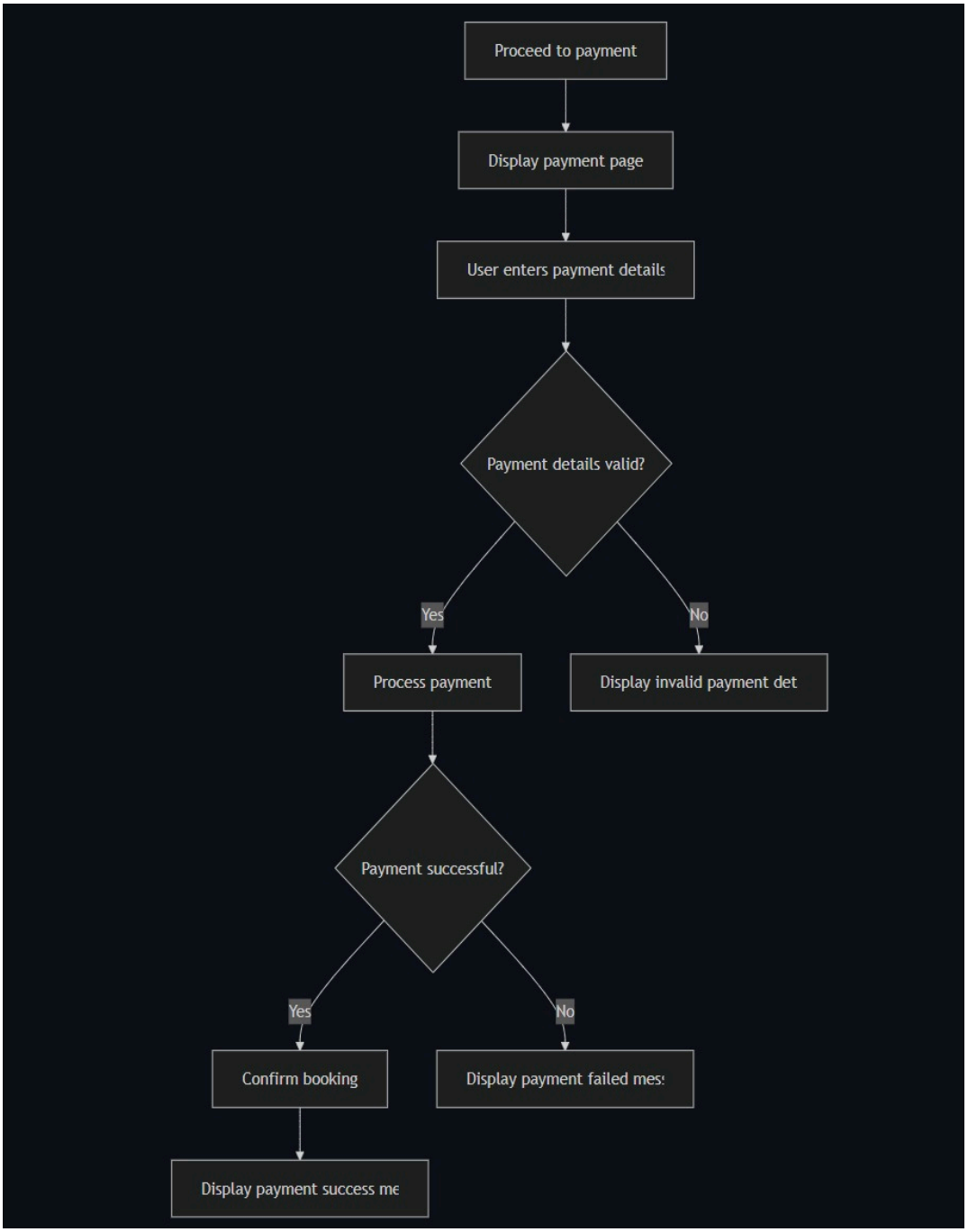
2.4 PAYMENT PROCSESSING

The system will be able to enable the user to pay the flight booking they chose. Once the user has made the choice of the flight, the system will show a payment page where the user will be expected to key in details like the card number or wherever the user has bank accounts online. The system verifies the payment information entered and forwards payment requests to the payment gateway. In case of a successful payment, the system makes a confirmation of the booking and the user is provided with a message of the payment success. In case of failure of payment or invalidity of payment details, then the system gives a suitable message of error and provides the user with an option of re-making the payment.

PSEUDOCODE :

1. START
2. DISPLAY payment page
3. GET payment details
4. READ payment details
5. IF payment is valid THEN
6. Process payment
7. IF payment successful THEN
8. Confirm booking
9. Display payment success message
10. ELSE
11. Display payment failed message
12. END IF
13. ELSE
14. Display invalid payment details message
15. END IF
16. END

FLOWCHART :


Proceed to payment
Display payment page
User enters payment details
Payment details valid?
Yes
No
Process payment
Display invalid payment det
Payment successful?
Yes
No
Confirm booking
Display payment failed mes
Display payment success me

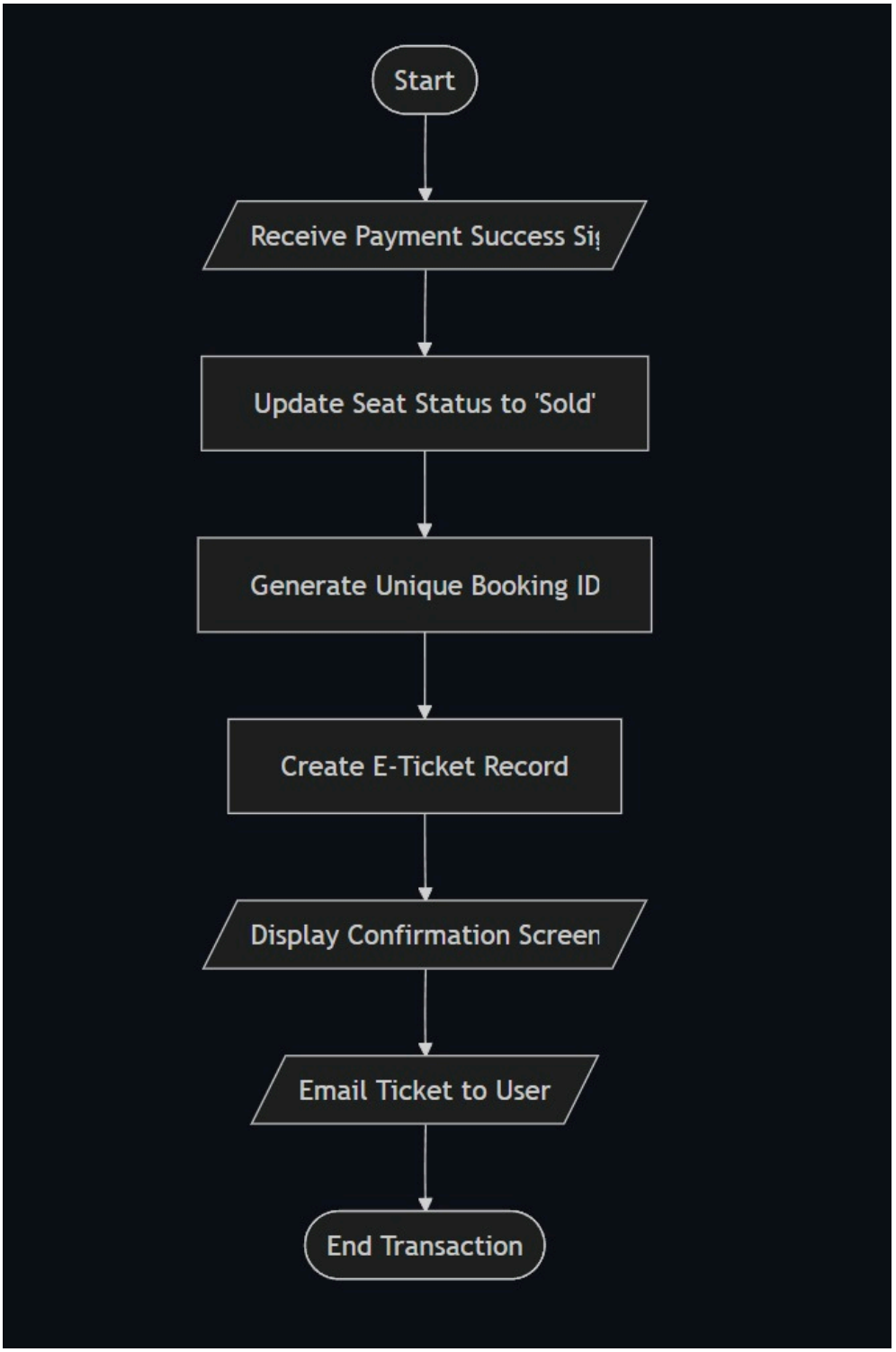
2.5 BOOKING CONFIRMATION

The system shall generate a booking confirmation after the payment for a flight is successfully completed. Once the payment is confirmed, the system records the booking details and generates a confirmation message containing information such as booking ID, flight details, passenger details, and payment status. The system then displays the booking confirmation to the user and provides an electronic ticket (e-ticket) or confirmation receipt for future reference.

PSEUDOCODE :

```
1. START
2.     INPUT payment_status
3.     INPUT selected_flight_id
4.     INPUT passenger_details
5.     IF payment_status EQUALS "SUCCESS" THEN
6.         UPDATE database_flights
7.         SET seat_availability = "Sold"
8.         WHERE flight_id = selected_flight_id
9.         SET booking_ref = GENERATE_RANDOM_STRING(6)
10.  SET new_ticket=CONSTRUCT_TICKET(booking_ref,passenger_details, selected_flight_id)
11.        SAVE new_ticket TO database_bookings
12.        PRINT "Booking Successful!"
13.        PRINT "Your Booking Reference is: " + booking_ref
14.        DISPLAY new_ticket
15.        CALL Send_Email(passenger_details.email, new_ticket)
16.    ELSE
17.        PRINT "Confirmation Failed: Payment not verified."
18.        GOTO Payment_Page
19.    ENDIF
20. END
```

FLOWCHART :


Start
Receive Payment Success Si
Update Seat Status to 'Sold'
Generate Unique Booking ID
Create E-Ticket Record
Display Confirmation Screen
Email Ticket to User
End Transaction

3. CONCLUSION

In conclusion, this project successfully applied system thinking and logical problem-solving concepts to the design of an online flight booking system. The system was analysed by breaking it down into five key functional requirements, allowing a clear understanding of how each component contributes to the overall operation of the system. This approach helped transform a complex real-world process into a structured and manageable system model.

Each functional requirement was clearly defined and supported with corresponding flowcharts and pseudocode. The flowcharts provided a visual representation of the system processes, making it easier to understand decision points and process flows. Meanwhile, the pseudocode translated these visual steps into logical instructions, ensuring that the system behaviour was clearly and systematically represented.

The use of GitHub as a collaboration platform enhanced teamwork and version control throughout the project. By assigning individual responsibilities and maintaining organised folders, the group was able to collaborate effectively while ensuring transparency in contributions. This also reflected real-world software development practices, where documentation and collaboration tools play a crucial role.

Overall, this project strengthened the group's understanding of system design, logical sequencing, and functional analysis. The online flight booking system designed in this report demonstrates how system thinking principles can be applied to model practical applications accurately and effectively. The knowledge gained from this project provides a strong foundation for analysing and designing more complex systems in future coursework and professional environments.

4. REFERENCE

1. Laudon, K. C., & Laudon, J. P. (2020). *Management Information Systems: Managing the Digital Firm* (16th ed.). Pearson Education. Retrieved from https://www.pearson.com/se/Nordics-Higher-Education/subject-catalogue/information-systems/Laudon-management-information-systems-digital-firm-16e.html

2. Sommerville, I. (2016). *Software Engineering* (10th ed.). Pearson Education. Retrieved from https://dn790001.ca.archive.org/0/items/bme-vik-konyvek/Software%20Engineering%20-%20Ian%20Sommerville.pdf

3. *Flowcharts - Basic Syntax | Mermaid*. (n.d.). https://emersonbottero.github.io/mermaid-docs/syntax/flowchart.html

4. Bharti, S., Nayeem, T., Rathore, S., Tripathi, S., Rathore, R., & Sharma, K. (2024). *Online flight and hotel booking with recommendation website: Review paper*. International Journal for Multidisciplinary Research, 6(2). https://www.ijfmr.com/papers/2024/2/18594.pdf

5. Gaddis, T. (2019). Starting Out with Programming Logic and Design (5th ed.). Pearson. https://www.pearson.com/en-us/subject-catalog/p/starting-out-with-programming-logic-and-design/P200000003348