"AIRLINE BOOKING RESERVATION"

INDEX

“AIRLINE BOOKING RESERVATION”

An "Airline Booking Reservation" project is a software application designed to facilitate the booking of airline tickets for travellers. This project involves the development of a comprehensive system that allows passengers to search for flights, reserve seats, make payments, and obtain tickets for their chosen flights.

Significance:

Air travel is a fundamental mode of transportation in today's world, connecting people and businesses across the globe.

An airline booking reservation system streamlines the flight booking process, making it more efficient, convenient, and accessible to a wide range of travellers.

Purpose:

The primary purpose of the "Airline Booking Reservation" project is to provide a user-friendly and efficient platform for passengers to:

Search for available flights based on criteria like destination, date, and class. Reserve seats on preferred flights. Make secure online payments for their bookings. Receive electronic tickets for their confirmed reservations.

Functionality:

- **Flight Search and Selection:**

Passengers can enter their travel details (origin, destination, date, and passenger count) to search for available flights. The system presents a list of flights meeting

the search criteria, showing details such as airline, departure/arrival times, and ticket prices. Passengers can select their preferred flight based on these options.

- **Seat Reservation:**

Passengers can view a seat map for the selected flight, indicating available and occupied seats. They can choose their desired seats and reserve them.

- **Payment Processing:**

The system offers secure payment options, including credit card, debit card, or digital payment methods. Passengers can complete the payment process, ensuring their bookings are confirmed.

- **Ticket Generation:**

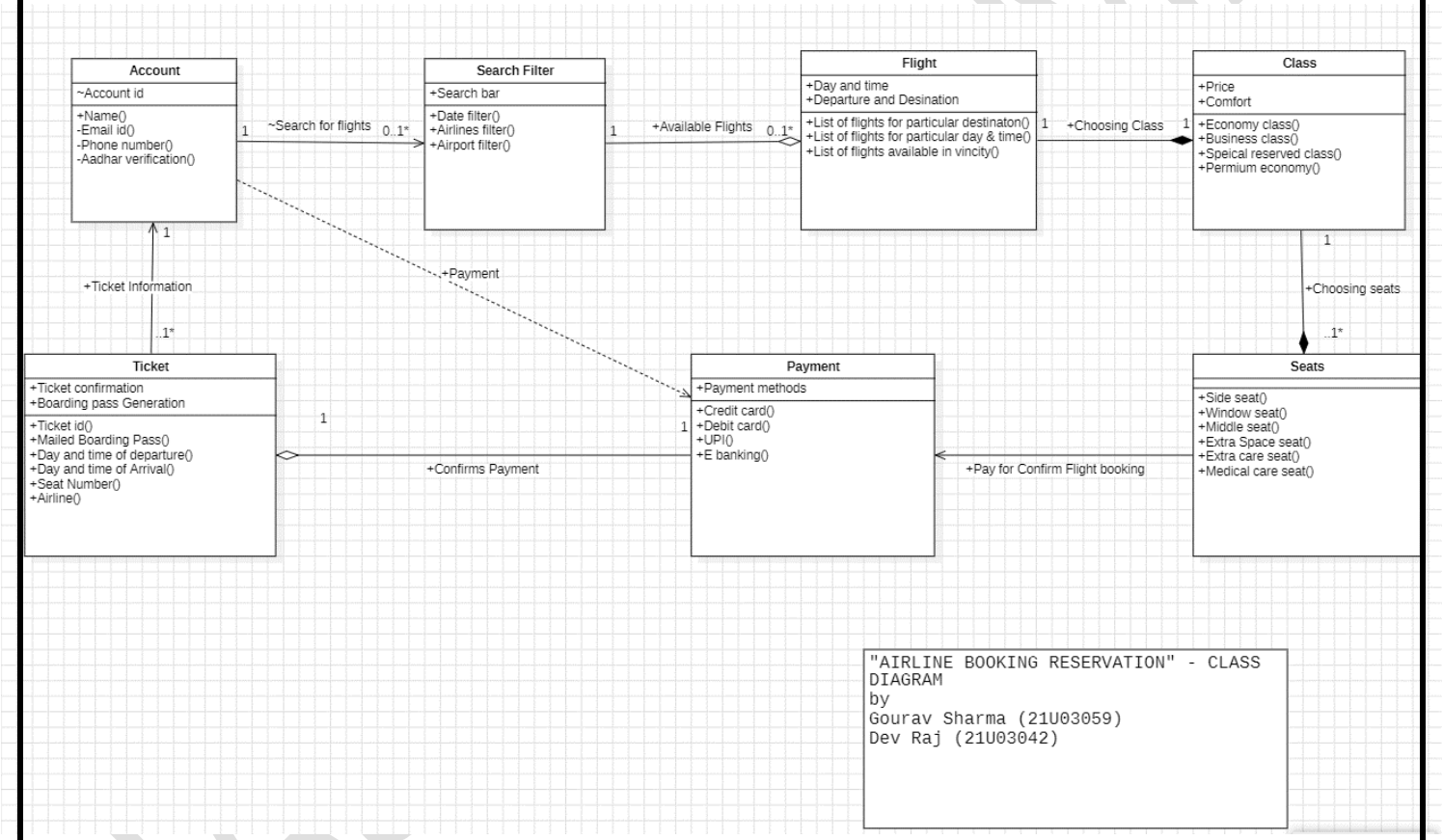
Once payment is successful, the system generates electronic tickets (e-tickets) for the passengers. Passengers receive these e-tickets via email or through their account on the platform.

- **User Accounts:**

Passengers can create accounts, which store their booking history and personal information for faster future bookings. Registered users may receive loyalty rewards or frequent flyer benefits.

Class Diagram:

Class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing, and documenting different aspects of a system but also for constructing executable code of the software application.



1. **Account**:
    - The "Account" class represents a user's account in the system.
    - States:
        - **Logged Out**: When the user is not logged into their account.
        - **Logged In**: When the user successfully logs into their account.
        - **Profile Updated**: When the user updates their account profile information.
    - Transitions:

- User logs in: Transition from "Logged Out" to "Logged In."
- User logs out: Transition from "Logged In" to "Logged Out."
- User updates profile: Transition from "Logged In" to "Profile Updated."

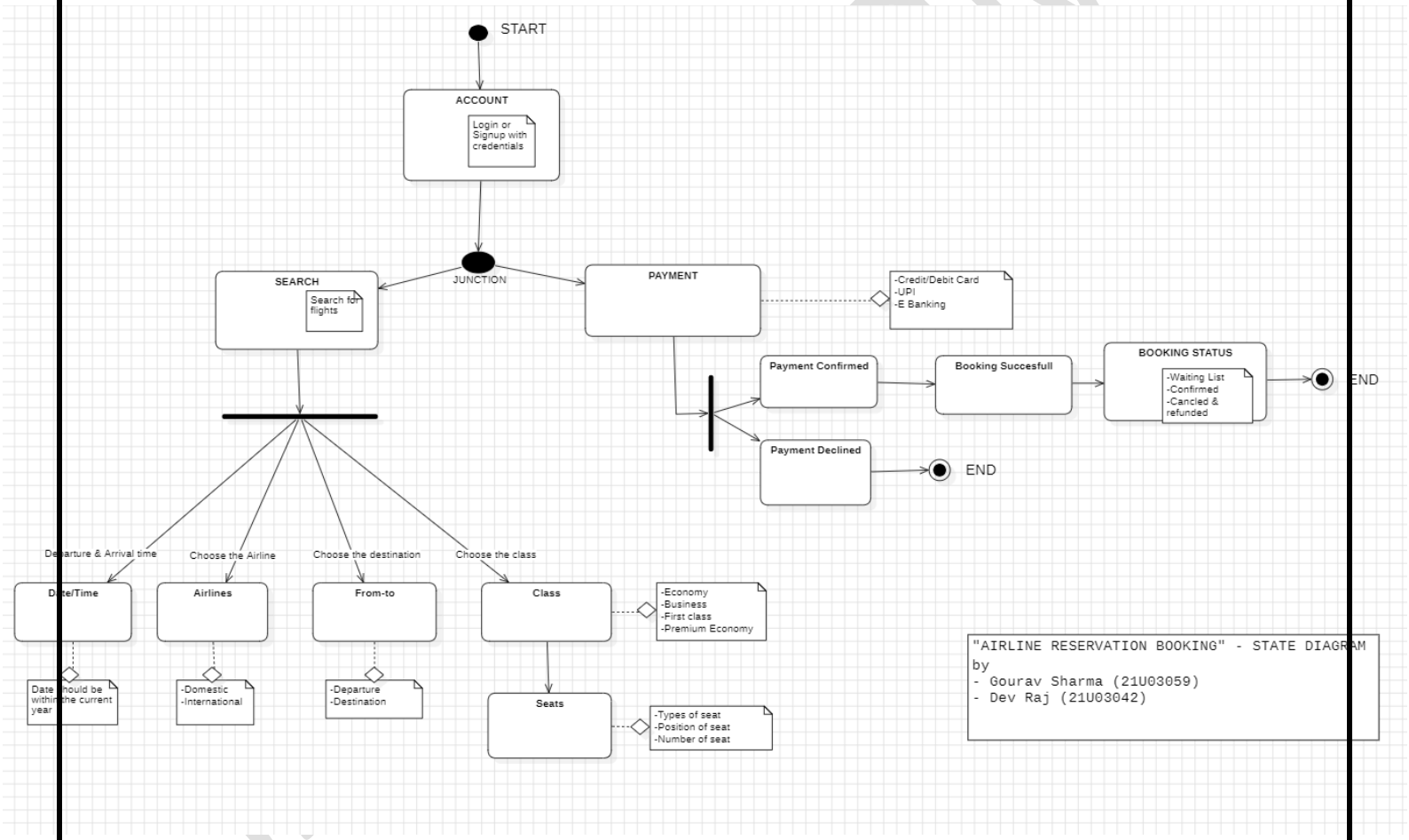
2. **Search Filter**:
   - The "Search Filter" class is used to specify search criteria for flights.
   - States:
     - **Default**: Initial state, indicating no specific search criteria set.
     - **Filter Applied**: When the user applies specific search filters.
   - Transitions:
     - User applies filters: Transition from "Default" to "Filter Applied."
     - User clears filters: Transition from "Filter Applied" to "Default."
3. **Flight**:
   - The "Flight" class represents individual flights available for booking.
   - States:
     - **Available**: When the flight is open for booking.
     - **Fully Booked**: When all seats on the flight are booked.
   - Transitions:
     - Flight opens for booking: Transition from "Fully Booked" to "Available."
     - Flight becomes fully booked: Transition from "Available" to "Fully Booked."
4. **Class**:
   - The "Class" class defines the different classes available on a flight (e.g., Economy, Business, First Class).
   - States:
     - **Available**: When seats in this class are available.
     - **Fully Booked**: When all seats in this class are booked.
   - Transitions:
     - Seats in this class become available: Transition from "Fully Booked" to "Available."

        - All seats in this class are booked: Transition from "Available" to "Fully Booked."

5. **Seats**:
    - The "Seats" class represents individual seat reservations on a flight.
    - States:
        - **Available**: When the seat is available for booking.
        - **Reserved**: When the seat is reserved by a passenger.
    - Transitions:
        - Seat becomes available: Transition from "Reserved" to "Available."
        - Seat is reserved by a passenger: Transition from "Available" to "Reserved."
6. **Payment**:
    - The "Payment" class handles the payment process for booking a flight.
    - States:
        - **Initial**: When payment process is initiated.
        - **Payment Successful**: When payment is successfully completed.
        - **Payment Failed**: When payment fails.
    - Transitions:
        - Payment process initiated: Transition from "Initial" to either "Payment Successful" or "Payment Failed" based on the payment outcome.
7. **Ticket**:
    - The "Ticket" class represents the passenger's ticket for a booked flight.
    - States:
        - **Issued**: When the ticket is issued after successful booking and payment.
    - Transitions:
        - Ticket is issued: Transition from "Issued."

State Diagram:

The state graph or state diagram is a pictorial representation of the relationships between the present state, the input state, the next state, and the output state of a sequential circuit.



1. **Account State**:
    - This state represents the user's account management within the system.
    - Substates:
        - **Logged In**: When the user is successfully logged into their account.
        - **Logged Out**: When the user is not logged into their account.

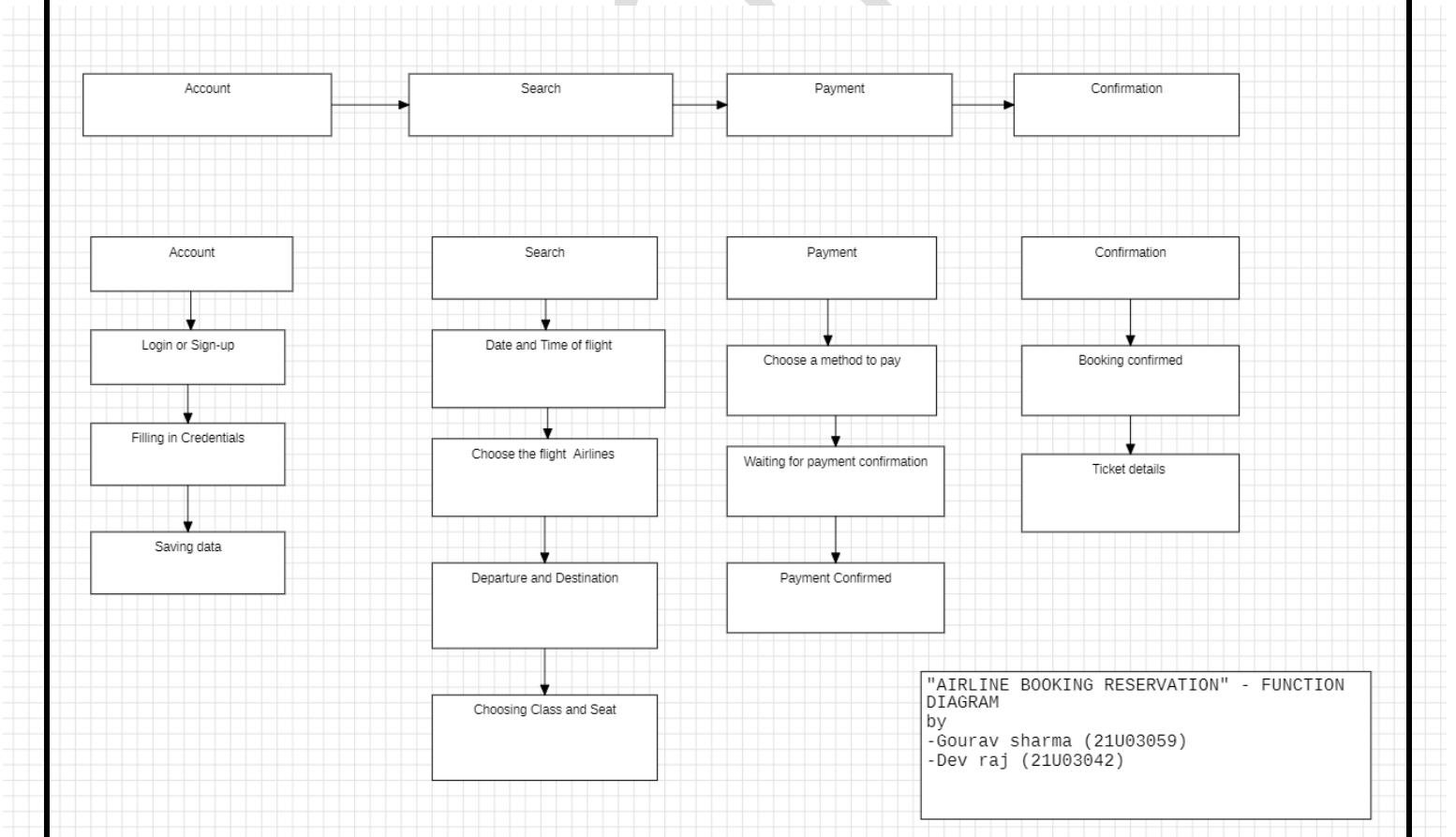
2. **Search State**:
    - This state encompasses the process of searching for available flights.
    - Substates:
        - **Date/Time Selection**: The user selects the desired travel date and time.
        - **Airlines Selection**: The user filters flights by preferred airlines.
        - **From-To Selection**: The user specifies the departure and destination cities.
        - **Class Selection**: The user chooses the travel class (e.g., Economy, Business, First Class).
        - **Seats Selection**: The user selects preferred seats on the chosen flight.
3. **Payment State**:
    - This state represents the payment process for the selected flight.
    - Substates:
        - **Payment Confirmed**: The user's payment is successfully processed.
        - **Payment Declined**: The user's payment transaction is declined.
4. **Transitions between States**:

- Logged Out to Logged in.
- Searching for flights.
- Date/Time Selection to Airlines Selection.
- Airlines Selection to From-To Selection.
- From-To Selection to Class Selection.
- Class Selection to Seats Selection.
- Seats Selection to Payment Confirmed or Payment Declined.
- Payment Confirmed to Booking Successful.
- Booking Successful to Booking Status.
- Booking Status to End.

Function Diagram:

It defines the function of the internal processes in the system with the aid of Data Flow Diagrams (DFDs). It depicts the functional derivation of the data values without indicating how they are derived when they are computed, or why they need to be computed. A data flow diagram is a graph which shows the flow of data values from their sources in objects through processes that transform them to their destinations on other objects.

Account State:



- **Login/Sign Up**:

This function allows users to either log in to their existing account or sign up as new users.

1. User clicks on "Login" or "Sign Up" option.
2. If logging in, user enters their username and password.
3. If signing up, user provides necessary information to create a new account.
4. System verifies login credentials or creates a new account.
5. User is either logged in or the account creation is confirmed.

- **Filling Credentials**:

  This function involves users filling in their personal and contact information as part of the account creation process.

1. After choosing to sign up, users provide their name, email, phone number, and other required details.
2. The system validates and saves this information to create a user profile.

- **Saving Data**:

This function involves saving user account data securely for future reference.

1. User account information, including login credentials and personal details, is stored securely in the system's database.
2. User data is associated with a unique identifier, such as a user ID.

Search State:

- **Date and Time of Flight**:

This function allows users to specify the date and time they wish to travel.

1. User selects the date and time from available options or a calendar interface.
2. The system records the user's travel preferences.

- **Choose the Flight Airlines**:

This function enables users to filter and select flights based on preferred airlines.

1. User selects one or more airlines from a list of available options.
2. The system filters search results based on the chosen airlines.

- **Departure and Destination**:

This function allows users to specify the departure and destination airports or cities.

1. User enters departure and destination locations in designated fields.
2. The system considers these locations in the flight search.

- **Choosing Class and Seat**:

This function permits users to select their preferred travel class (e.g., Economy, Business, First Class) and seat on the chosen flight.

1. User selects a travel class and seating preferences.
2. The system presents available seats based on the user's choices

Payment State:

- **Choose a Method to Pay**:

This function allows users to select their preferred payment method.

1. User chooses from available payment options, such as credit card, debit card, or digital wallet.
2. The system provides the necessary fields for payment information.

- **Waiting for Payment Confirmation**:

This function involves users waiting for the system to process their payment.

1. User submits payment details.
2. The system initiates the payment transaction.
3. User waits for the payment processing to complete.

- **Payment Confirmed**:

This function indicates that the payment has been successfully confirmed.

1. Upon successful payment processing, the system generates a confirmation message.
2. User receives confirmation of the payment and booking.

Confirmation State:

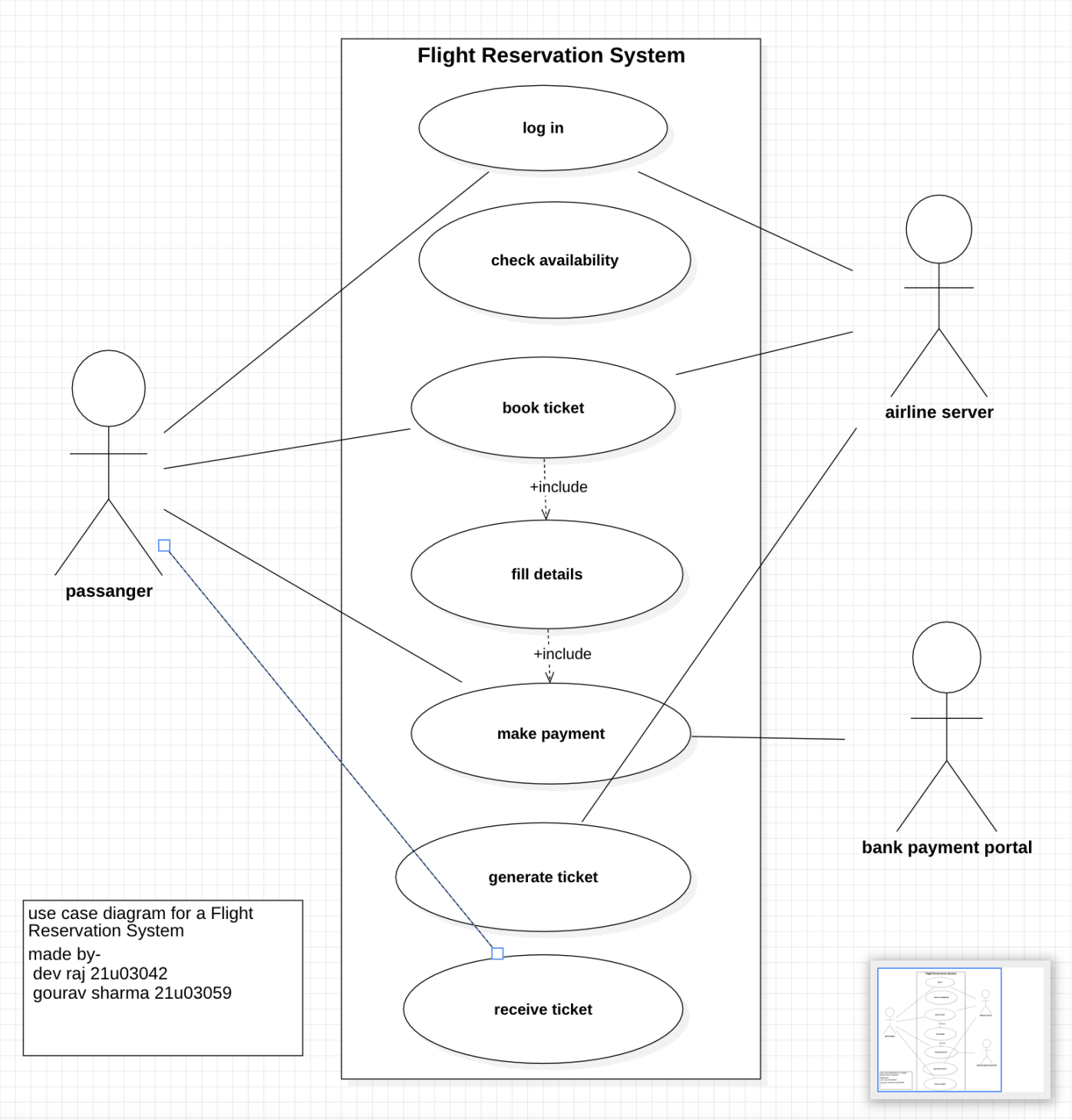
- **Booking Confirmation:**

This function provides users with a confirmation of their flight booking.

1. The system generates a booking confirmation page or message.
2. User views and acknowledges the booking details.
3. The confirmation includes flight itinerary, booking reference, and payment information.

Use Cases:

A use case is a written description of how users will perform tasks on your website. It outlines, from a user's point of view, a system's behaviour as it responds to a request. Each use case is represented as a sequence of simple steps, beginning with a user's goal and ending when that goal is fulfilled.



1. **Passenger:**
    - **Log In:** The Passenger initiates the interaction with the system by logging in. This use case allows the Passenger to access their account and start the booking process.
2. **Airline Server:**
    - **Check Availability:** The Airline Server is responsible for providing information on available flights based on the Passenger's preferences, such as date, time,

destination, and class. This use case helps the Passenger browse and select flights.

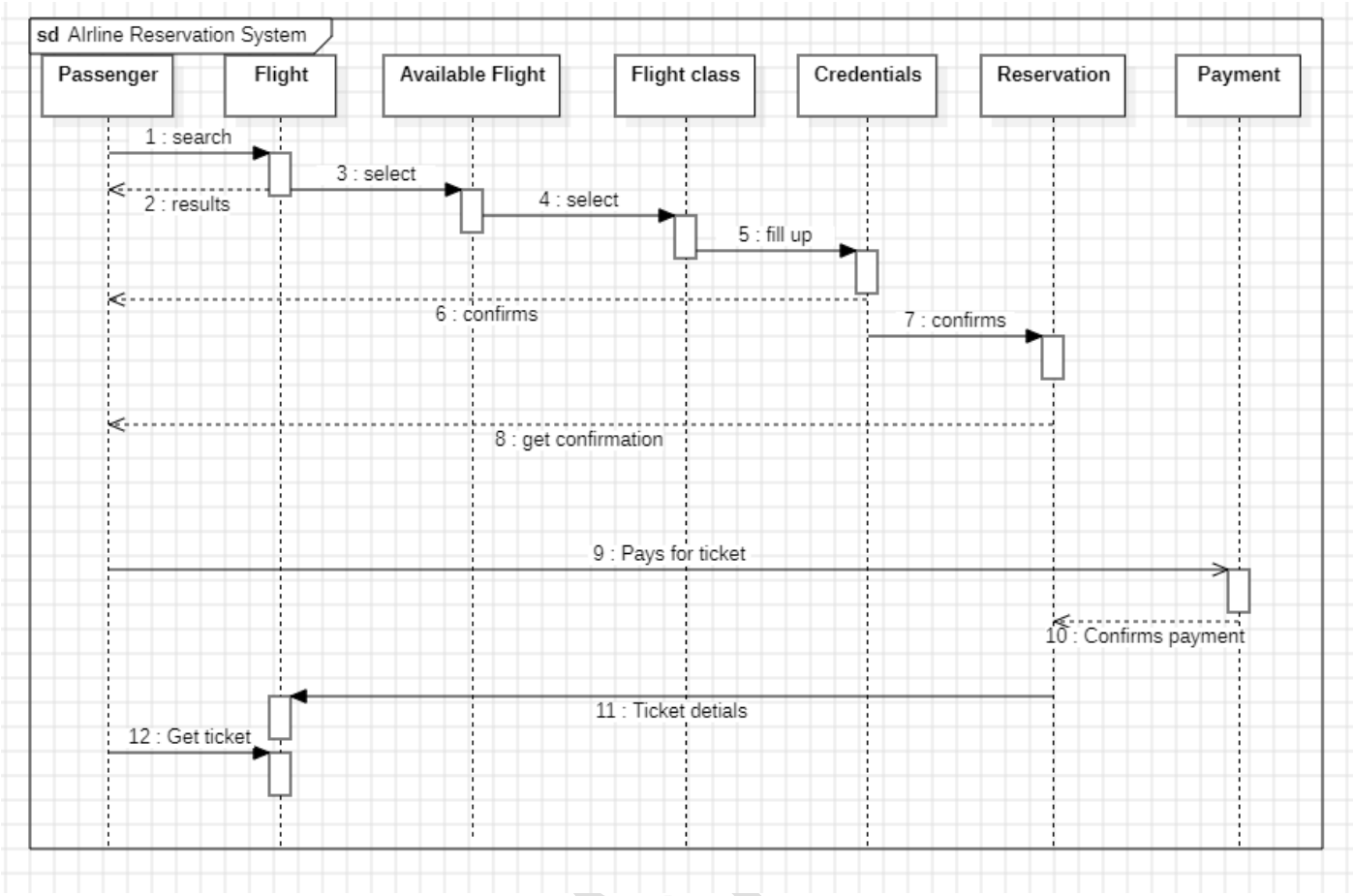
- **Book Ticket:** Once the Passenger has chosen a flight, the Airline Server processes the booking request, reserving a seat on the selected flight.
- **Fill Details:** The Passenger provides necessary personal and booking information, such as names, passport details, and contact information, to complete the booking process.
- **Generate Ticket:** After receiving the required details, the Airline Server generates an electronic ticket for the Passenger. This ticket confirms the booking and contains essential flight information.

3. **Bank Payment Portal:**
    - **Make Payment:** The Bank Payment Portal handles the financial transaction associated with the booking. The Passenger provides payment details (credit/debit card, bank account, etc.), and the Bank Payment Portal processes the payment securely.
4. **States (representing various system states or processes):**
    - **Receive Ticket:** Once the payment is successfully processed, the system moves to this state to deliver the electronic ticket to the Passenger. The Passenger receives the ticket via email or on the system interface.

In summary, this use case diagram illustrates the interaction between a Passenger and the airline's reservation system, highlighting the key steps in booking a flight. The Airline Server facilitates flight selection, booking, and ticket generation, while the Bank Payment Portal manages the financial transaction. The States represent different phases or processes within the system. This diagram helps stakeholders understand how the various actors and system components work together to enable passengers to book flights seamlessly.

Sequence Diagram:

A sequence diagram is a powerful visual tool used in software engineering to depict the interactions between different components or objects in a system.



1. **Passenger:** The "Passenger" lifeline represents an individual looking to make an airline reservation. They initiate the interaction by starting the process.
2. **Credentials:** The "Credentials" lifeline is responsible for authenticating the passenger. When a passenger interacts with the system, they must provide their credentials, such as a username and password, to log in.
3. **Available Flights:** The "Available Flights" lifeline represents the system's component responsible for providing a list of flights that match the passenger's search criteria. The "Passenger" interacts with this lifeline to search for flights based on various parameters like date, destination, and flight class.
4. **Flight Class:** The "Flight Class" lifeline deals with the passenger's choice of class (economy, business, etc.). Once the passenger selects a flight, they may choose their preferred class. This choice interacts with the "Available Flights" lifeline, which might filter the options accordingly.
5. **Flight:** The "Flight" lifeline represents an individual flight within the system. When a passenger selects a flight, they're essentially interacting with this lifeline to view the flight details, including the departure time, arrival time, and price.

6. **Reservations**: The "Reservations" lifeline is responsible for handling the creation and management of reservations. After choosing a flight and specifying class preferences, the "Passenger" interacts with this lifeline to make a reservation.
7. **Payment:** The "Payment" lifeline manages the financial aspect of the process. When a reservation is made, the passenger interacts with "Payment" to provide payment information, including credit card details, to complete the booking.

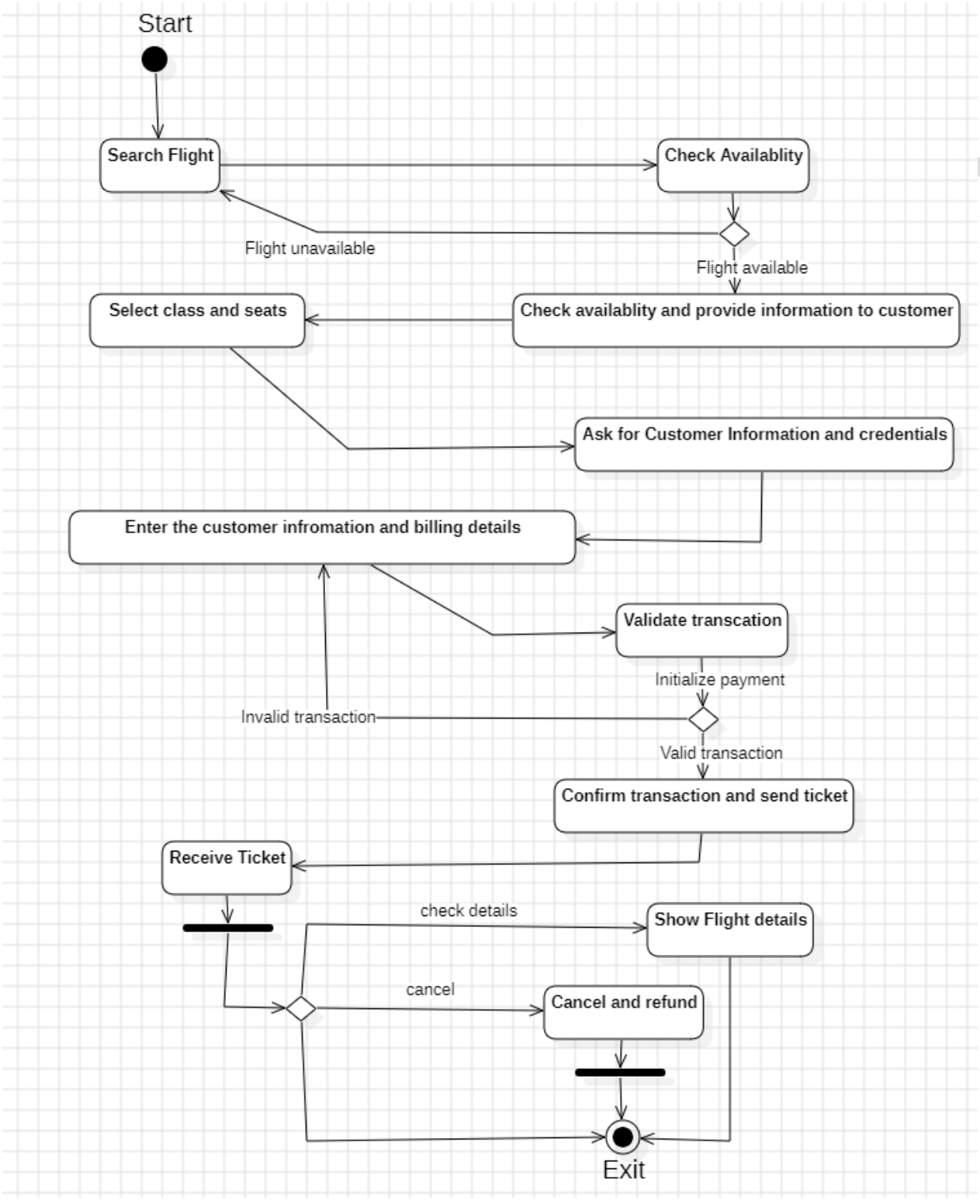
Here's an explanation of how these lifelines interact in the sequence diagram:

1. The "Passenger" begins the process by logging in with their "Credentials" or can simply search for the flights.
2. Then, the "Passenger" interacts with the "Available Flights" to search for suitable flights.
3. The "Available Flights" fetches flight options and presents them to the "Passenger."
4. The "Passenger" selects a flight and interacts with the "Flight Class" lifeline to choose their preferred class.
5. "Flight Class" updates the options based on the passenger's selection.
6. The "Passenger" selects a specific flight, and the "Flight" lifeline provides detailed information about that flight.
7. The "Passenger" confirms their choice, and create and account with proper credentials and the "Reservations" lifeline records the booking.
8. To finalize the reservation, the "Passenger" interacts with "Payment" to provide payment details.
9. "Payment" processes the payment information, confirms the booking, and provides a confirmation to the "Passenger."

This sequence diagram visually represents the dynamic interactions and order of operations in an airline reservation system, providing a clear overview of how different lifelines collaborate to facilitate the booking process.

Activity Diagram:

An activity diagram is a type of UML (Unified Modelling Language) diagram used to model the workflow or activities in a system, showing the flow of control and data between different activities. In the context of an airline reservation system, an activity diagram can help illustrate the various activities and their interrelationships.



Start and End: An activity diagram typically begins with a rounded rectangle representing the start point and ends with another rounded rectangle representing the end point. In the

case of an airline reservation system, the start point might be the initiation of the reservation process, and the end point would represent the completion of the reservation.

Activities:

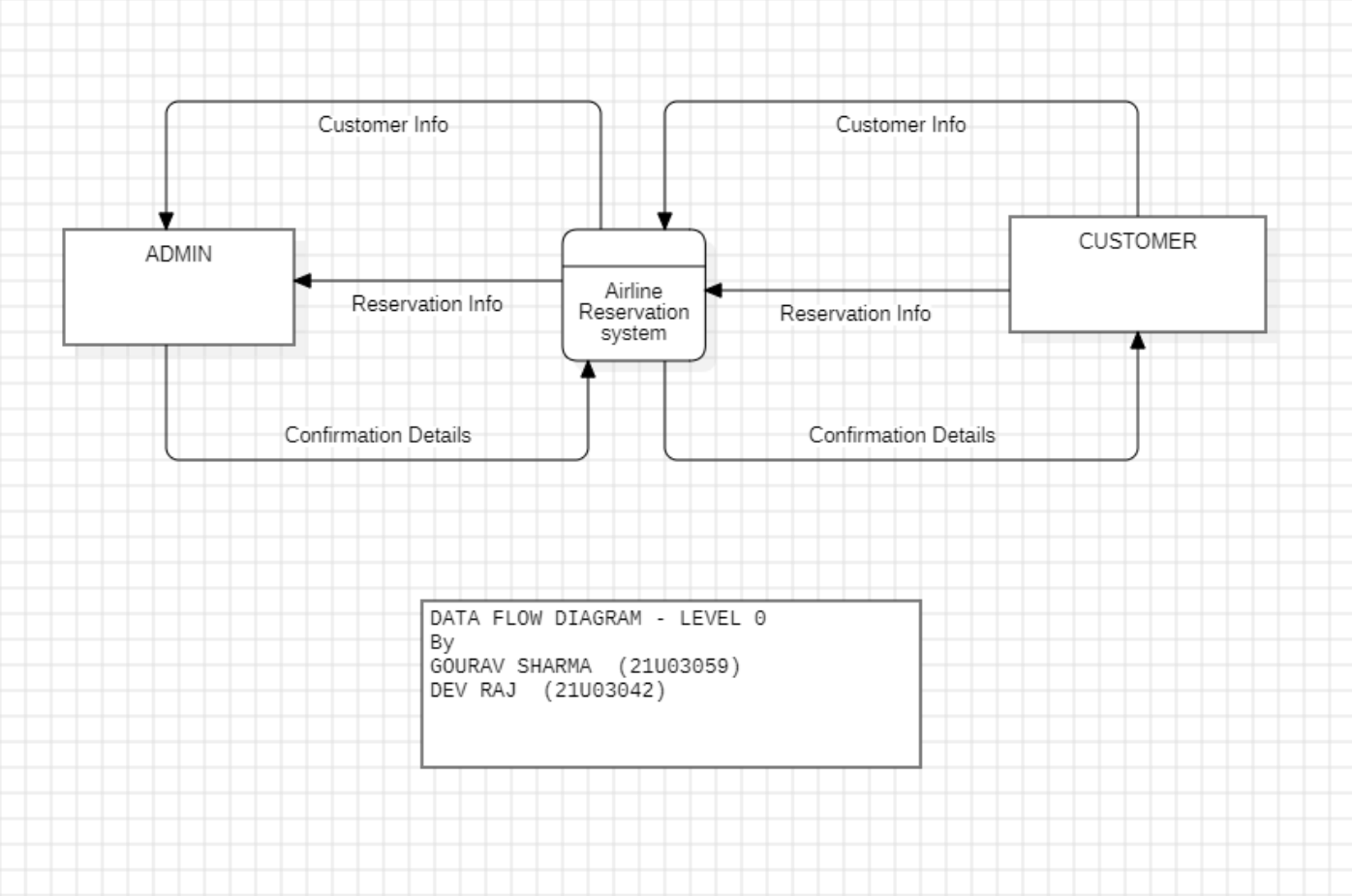
1. **Login:** This activity represents the process of a user, i.e., a passenger, logging into the system. The user must provide valid credentials to access the system.
2. **Search Flights:** After logging in, the user can search for available flights. This activity involves specifying search parameters such as the departure city, destination, date, and flight class.
3. **View Flights:** The system retrieves a list of available flights based on the search criteria. The user can then view a list of flights that match their preferences.
4. **Select Flight:** The user selects a specific flight from the list of available options. This activity signifies the user's choice to proceed with booking a particular flight.
5. **Select Class:** If the flight offers multiple classes (economy, business, etc.), the user may choose a class for the reservation.
6. **Book Flight:** The user confirms their choice, and the system records the booking details. This activity also includes entering passenger information and any special requests.
7. **Payment:** After booking, the user proceeds to the payment activity, where they provide payment details to confirm the reservation.
8. **Confirm Reservation:** The system confirms the reservation and issues a booking confirmation to the user.
9. **End:** The diagram concludes with the reservation process successfully completed.

Data-flow Diagram:

Airline Reservation System DFD level 0

A context diagram is another name for it. With the mechanism represented as a single process and external parties it will be an abstract view. It depicts the overall structure as a single bubble with incoming/outgoing indicators showing input and output data.

The data flow diagram level 0 also considers the entire system to be a single process and emphasizes the system's interaction with external entities. Context diagrams (DFD level 0) are diagrams in which the entire system is represented as a single process.

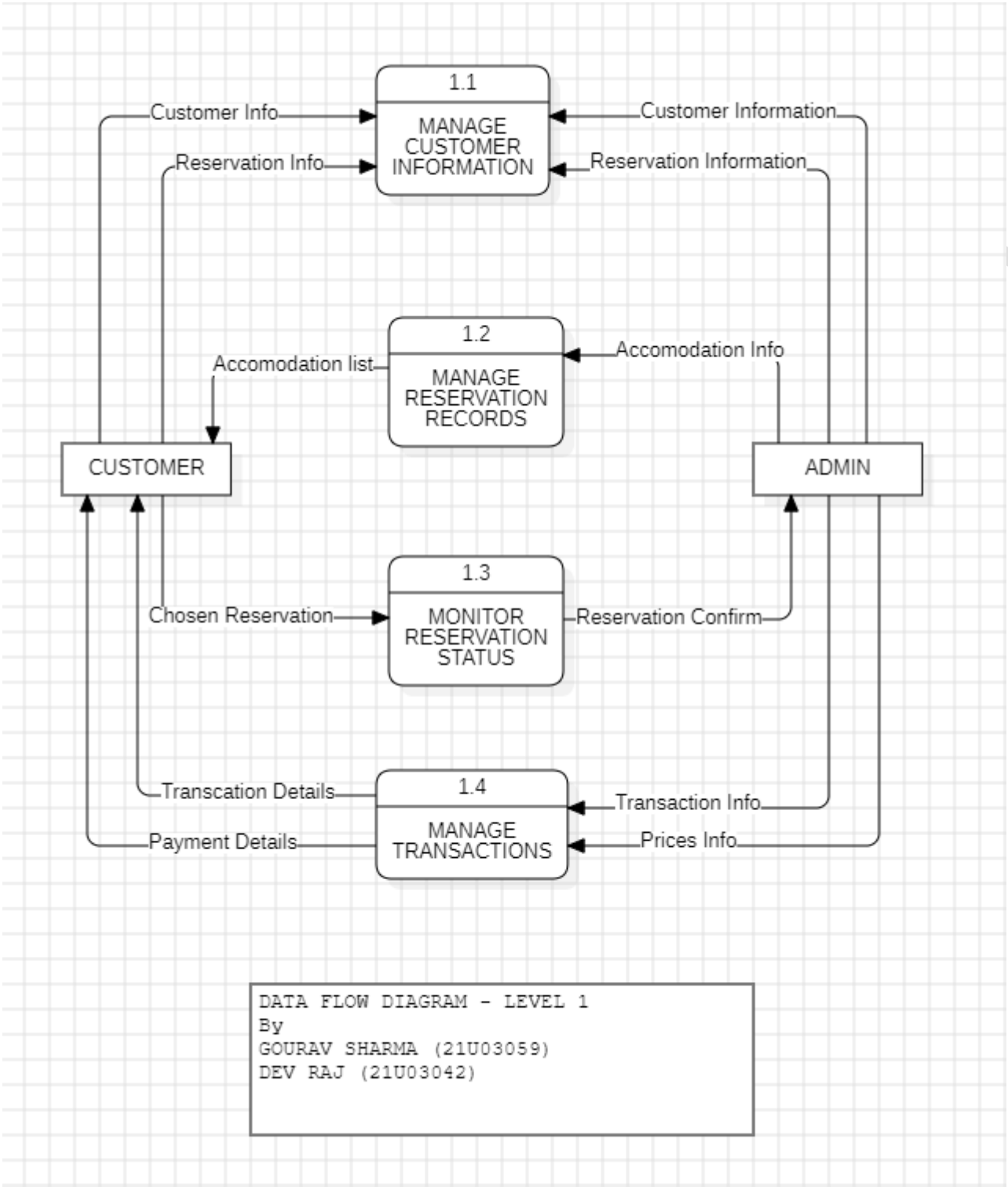


Airline Reservation System DFD level 1

DFD Level 1 provide a broad overview but go into greater depth than a context diagram. The single process node from the context diagram is broken down into sub processes in a level 1 data flow diagram. The System level 1 DFD lists all of the major sub-processes that make up the entire system. A level 1 DFD can be thought of as a "detonated view" of the context diagram.

A single production node and its relationships to external entities are depicted in Level 0 data flow diagrams. Level 1 DFDs provide a broad overview but go into greater depth than a

context diagram. The single process node from the context diagram is broken down into sub processes in a level 1 data flow diagram.



Airline Reservation System DFD level 2

The DFD Level 2 delves even deeper into the concepts of DFD level 1. It can be used to plan or record all of the specific/necessary information about how the system works. DFD Level 2 is the DFD's highest abstraction level and represents the system's basic modules as well as data flow between them.