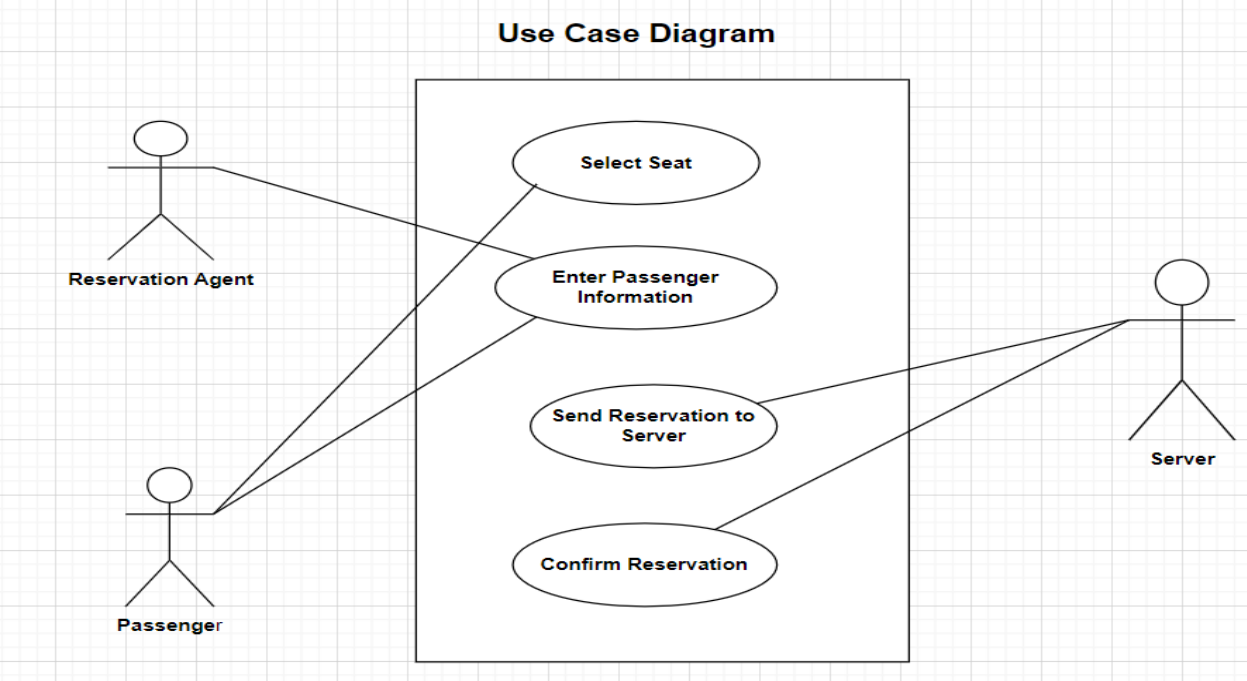
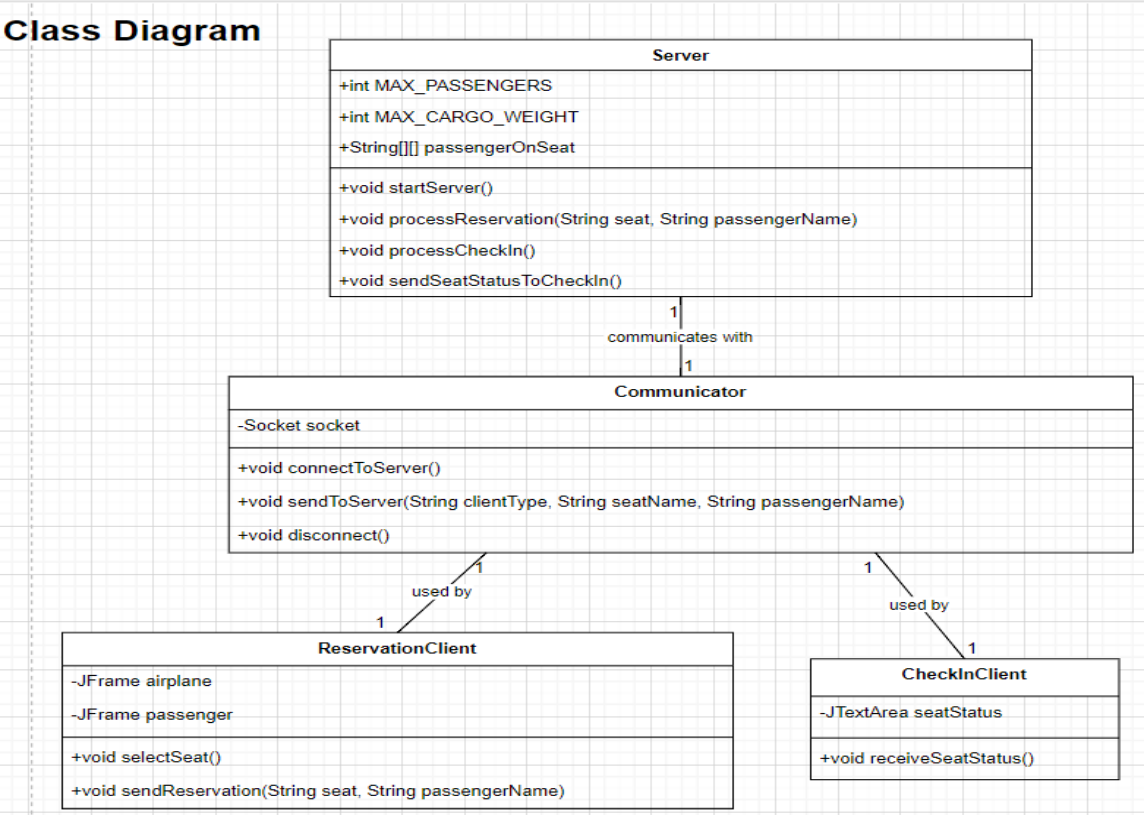
Kalonji Luboya **223234095** **FISA** **14 November 2024**

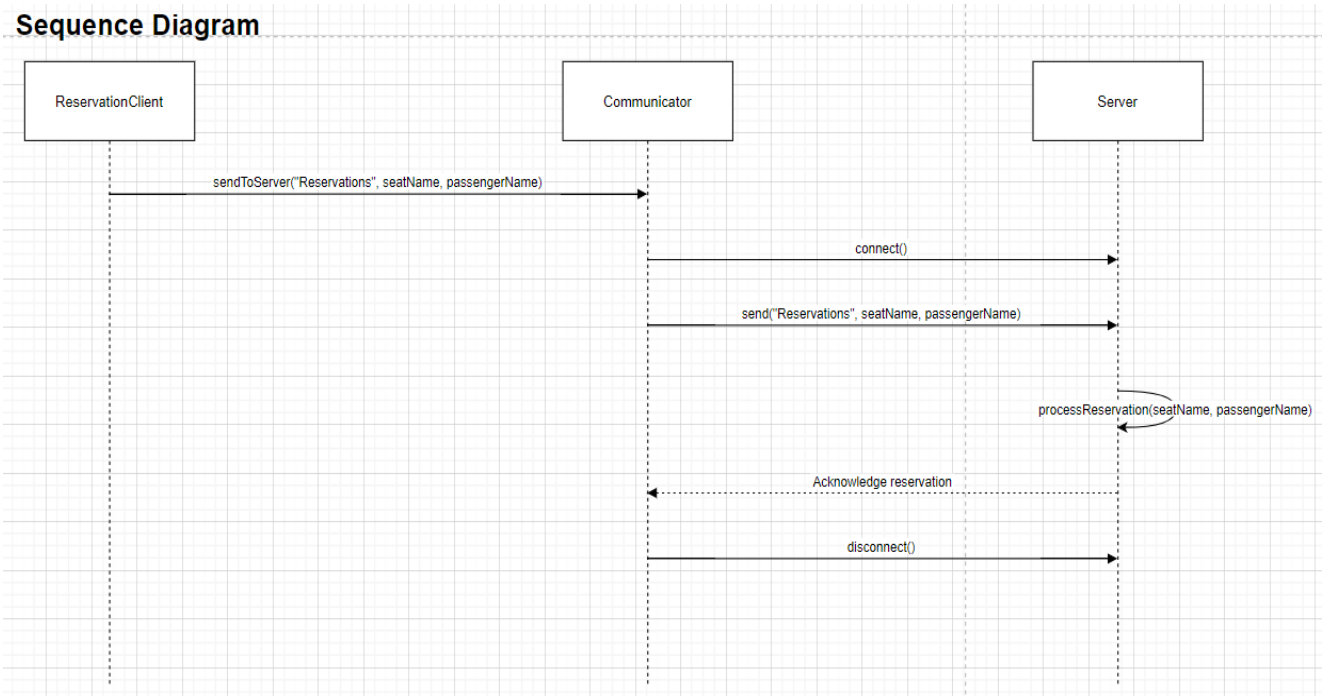
1.Use Case Diagram



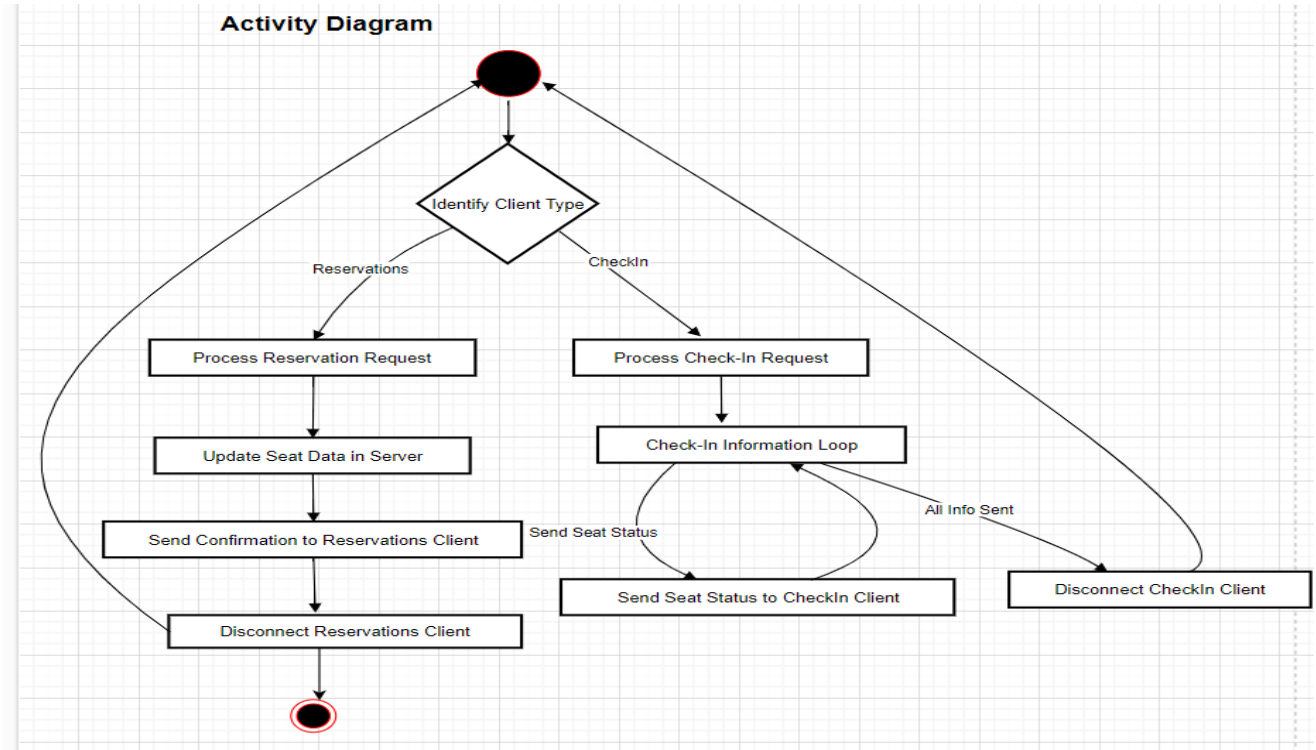
2. Class Diagram



3.Sequence Diagram



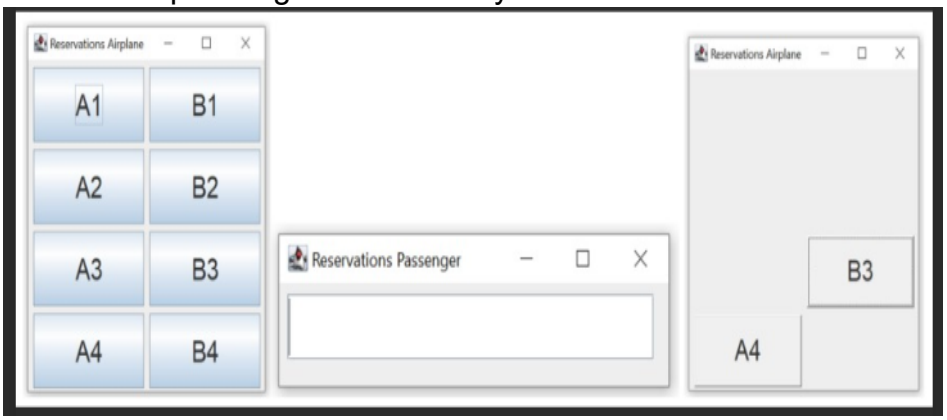
4.Activity Diagram



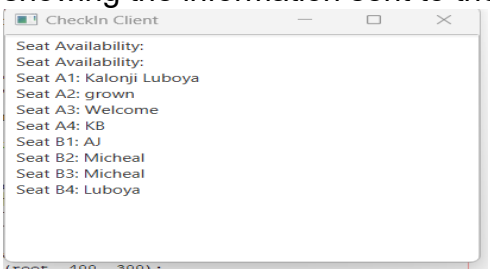
Evidence for the Validation Test

To validate that the system functions as expected, here are some tests that could be performed, along with evidence (observations or screenshots) of each step's correct functionality.

1. Reservation Process Validation:
    - Test: Select a seat from the "Reservations Airplane" GUI, enter a passenger's name in the "Reservations Passenger" window, and submit the reservation.
    - Expected Outcome: The seat button on the main GUI disappears, and the passenger's name is recorded in the server's passengerOnSeat array. The server acknowledges the reservation, and the Communicator disconnects.
    - Evidence: Screenshots or printouts of the GUI showing the removed seat, server logs showing the reservation details, and the updated state of the passengerOnSeat array.



2. Check-in Process Validation:
    - Test: Run the "CheckIn" client, which connects to the server, receives the updated seat assignments, and displays them in the CheckIn GUI.
    - Expected Outcome: The CheckIn client displays a list of reserved seats with passenger names and available seats as null.
    - Evidence: A screenshot of the CheckIn client's GUI showing all seats and corresponding names or null values, as well as server logs showing the information sent to the CheckIn client.



3. Multiple Connections Test:
    - Test: Ensure that only one client (either ReservationClient or CheckInClient) can connect to the server at a time.
    - Expected Outcome: If one client is connected, a second client attempting to connect should not establish a connection until the first client disconnects.
    - Evidence: Server logs showing successful connections and disconnections, and messages indicating when an attempted connection is refused because another client is connected.
4. Server's Bidirectional Communication:
    - Test: Connect the CheckInClient to the server and ensure the server sends updated seat information multiple times.
    - Expected Outcome: The CheckInClient should receive the initial seat assignments and any updates when a reservation is made.
    - Evidence: Server logs showing each communication with the CheckInClient, and the updated seat status displayed in the CheckInClient GUI each time a reservation is processed.

Explanation of Validation Test

A validation test is designed to confirm that the system behaves as intended in real-world scenarios. In this case, the validation tests are meant to verify:

1.Correct Functionality:

- Ensuring that reservations and check-ins can be performed as specified.

2.Data Integrity:

- Confirming that the server's passengerOnSeat array is accurately updated and that information displayed to each client is correct.

3.Client-Server Communication:

- Testing bidirectional communication and enforcing a single client connection at a time to match the requirements.

4.Error Handling:

- Making sure that only one client can connect at a time and that attempts to connect multiple clients are gracefully handled.