**DESIGN OF AIRLINE MANAGEMENT SYSTEM**

5.1.GUI

5.2. Static model – class diagram

A screenshot of a computer

Description automatically generated

5.3.Dynamic model – sequence diagram

A diagram with blue squares and black text

Description automatically generated

A diagram of a diagram

Description automatically generated

5.4.Rationale of the detailed designs above

Static Class Diagram Rationale:

-Passenger Class:

* Represents passengers in the system.
* Contains attributes such as passengerID, name, and contactInfo to store passenger information.
* Has a one-to-many relationship with reservations, allowing passengers to have multiple reservations.

-Administrator Class:

* Represents system administrators.
* Contains attributes such as adminID, username, and password to manage administrator accounts.
* Includes operations for managing passengers and flights, as administrators need to have control over these aspects.

-Flight Class:

* Represents flight information.
* Contains attributes such as flightID, airline, origin, destination, and schedule details.
* Has a one-to-many relationship with reservations, allowing multiple reservations for a single flight.

-Reservation Class:

* Represents passenger reservations.
* Contains attributes like reservationID, passenger, flight, status, and bookingTime.
* Relates passengers to their reservations and flight bookings.

Sequence diagram Rationale:

User:

* Users provide their information to log in.
* The login page communicates user-entered information to the verification page.
* The verification page verifies the information from database page.
* Verification page communicates the validation status to the login page.
* User interaction with the authenticated system.

Admin:

* Admins provide their information to log in.
* Admins can change/edit flights, passengers, tickets and reservations.

5.5. Traceability from requirements to detailed design model

1.Requirement Identification:

* Identify and list the key requirements outlined in the Airline Management System description provided.
* Clearly specify the user and system inputs, operations, and expected outputs.

2. Administrator Module Design:

* Based on the administrator operations outlined in the requirements (Login, Manage passengers, Update passenger's status), design the module structure.
* Specify how the administrator login functionality is implemented in the detailed design.

3. Passenger Module Design:

* For passenger operations (Login, Edit Profile, View available flights, Reservation, Ticket cancellation), design the module structure.
* Provide details on how the passenger login and profile editing are implemented in the detailed design.

4. Flight Information Module Design:

* Design the module for handling Flight Information based on the provided requirements.
* Clearly specify how both Domestic and International flights are represented and managed in the detailed design.

5. Input/Output Handling:

* Detail how user inputs (Passenger and Administrator information, Flight details) are processed in the system.
* Specify the mechanisms for displaying available flights, confirming reservations, and printing tickets.

6. Error Handling:

* Describe how errors or exceptional scenarios, as mentioned in the requirements, are handled in the detailed design.
* Ensure that error messages are appropriately generated and displayed to users.

7. Documentation:

* Include appropriate comments, annotations, or documentation within the design model to explain the rationale behind design decisions and how they fulfill the specified requirements.