OOAD

SKY-FLEET

AVIATION CONTROL

TEAM :

MADHUSUDANA M	:	PES1UG22CS320
M ANIRUDDHA SESHU	:	PES1UG22CS309
MANISH P K	:	PES1UG22CS332
PRIYAM ROY	:	PES1UG22CS453

PROBLEM STATEMENT

Efficient flight scheduling and management is crucial for airlines to optimise operations, minimise delays, and enhance passenger experience. However, airlines often face challenges such as inaccurate scheduling, inefficient aircraft allocation, and poor real-time monitoring, leading to delays, resource wastage, and customer dissatisfaction. This system aims to provide a **smart, automated, and adaptive** solution for **flight scheduling, aircraft assignment, and real-time flight monitoring** to streamline airline operations and improve efficiency

SYNOPSIS

INTRODUCTION

Modern airlines handle a vast number of flights daily, requiring precise scheduling, aircraft allocation, and real-time status tracking. Traditional methods often involve **manual coordination, lack of optimisation, and reactive management**, leading to inefficiencies. **SkyFleet** aims to address these challenges by providing an intelligent, automated system that **dynamically schedules flights, assigns aircraft efficiently, and continuously monitors flight status** to improve operational efficiency and passenger experience.

OBJECTIVES

Automate Flight Scheduling: Enable airline staff to create, update, and cancel flight schedules efficiently.
Optimise Aircraft Assignment: Assign available aircraft dynamically based on capacity, maintenance status, and demand.
Real-time Flight Monitoring: Track departures, arrivals, and delays with automated alerts.

Minimise Operational Delays: Reduce turnaround time by ensuring smooth coordination between flights, gates, and runways.
Enhance Passenger & Crew Communication: Provide real-time notifications for flight changes, cancellations, and gate assignments.

In conclusion ,Flight scheduling is the **backbone of airline operations**. An intelligent system like **SkyFleet** ensures that flights are **planned, monitored, and adjusted in real time**, minimising delays and disruptions while maximising efficiency and customer satisfaction.

USE CASES

MAJOR USE CASES

1. FLIGHT SCHEDULING & MANAGEMENT

Actors: Airline Staff, System

Description:

- Create, update, and cancel flight schedules.
- Assign aircraft to scheduled flights.
- Monitor real-time flight status.

2. PASSENGER & TICKET MANAGEMENT

Actors: Passenger, Airline Staff, System

Description:

- Passengers book, modify, or cancel tickets.
- The system manages seat availability and payments.
- Generates boarding passes and updates check-in status.

3. GATE & RUNWAY ASSIGNMENT

Actors: airline admin,Pilot, System

Description:

- Assigns gates for boarding and de-boarding.
- Assigns runways for takeoff and landing.
- Ensures proper sequencing of flights.

4. CREW & STAFF MANAGEMENT

Actors: Crew Manager, Pilot, Cabin Crew, System

Description:

- Assigns pilots and cabin crew to flights.
- Tracks working hours and schedules.
- Ensures compliance with regulations.

Minor use cases

BAGGAGE TRACKING & HANDLING

- Tracks passenger baggage from check-in to arrival.
- Alerts passengers if baggage is delayed or misplaced.

FLIGHT DELAY & NOTIFICATION SYSTEM

- Sends real-time flight status updates to passengers & crew.
- Notifies gate changes, delays, and cancellations.

AIRCRAFT MAINTENANCE & SAFETY CHECKS

- Schedules routine aircraft maintenance.
- Logs and tracks safety inspections before flights.
- Alerts staff if an aircraft is unfit for service.

PASSENGER FEEDBACK & COMPLAINT SYSTEM

- Allows passengers to submit feedback or complaints.
- Tracks and resolves customer issues efficiently.
- Helps improve airline services based on user input.

CLASSES

1. PERSON (ABSTRACT CLASS)

Represents any individual interacting with the system.

Attributes:

- personID: Unique identifier.
- name: Full name.
- contactInfo: Contact details.

Methods:

- getDetails(): Returns basic details of the person (name, ID, contact).

Inherited By: Passenger, CrewMember

2. PASSENGER (INHERITS FROM PERSON)

Stores passenger details and booking information.

Attributes:

- passengerID: Unique identifier for the passenger.
- passportNumber: Passport/ID number.
- ticket: Ticket associated with the flight.

Methods:

- bookFlight(Flight flight): Books a flight.
- cancelBooking(): Cancels a booking.
- getBoardingPass(): Generates a boarding pass.

3. CREWMEMBER (INHERITS FROM PERSON)

Stores details of crew members like pilots and attendants.

Attributes:

- crewID: Unique identifier.
- role: Crew role (Pilot, Co-Pilot, Cabin Crew, etc.).
- assignedFlight: The flight they are currently assigned to.

Methods:

- assignFlight(Flight flight): Assigns the crew member to a flight.
- removeFromFlight(): Unassigns the crew member.

4. AIRPORTRESOURCE (ABSTRACT CLASS)

Represents shared airport infrastructure like gates & runways.

Attributes:

- resourceID: Unique identifier.
- availability: True/False based on flight assignment.

Methods:

- isAvailable(): Returns if the resource is free.
- assignFlight(Flight flight): Assigns a flight to the resource.
- freeResource(): Marks resource as available.

Inherited By: Gate, Runway

5. GATE (INHERITS FROM AIRPORTRESOURCE)

Handles gate allocation for flights.

Attributes:

- gateNumber: Unique identifier for the gate.
- assignedFlight: The flight currently using the gate.

Methods:

- assignFlight(Flight flight): Assigns a flight to this gate.
- freeGate(): Marks gate as available.

6. RUNWAY (INHERITS FROM AIRPORTRESOURCE)

Manages runways for takeoff and landing.

Attributes:

- runwayID: Unique identifier for the runway.
- assignedFlight: The flight currently using the runway.

Methods:

- assignFlight(Flight flight): Assigns a flight to the runway.
- freeRunway(): Marks runway as available.

7. AIRCRAFT

Represents an aircraft and its capabilities.

Attributes:

- aircraftID: Unique identifier.
- model: Aircraft model (Boeing 737, Airbus A320, etc.).
- capacity: Passenger capacity.
- status: Available, In Maintenance, Assigned.

Methods:

- assignFlight(Flight flight): Assigns the aircraft to a flight.
- markAsMaintenance(): Marks aircraft as under maintenance.

8. FLIGHT

Represents a scheduled flight .

Attributes:

- flightID: Unique identifier for the flight.
- airline: Airline operating the flight.
- origin: Departure airport.
- destination: Arrival airport.
- departureTime: Scheduled departure time.
- arrivalTime: Scheduled arrival time.
- status: Flight status (Scheduled, Boarding, Departed, Landed, Canceled).
- assignedGate: Gate assigned for boarding.
- assignedRunway: Runway assigned for takeoff/landing.
- aircraft: Assigned aircraft for the flight.
- crewList: List of assigned crew members.

- passengerList: List of booked passengers.

Methods:

- assignGate(Gate gate): Assigns an available gate.
- assignRunway(Runway runway): Assigns an available runway.
- assignAircraft(Aircraft aircraft): Assigns an aircraft to the flight.
- updateStatus(String status): Updates flight status.
- addPassenger(Passenger passenger): Adds a passenger to the flight.
- assignCrew(CrewMember crew): Assigns crew members.

9. TICKET

Represents a flight ticket, linking to the assigned passenger and flight.

Attributes:

- ticketID: Unique ticket number.
- flight: Associated flight.
- passenger: Associated passenger.
- seatNumber: Assigned seat.
- ticketStatus: Active, Canceled, Expired.

Methods:

- validateTicket(): Checks if the ticket is valid.
- updateStatus(String status): Updates ticket status.

10. AIRPORTSYSTEM

Main system responsible for managing flights, gates, and operations.

Attributes:

- flights: List of all flights.
- gates: List of gates.
- runways: List of runways.
- aircrafts: List of aircraft.
- crewMembers: List of available crew members.

Methods:

- assignGateToFlight(Flight flight): Finds an available gate and assigns it.
- assignRunwayToFlight(Flight flight): Finds an available runway and assigns it.
- scheduleFlight(Flight flight): Adds a flight to the system.
- cancelFlight(Flight flight): Cancels a flight

11. BAGGAGE

Represents a passenger's baggage in the system.

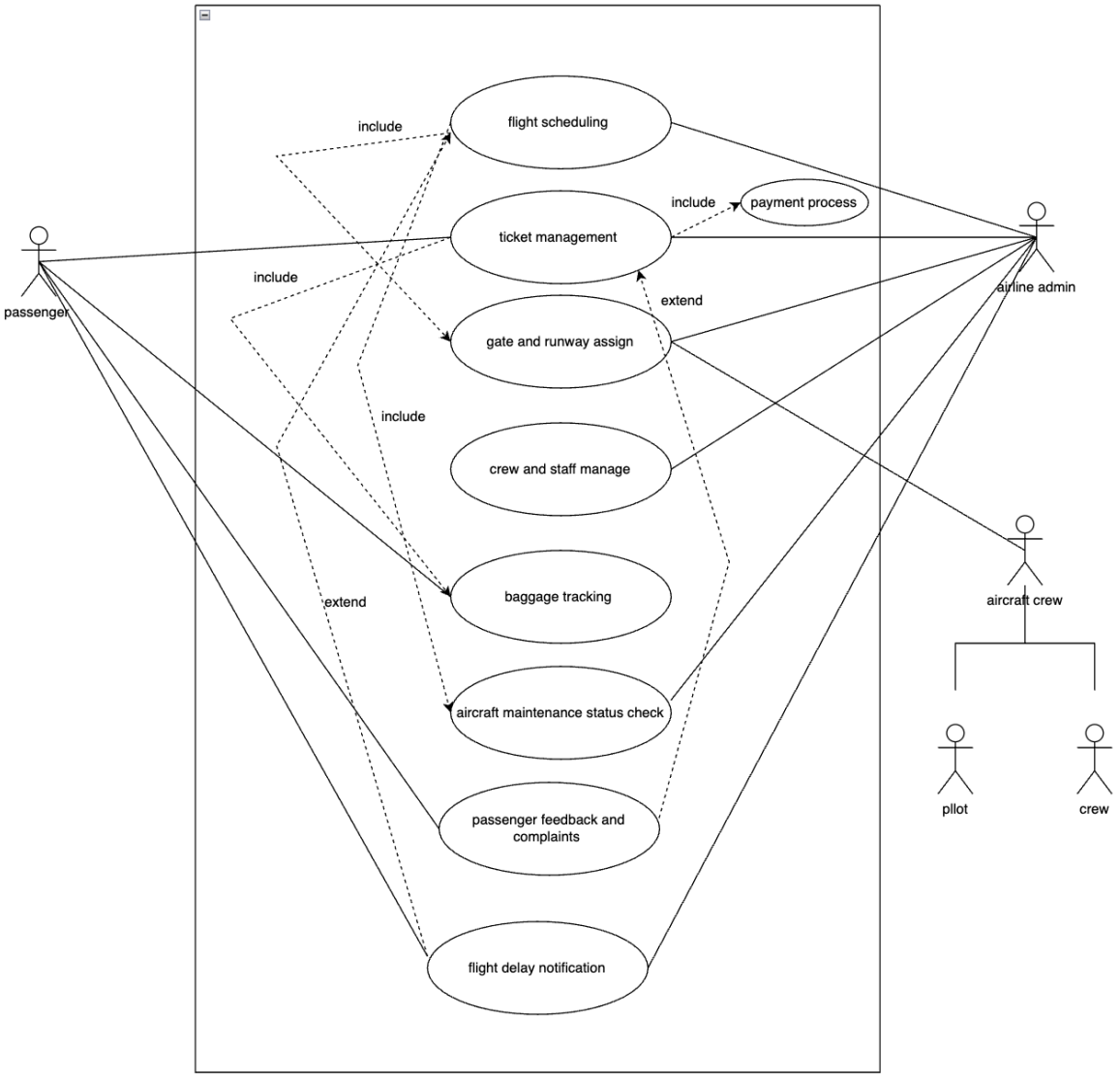
Attributes:

- baggageID: Unique identifier.
- owner: Associated passenger.
- weight: Weight of the baggage.
- status: Checked-In, Loaded, In Transit, Delivered.

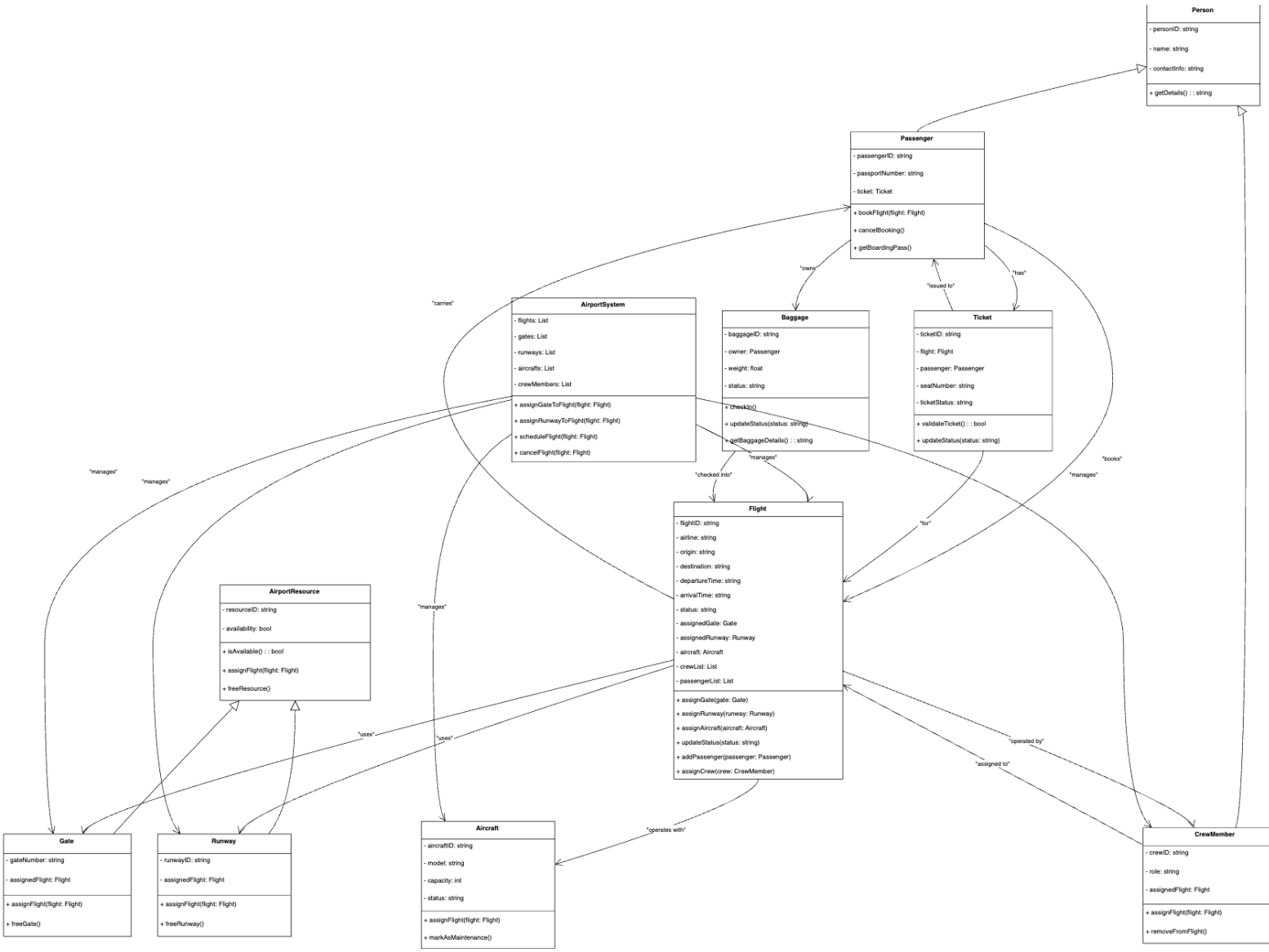
Methods:

- checkIn(): Marks baggage as checked in.
- updateStatus(String status): Updates baggage status.
- getBaggageDetails(): Returns baggage information.

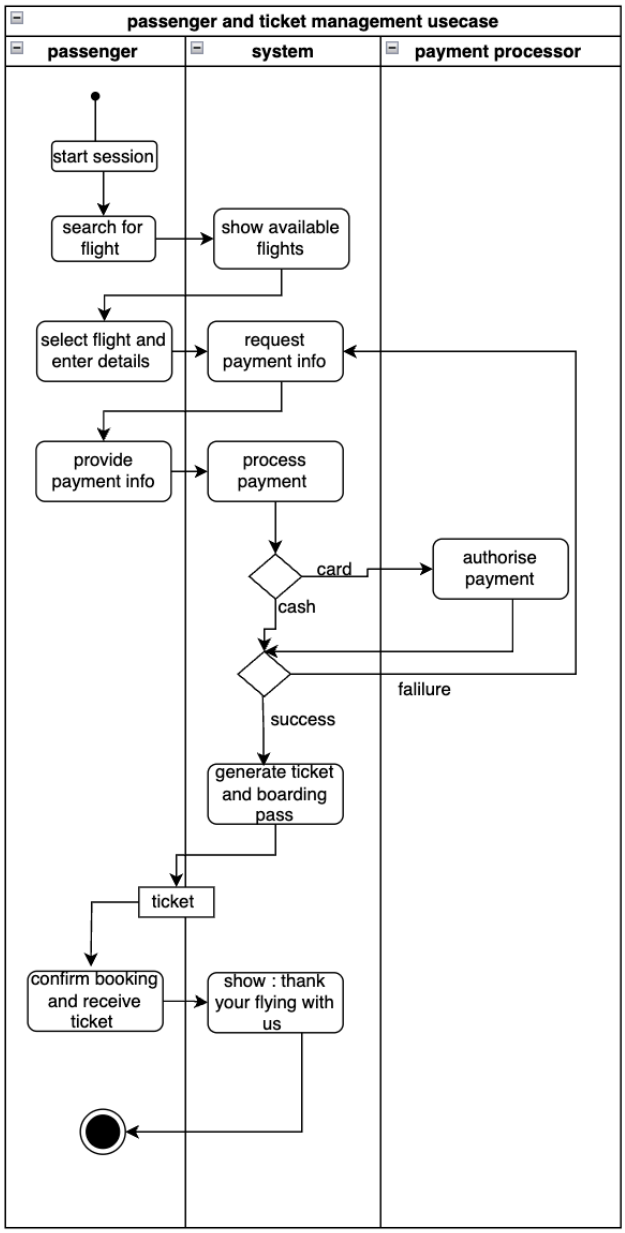
USE CASE DIAGRAM

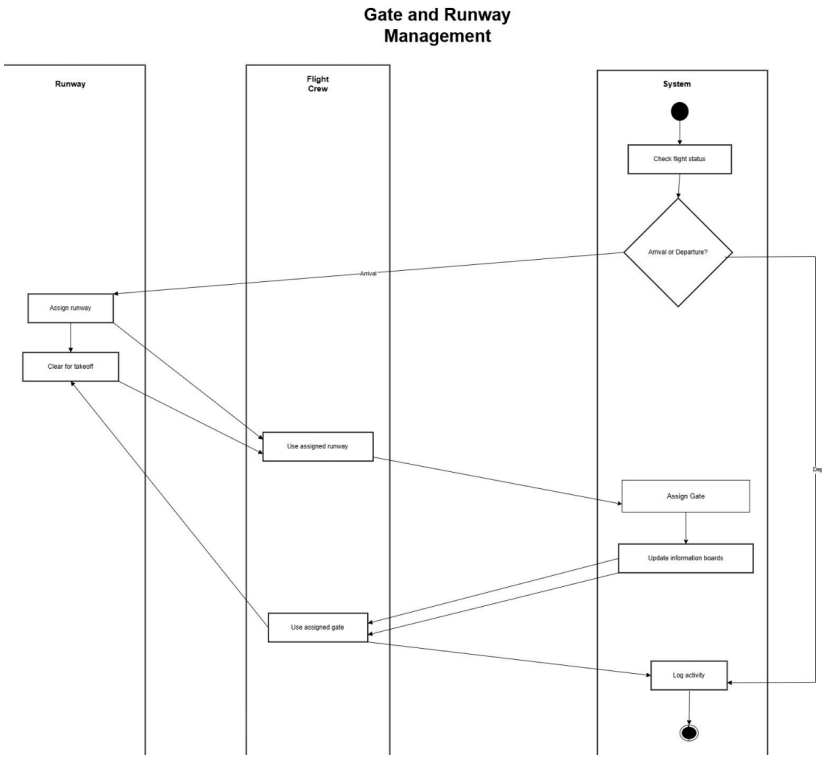


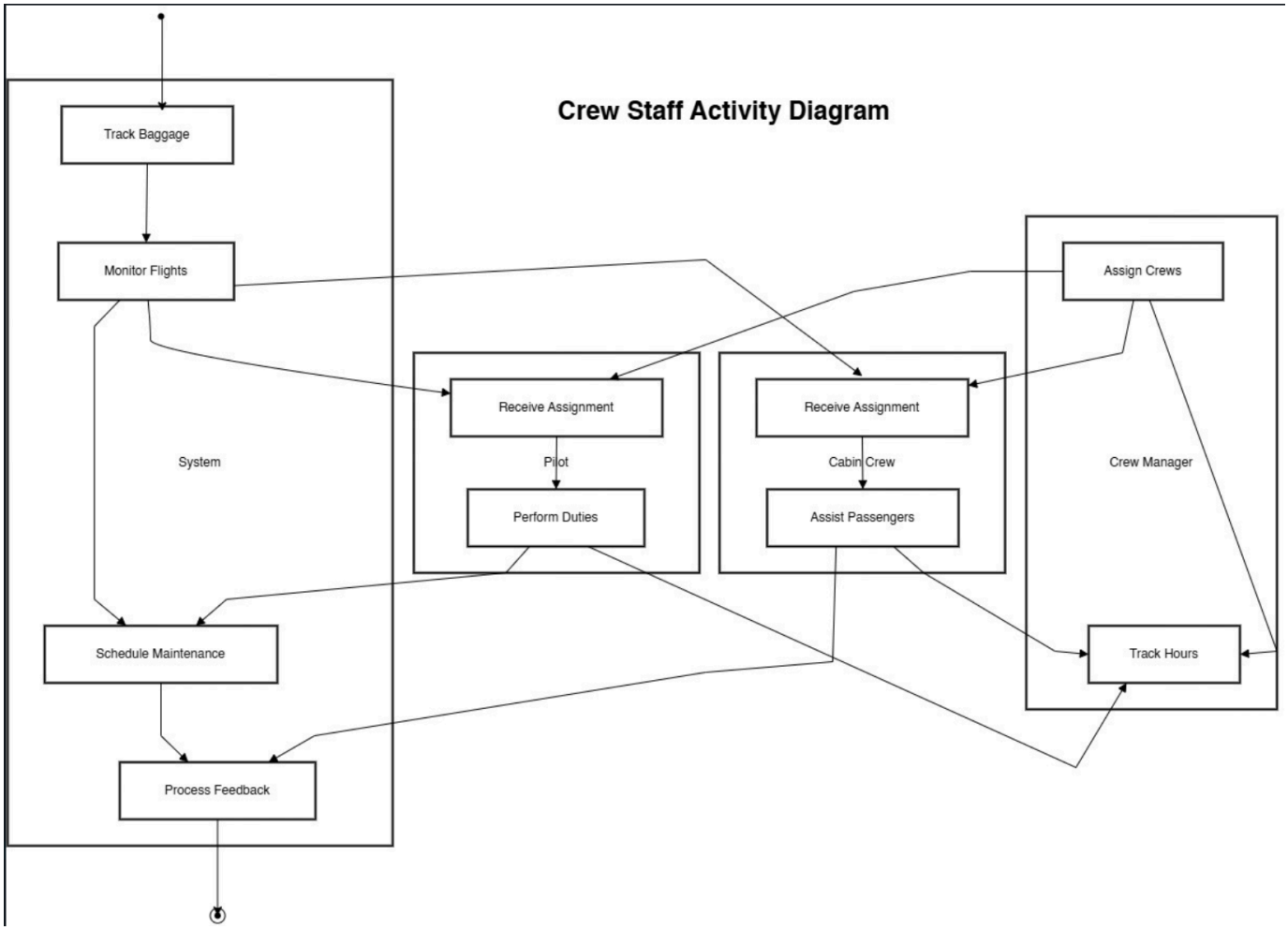
CLASS DIAGRAM

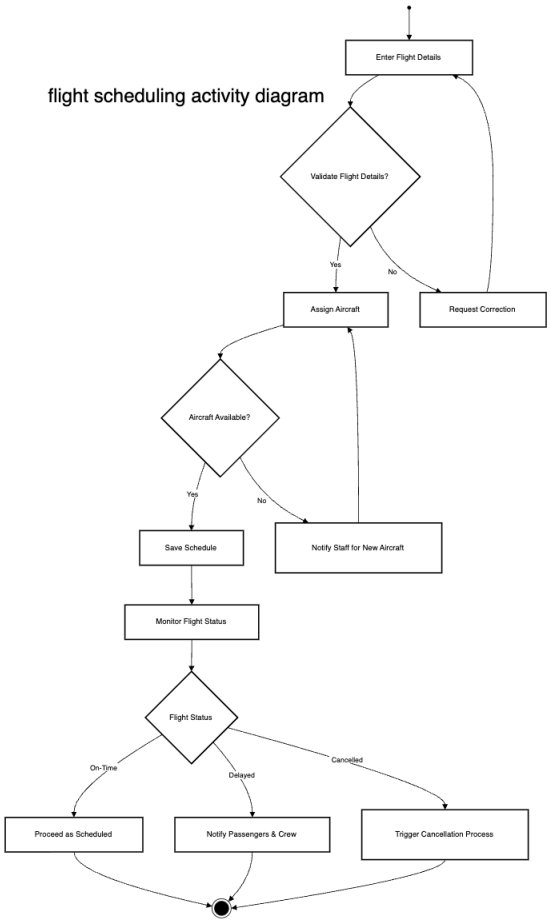


ACTIVITY DIAGRAMS

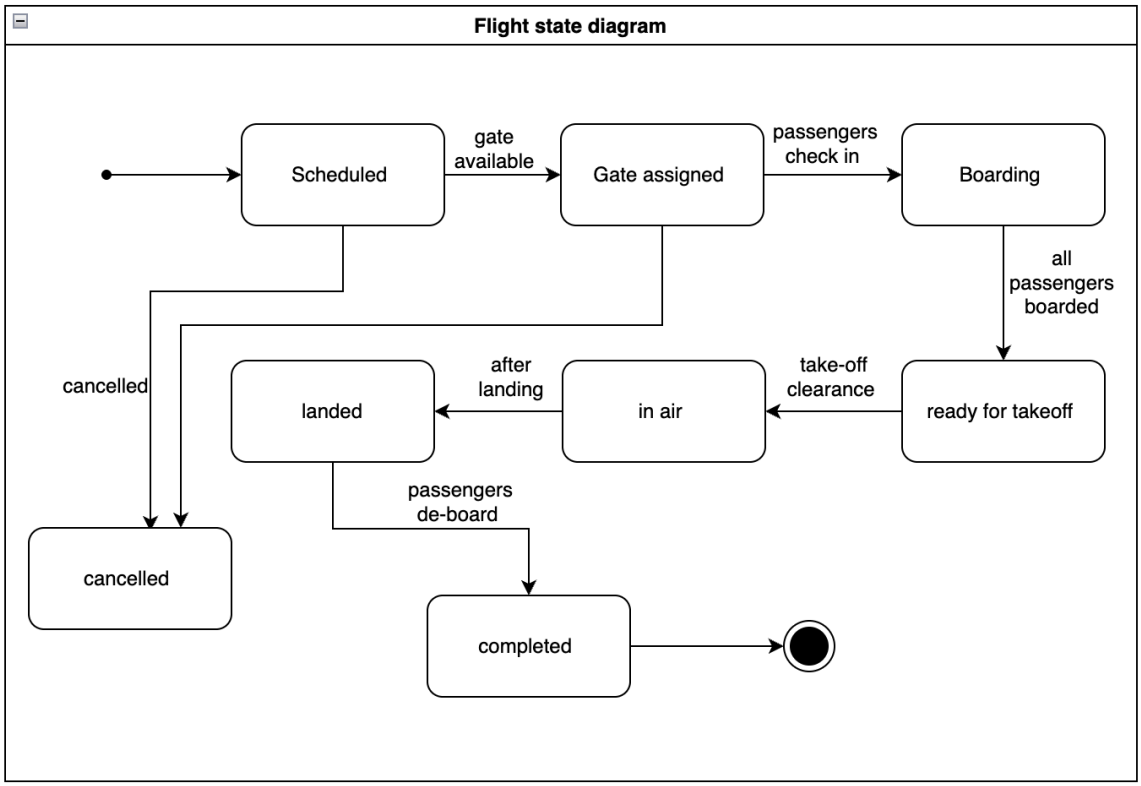


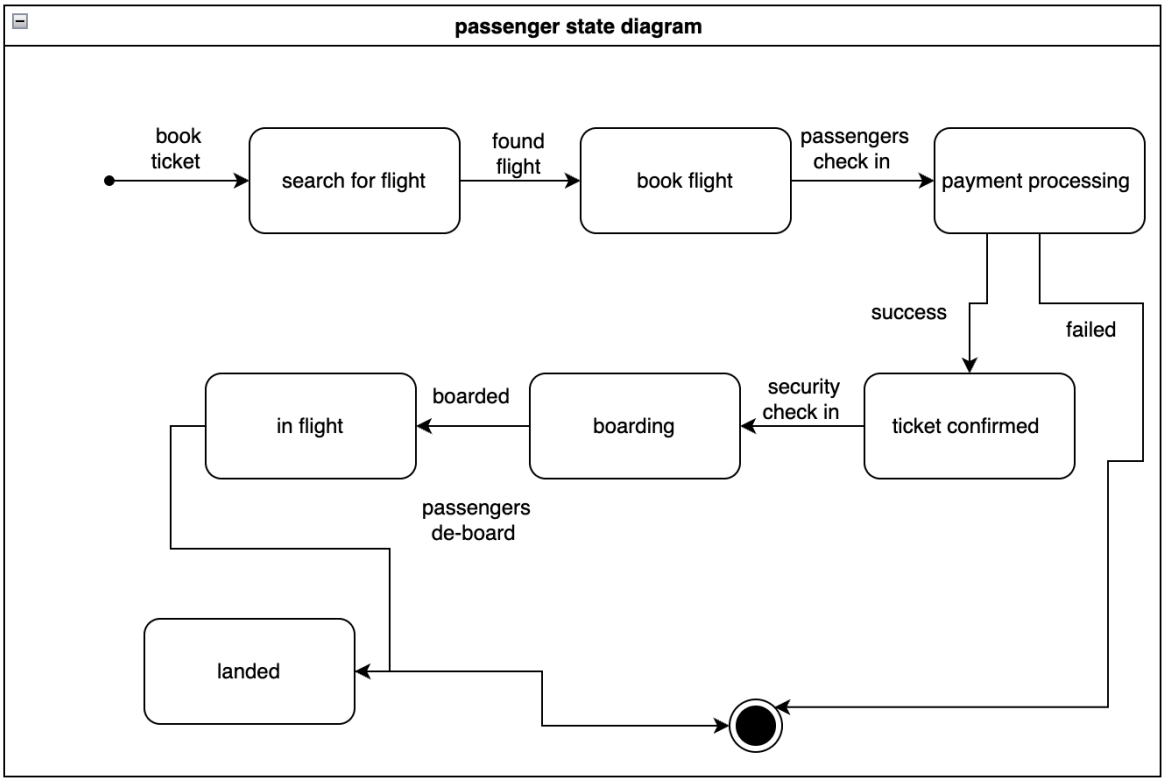


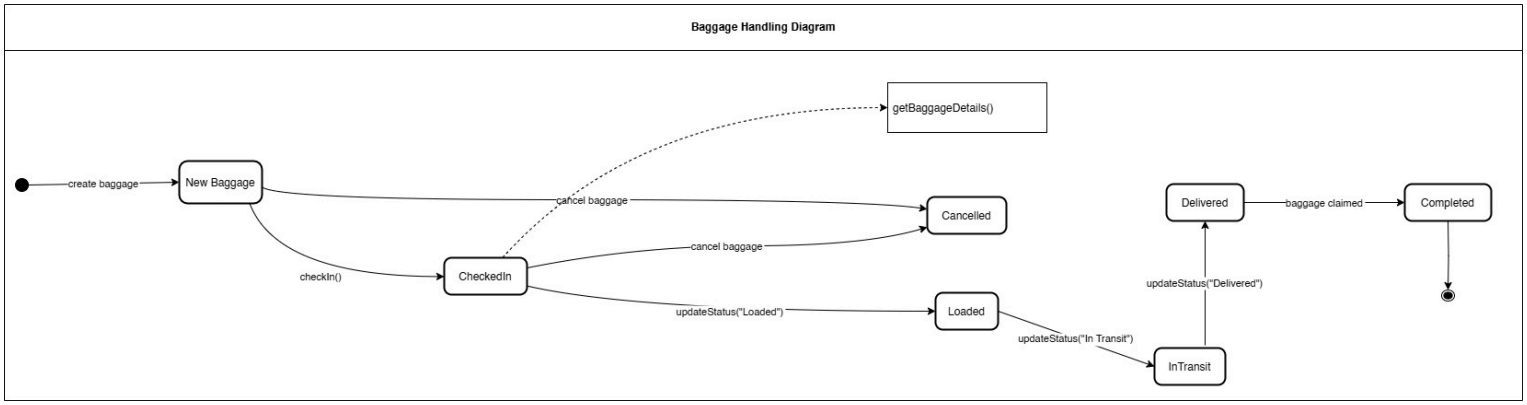

Crew Staff Activity Diagram
Track Baggage
Monitor Flights
System
Receive Assignment
Pilot
Perform Duties
Receive Assignment
Cabin Crew
Assist Passengers
Assign Crews
Crew Manager
Track Hours
Schedule Maintenance
Process Feedback

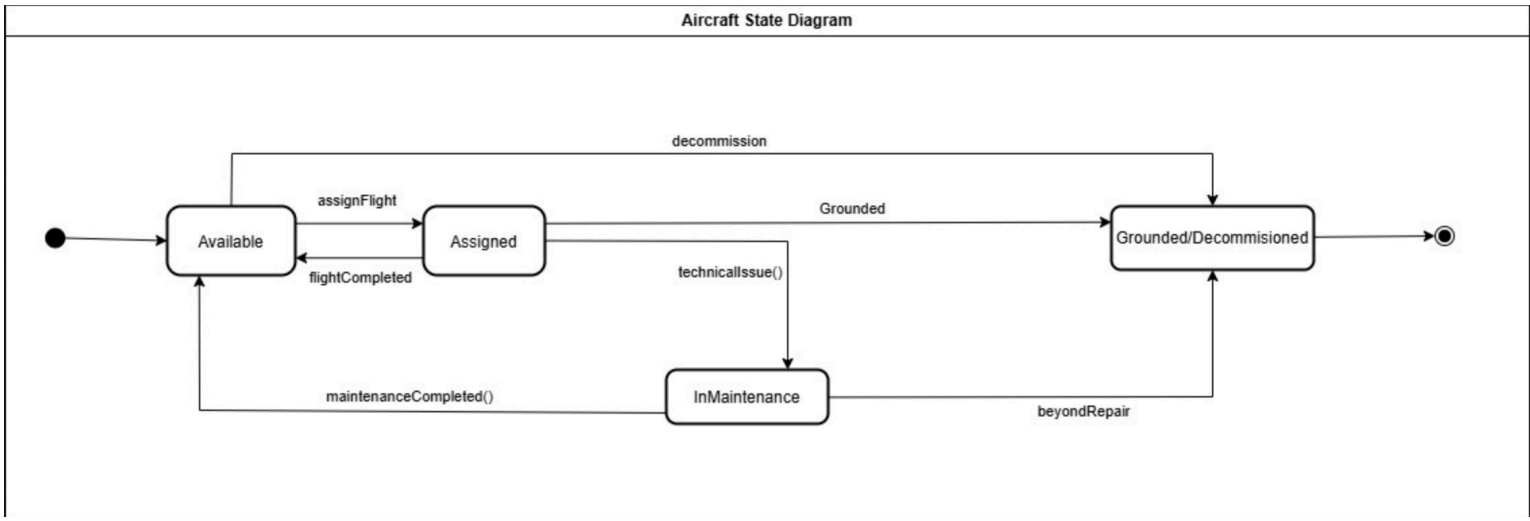

flight scheduling activity diagram
Enter Flight Details
Validate Flight Details?
Yes
No
Assign Aircraft
Request Correction
Aircraft Available?
Yes
No
Save Schedule
Notify Staff for New Aircraft
Monitor Flight Status
Flight Status
On-Time
Delayed
Cancelled
Proceed as Scheduled
Notify Passengers & Crew
Trigger Cancellation Process

STATE DIAGRAMS






Baggage Handling Diagram
getBaggageDetails()
create baggage
New Baggage
cancel baggage
Cancelled
checkIn()
CheckedIn
cancel baggage
updateStatus("Loaded")
Loaded
updateStatus("In Transit")
InTransit
updateStatus("Delivered")
Delivered
baggage claimed
Completed


Aircraft State Diagram
decommission
assignFlight
Available
Assigned
flightCompleted
Grounded
Grounded/Decommisioned
technicalIssue()
maintenanceCompleted()
InMaintenance
beyondRepair