AIRLINE MANAGEMENT SYSTEM

A PROJECT REPORT

for

OBJECT ORIENTED ANALYSIS AND DESIGN (SWE2018)

in

M. Tech (Software Engineering)

by

NITHISH KUMAR (20MIS0024)

JAYASREE N (20MIS0370)

DEEPTHI I (20MIS0387)

7th Semester, 4th Year

Under the Guidance of

Prof. KIRUBA THANGAM R

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

School of Information Technology and Engineering

PROBLEM ANALYSIS:

An Airline Management System is a managerial software which targets to control all operations of an airline. Airlines provide transport services for their passengers. They carry or hire aircraft for this purpose. All operations of an airline company are controlled by the airline management system. This system involves the scheduling of flights, air ticket reservations, flight cancellations, customer support, and staff management. In addition to these functionalities, this system provides services to book hotels, restaurants and travel facilities as per the request of the passenger. Daily flights updates can also be retrieved by using the system.

Passengers should be able to search for flights for a given date and source or destination airport. Passengers should be able to reserve a ticket for any scheduled flight. Passengers can also build a multi-flight itinerary. Users of the system can check flight schedules, their departure time, available seats, arrival time, and other flight details. Passengers can make reservations for multiple passengers under one itinerary. Only the admin of the system can add new aircrafts, flights, and flight schedules. Admin can cancel any pre-scheduled flight (all stakeholders will be notified). Passengers can cancel their reservation and itinerary. The system should be able to handle the assignment of pilots and crew members to flights. The system should be able to take the leave request of pilots and crew members. The system should be able to handle payments for reservations. The system should be able to send notifications to passengers whenever a reservation is made / modified or there is an update for their flights. The system should be able to provide with hotels, restaurants and travel suggestions. Passengers should be able to provide feedbacks about the airline as per their convenience. Passengers should be able to get in touch with customer service as per their requirement.

EXISTING SYSTEM:

- Passengers should be able to search for flights for a given date and source / destination airport.
- Passengers should be able to reserve a ticket for any scheduled flight.
- Users of the system can check flight schedules, their departure time, available seats, arrival time, and other flight details.
- The system should be able to handle payments for reservations.
- Passengers can also build a multi-flight itinerary.

PROPOSED SYSTEM:

- The system should be able to handle the assignment of pilots and crew members to flights.
- The system should be able to take the leave request of pilots and crew members.
- The system should be able to send notifications to passengers whenever a reservation is made / modified or there is an update for their flights.
- The system should be able to provide with hotels, restaurants and travel suggestions.
- The system should be able to provide information about the hot spots of the destination.
- Passengers should be able to provide feedbacks about the airline as per their convenience.
- Passengers should be able to get in touch with customer service as per their requirement.
- Only the admin of the system can add new aircrafts, flights, and flight schedules.

STAKEHOLDERS:

ACTOR STAKEHOLDERS	NON ACTOR STAKEHOLDERS
Flight Passengers	Travel Retailers
Pilot & Crew Members	Hotel Management
Airline Officials	Restaurant Management
Customer Service	UPI Service Providers
Admin	Data Organizers
Maintenance Team	Network Service Providers
	Project Managers
	Client of the Project

FEATURES:

- Authenticate the user into a system.
- Registering the passenger into a system.
- Booking the flight by a passenger.
- Ordering the snacks by the passenger.
- Modifying the ticket by the passenger.
- Providing the review by the passenger.
- Seeking for assistance by the passenger.
- Managing the profile of the passenger.
- Assisting with travel choices for the passenger.
- Utilizing the additional functionalities by the passenger.
- Checking the flight status by the user.
- Managing of flights by the admin.
- Assigning the schedule by the admin.
- Onboarding the employees by the admin.

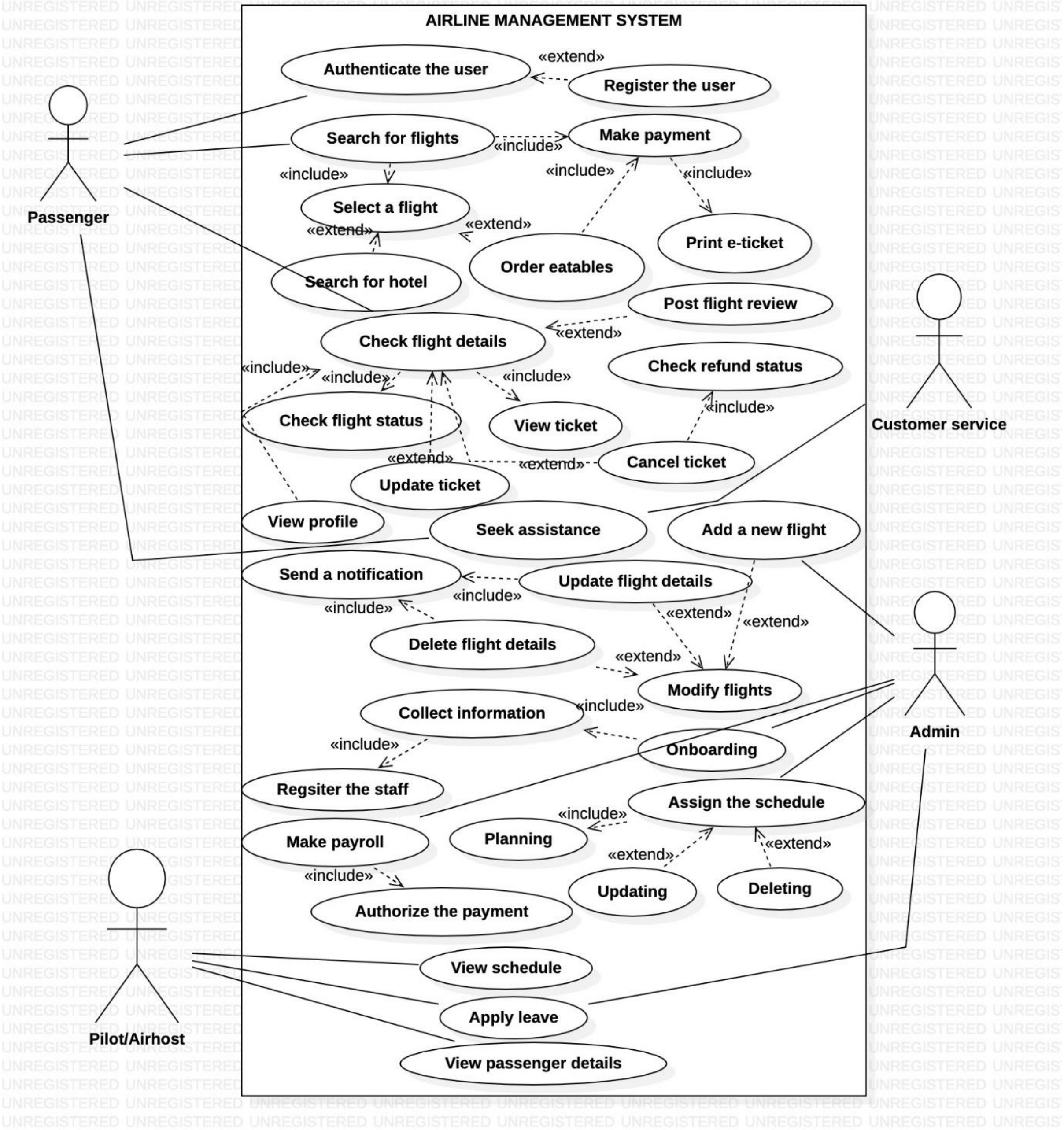
- Processing the payroll by admin.
- Checking the schedule by the flight crew.
- Applying the leave by the employees.
- Assisting the passengers by customer support.
- Sending the notification to the user.
- Logging out by the user.

USECASES:

- Authenticate the passenger into a system.
- Registering the user into a system.
- Searching for flights by a passenger.
- Selecting the flight by a passenger.
- Ordering the snacks by the passenger.
- Finalizing the booking by making a payment.
- Updating the flight ticket by the passenger.
- Cancelling the flight ticket by the passenger.
- Providing the review by the passenger.
- Seeking for assistance by the passenger.
- View the profile of the passenger.
- Modify the profile of the passenger.
- Assisting with travel choices for the passenger.
- Viewing the accommodation by the passenger.
- Viewing the transportation by the passenger.
- Viewing the restaurants by the passenger.
- Checking the flight status by the user.
- Registering the flights by the admin.
- Updating the flights by the admin

- Deleting the flights by the admin
- Assigning the schedule by the admin.
- Onboarding the employees by the admin.
- Processing the payroll by admin.
- Checking the schedule by the flight crew.
- Applying the leave by the employees.
- Assisting the passengers by customer support.
- Sending the notification to the user.
- Logging out by the user.

USECASE DIAGRAM:



USECASE DOCUMENTATION:

Use Case – I:

Name of the Use case:

Selecting a Flight by a Passenger

Actors Involved:

- Passenger

Brief Description of the Use case:

This use case describes the process by which a passenger interacts with the Airline Management System to select a flight for their travel.

Pre – conditions:

- The passenger must be logged into the Airline Management System.
- The passenger's personal and payment information must be up to date in the system.

Basic Flow of Events:

1. The passenger logs into the Airline Management System using their username and password.
2. Once logged in, the system displays the main menu.
3. The passenger selects the "Book a Flight" option from the main menu.

4. The system presents a list of available flights based on the passenger's preferences, including departure and destination cities, dates, and times.
5. The passenger reviews the available flight options, including their prices and schedules.
6. The passenger selects a flight by clicking on it or selecting it through other input methods.
7. The system displays a confirmation screen with details of the selected flight, including the total cost.
8. The passenger confirms the selection.
9. If applicable, the system prompts the passenger to enter their payment information. The payment is processed securely.
10. The system generates a booking confirmation with a unique booking reference number.
11. The passenger's booking is now confirmed, and the selected flight is reserved for them.

Alternative Flow of Events:

1. Invalid Login Credentials:

   If the passenger enters incorrect login credentials, an error message is displayed, and the system returns to step 1 of the basic flow.

2. No Available Flights:

   If there are no available flights that match the passenger's criteria, the system displays a message indicating that no flights are found, and the process ends.

3. Payment Failure:

   If the passenger's payment information is invalid or the payment transaction fails, the system displays an error message and prompts the passenger to provide valid payment information.

Post – conditions:

- The selected flight is reserved for the passenger.
- The passenger's payment method is charged for the flight, if applicable.

Supportability Requirement:

- Performance

  The Airline Management System must be accessible via a web interface or mobile application.
- Performance

  The system must be able to handle concurrent requests from multiple passengers.
- Security

  The system must have appropriate security measures in place to protect passenger data and payment information.

Use Case – II:

Name of the Use case:

Ordering Snacks by the Passenger

Actors Involved:

- Passenger
- Flight Attendant
- Airline Management System

Brief Description of the Use case:

This use case describes the process by which a passenger can order snacks while on board an airline flight using the Airline Management System.

Pre – conditions:

- The passenger must be on board the flight.
- The Airline Management System (AMS) must be operational.
- The passenger's seat should be equipped with a seatback entertainment system or a personal device with the AMS app installed.

Basic Flow of Events:

1. The passenger boards the flight and takes their seat.
2. The passenger turns on the seatback entertainment system or opens the AMS app on their personal device.
3. The passenger logs in to their account (if required) or provides necessary identifying information such as name and seat number.
4. The passenger navigates to the "Snack Ordering" section of the AMS interface.

5. The passenger browses the available snack menu, which includes a list of snacks, their descriptions, prices, and images.
6. The passenger selects one or more snacks they wish to order by adding them to their cart.
7. The passenger reviews the contents of their cart and can make any necessary changes (add, remove, or update quantities).
8. Once satisfied with the order, the passenger confirms their snack order.
9. The AMS processes the order and displays a confirmation message to the passenger, including the total cost of the snacks.
10. The passenger's payment information, stored securely by the airline, is used to charge for the snacks.
11. The Flight Attendant is notified of the passenger's snack order through a notification on their handheld device or through the cabin crew interface of the AMS.
12. The Flight Attendant prepares the snacks and beverages as per the order.
13. The Flight Attendant delivers the snacks to the passenger's seat.
14. The passenger receives the snacks and enjoys them during the flight.

Alternative Flow of Events:

1. Payment Failure:
    a. If the payment for the snacks fails for any reason, the AMS displays an error message to the passenger and prompts them to update their payment information.

b. The passenger enters valid payment information.

c. The AMS reprocesses the payment and, upon successful completion, confirms the snack order.

2. AMS Unavailability:

a. If the AMS is temporarily unavailable due to technical issues, the passenger is informed of the unavailability and asked to try again later.

b. The passenger may retry the snack ordering process when the system becomes available.

Post – conditions:

- The passenger's snack order is placed and confirmed.
- The Flight Attendant is notified of the order.
- The payment for the snacks is processed.

Supportability Requirement:

- Usability

  The ordering process should be intuitive and user-friendly to ensure a seamless experience for passengers.

- Security

  Passenger payment information must be securely stored and processed to protect sensitive data.

- Performance

  The Flight Attendant should have easy access to the snack order details to fulfill passenger requests promptly.

Use Case – III:

Name of the Use case:

Finalizing the Booking by Making a Payment

Actors Involved:

- Passenger

Brief Description of the Use case:

This use case covers the interaction between the passenger and the Airline Management System for making a payment to confirm a flight booking. It includes scenarios where the payment is successful or fails.

Pre – conditions:

- The passenger has selected a flight.
- The passenger has provided necessary personal and flight information.
- The passenger has selected a payment method.

Basic Flow of Events:

1. The passenger selects a flight.
2. The passenger provides personal information, including name, contact details, and any required identification.
3. The passenger selects a payment method (e.g., credit card, PayPal, bank transfer).
4. The system calculates the total cost of the booking, including taxes and fees.

5. The passenger reviews the booking details and the total cost.
6. The passenger confirms the booking and initiates the payment.
7. The system processes the payment using the selected payment method.
8. If the payment is successful:
    a. The system generates a booking confirmation with a unique booking reference.
    b. The booking details are updated with the confirmed status.
    c. The passenger receives the booking confirmation via email or on the website.
9. If the payment fails, the system informs the passenger of the failure and provides an opportunity to retry with a different payment method.

Alternative Flow of Events:

1. Payment Failure
    a. If the payment fails:
    b. The system notifies the passenger of the payment failure and provides an error message explaining the issue.
    c. The passenger can choose to:
        i. Retry the payment with the same or a different payment method.
        ii. Cancel the booking.
2. Booking Cancellation
3. If the passenger decides to cancel the booking:

a. The system cancels the booking and releases the reserved seats.

b. The payment, if already processed, may be subject to a refund policy.

Post – conditions:

- The flight booking is finalized and confirmed.
- A booking confirmation is generated and provided to the passenger.

Supportability Requirement:

- Performance

  The system must be able to handle concurrent requests from multiple passengers.

- Security

  The system must have appropriate security measures in place to protect passenger data and payment information.

Use Case – IV:

Name of the Use case:

Updating the flight ticket by passenger

Actors Involved:

- Passenger

Brief Description of the Use case:

This use case covers the interaction between the passenger and the Airline Management System for updating an existing flight reservation. It includes scenarios where the passenger can change flight dates, passenger names, seat preferences, or other related details.

Pre – conditions:

- The passenger has an existing flight reservation.
- The passenger is logged into the Airline Management System.

Basic Flow of Events:

1. The passenger logs into the Airline Management System using their credentials.
2. The system retrieves the passenger's existing flight reservation.
3. The passenger selects the flight reservation they want to update.
4. The passenger specifies the modifications they want to make, such as changing flight dates, passenger names, seat preferences, or other related details.
5. The system validates the requested modifications to ensure they are within the permissible limits and comply with the airline's policies.
6. If the modifications are valid, the system updates the flight reservation accordingly.
7. The system generates an updated booking confirmation with the modified details.
8. The updated booking confirmation is sent to the passenger via email or is made available for download on the website.

Alternative Flow of Events:

1. Ticket Modification Rejection
    a. If the modifications requested by the passenger are not valid.
    b. The system notifies the passenger of the rejection and provides a clear reason for the rejection.
    c. The passenger can choose to modify the reservation again, ensuring the modifications comply with the airline's policies.
    d. If the passenger decides not to proceed with the modifications, the original booking remains unchanged.

Post – conditions:

- The flight reservation is updated according to the passenger's modifications.
- An updated booking confirmation is generated and provided to the passenger.

Supportability Requirement:

- Performance

  The system must be able to handle concurrent requests from multiple passengers.
- Security

  The system must have appropriate security measures in place to protect passenger data and payment information.

Use Case – V:

Name of the Use case:

Cancelling the flight ticket by passenger

Actors Involved:

- Passenger

Brief Description of the Use case:

This use case covers the interaction between the passenger (actor) and the Airline Management System for cancelling an existing flight reservation. It includes scenarios where the passenger can cancel their reservation entirely or cancel specific passengers within a reservation.

Pre – conditions:

- The passenger has an existing flight reservation.
- The passenger is logged into the Airline Management System.

Basic Flow of Events:

1. The passenger logs into the Airline Management System using their credentials.
2. The system retrieves the passenger's existing flight reservation.
3. The passenger selects the flight reservation they want to cancel.
4. The system displays the details of the selected reservation for confirmation, including flight information and passenger names.
5. The passenger confirms the cancellation of the entire reservation or specific passengers within the reservation.

6. The system validates the cancellation request to ensure it complies with the airline's cancellation policy.
7. If the cancellation request is valid, the system cancels the reservation or specific passengers within the reservation.
8. The system generates a cancellation confirmation with the cancelled details.
9. The cancellation confirmation is sent to the passenger via email or is made available for download on the website.

Alternative Flow of Events:

1. Ticket Cancellation Confirmation
    a. After the passenger confirms the cancellation request:
    b. The system displays a confirmation message to the passenger, summarizing the cancelled reservation details.
    c. The passenger receives an email confirmation with the cancelled reservation details.

Post – conditions:

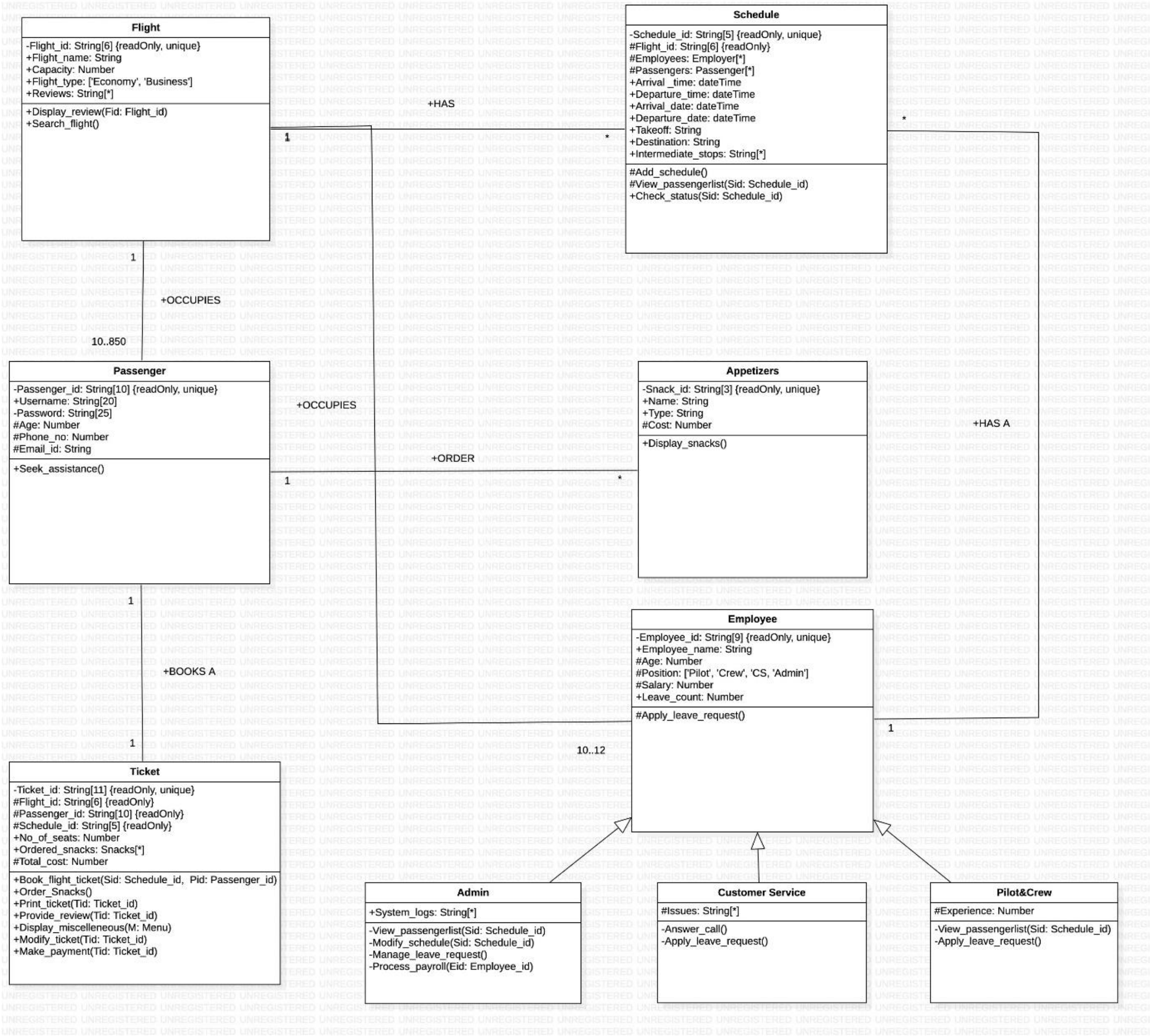
- The flight reservation is cancelled according to the passenger's request.
- A cancellation confirmation is generated and provided to the passenger.

Supportability Requirement:

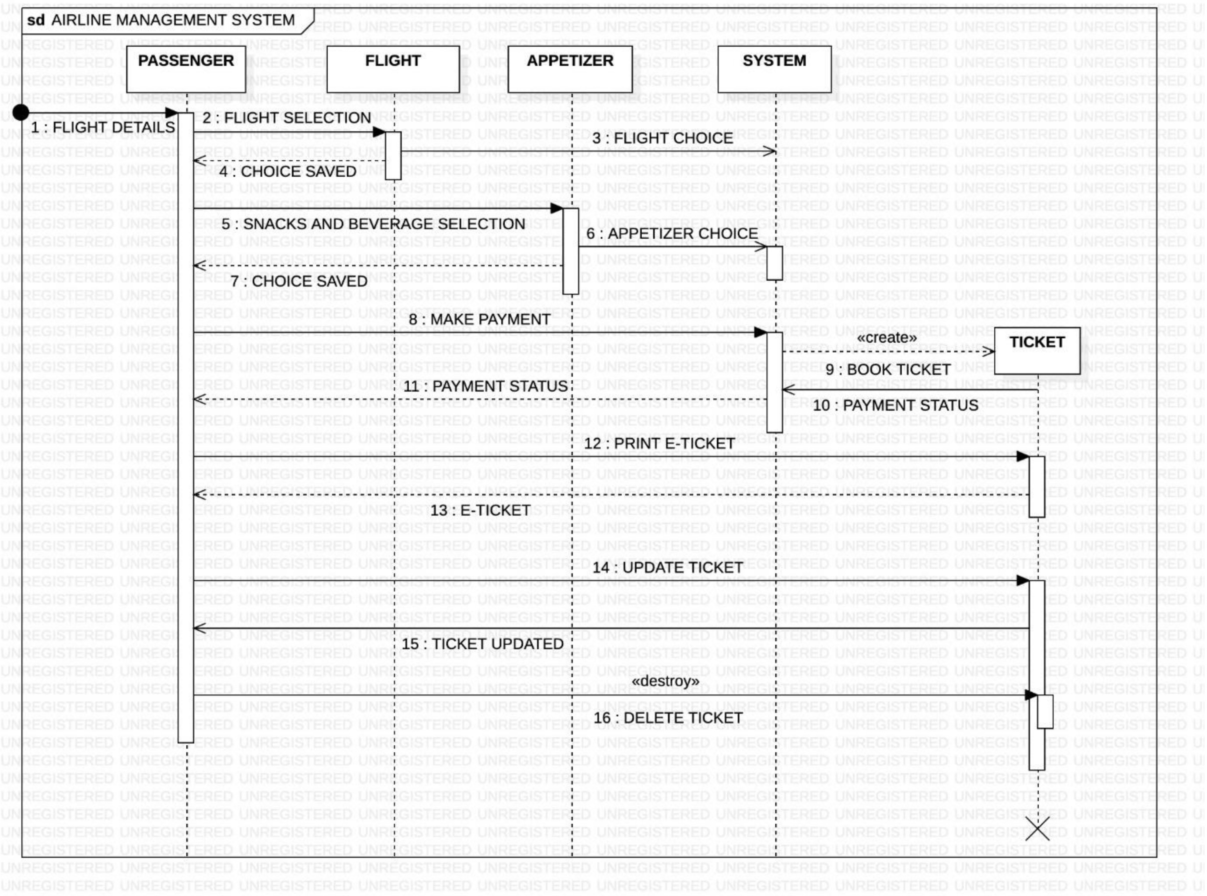
- Security

  The system must have appropriate security measures in place to protect passenger data and payment information.

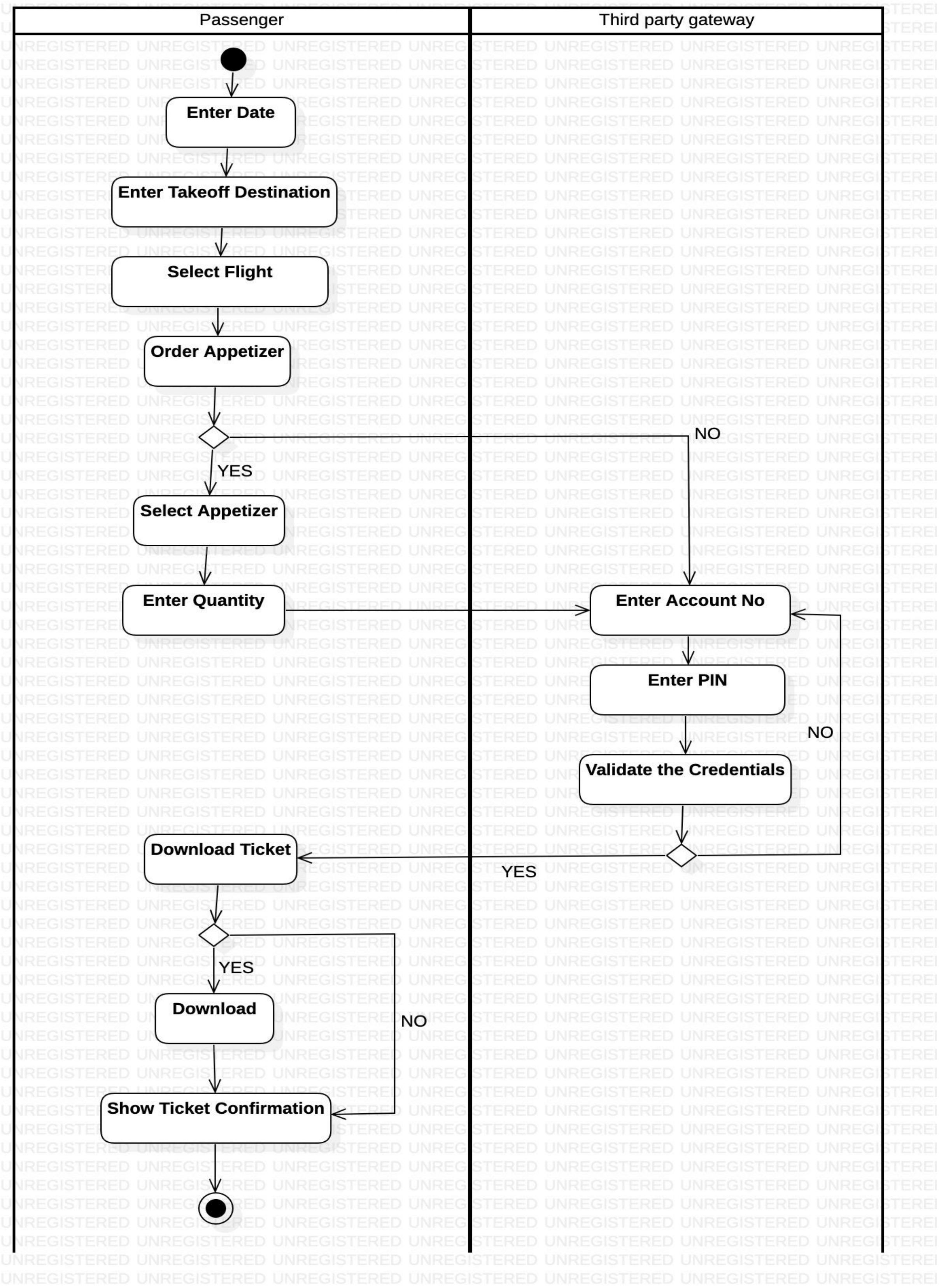
CLASS DIAGRAM


Flight
-Flight_id: String[6] {readOnly, unique}
+Flight_name: String
+Capacity: Number
+Flight_type: ['Economy', 'Business']
+Reviews: String[*]
+Display_review(Fid: Flight_id)
+Search_flight()
+HAS
1
*
Schedule
-Schedule_id: String[5] {readOnly, unique}
#Flight_id: String[6] {readOnly}
#Employees: Employer[*]
#Passengers: Passenger[*]
+Arrival _time: dateTime
+Departure_time: dateTime
+Arrival_date: dateTime
+Departure_date: dateTime
+Takeoff: String
+Destination: String
+Intermediate_stops: String[*]
#Add_schedule()
#View_passengerlist(Sid: Schedule_id)
+Check_status(Sid: Schedule_id)
*
1
+OCCUPIES
10..850
Passenger
-Passenger_id: String[10] {readOnly, unique}
+Username: String[20]
-Password: String[25]
#Age: Number
#Phone_no: Number
#Email_id: String
+Seek_assistance()
+OCCUPIES
+ORDER
1
*
Appetizers
-Snack_id: String[3] {readOnly, unique}
+Name: String
+Type: String
#Cost: Number
+Display_snacks()
+HAS A
1
+BOOKS A
1
Employee
-Employee_id: String[9] {readOnly, unique}
+Employee_name: String
#Age: Number
#Position: ['Pilot', 'Crew', 'CS, 'Admin']
#Salary: Number
+Leave_count: Number
#Apply_leave_request()
10..12
1
Ticket
-Ticket_id: String[11] {readOnly, unique}
#Flight_id: String[6] {readOnly}
#Passenger_id: String[10] {readOnly}
#Schedule_id: String[5] {readOnly}
+No_of_seats: Number
+Ordered_snacks: Snacks[*]
#Total_cost: Number
+Book_flight_ticket(Sid: Schedule_id, Pid: Passenger_id)
+Order_Snacks()
+Print_ticket(Tid: Ticket_id)
+Provide_review(Tid: Ticket_id)
+Display_miscelleneous(M: Menu)
+Modify_ticket(Tid: Ticket_id)
+Make_payment(Tid: Ticket_id)
Admin
+System_logs: String[*]
-View_passengerlist(Sid: Schedule_id)
-Modify_schedule(Sid: Schedule_id)
-Manage_leave_request()
-Process_payroll(Eid: Employee_id)
Customer Service
#Issues: String[*]
-Answer_call()
-Apply_leave_request()
Pilot&Crew
#Experience: Number
-View_passengerlist(Sid: Schedule_id)
-Apply_leave_request()

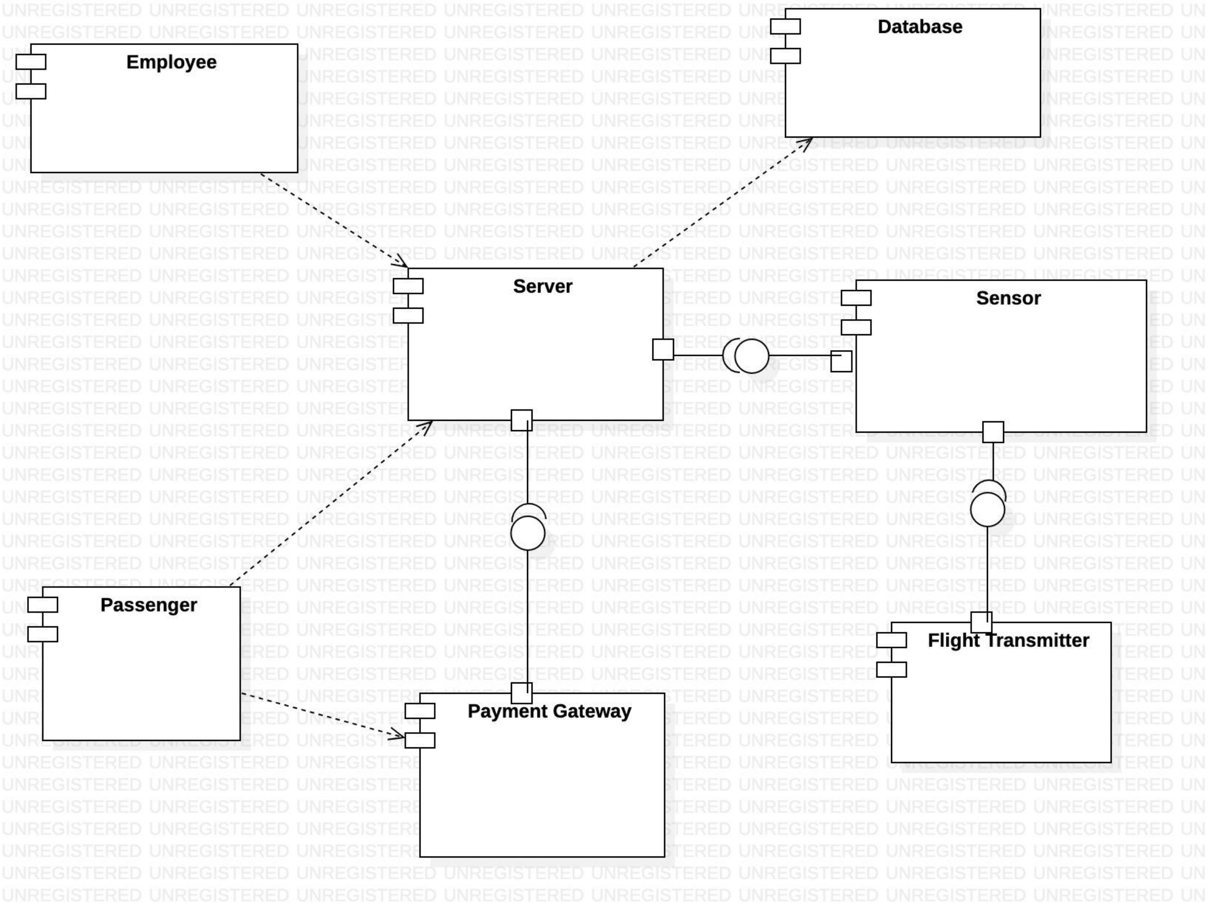
SEQUENCE DIAGRAM



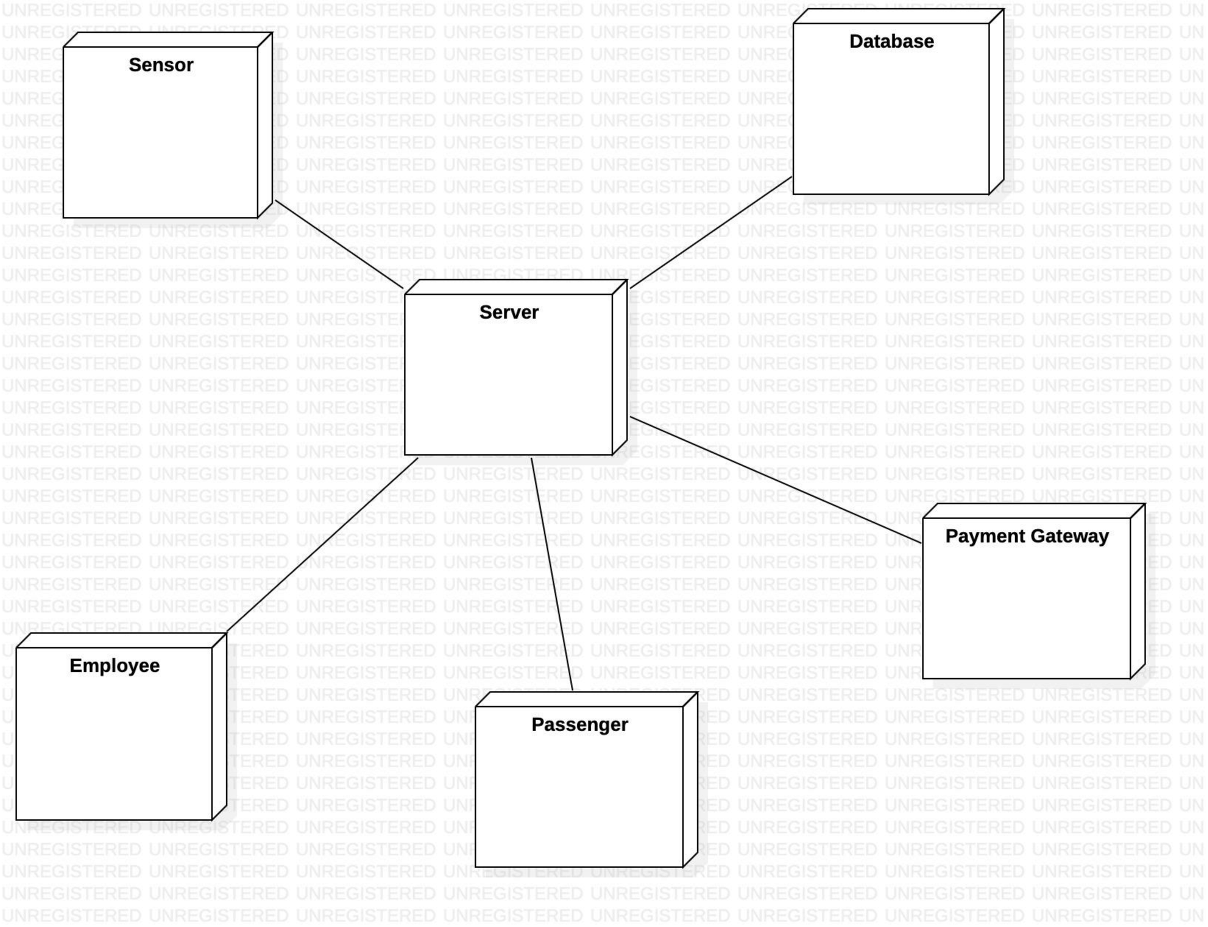
ACTIVITY DIAGRAM



COMPONENT DIAGRAM



DEPLOYMENT DIAGRAM


Sensor
Database
Server
Payment Gateway
Employee
Passenger