Project Topic: Airline Reservation System

Description:

In the realm of air travel, the Airline Reservation System stands as a beacon, guiding travelers and airlines alike through the complexities of flight bookings and management. It serves as a pivotal response to the pressing need for efficient coordination between passengers and airline services. Understanding the intricacies of travel logistics, our mission is to streamline the flight booking process and enhance the overall travel experience. With a clear vision, we aim to bridge the gap between travelers and airlines, fostering seamless interactions and facilitating smooth journeys for all.

Scope:

The scope of the Airline Reservation System is to develop a comprehensive online platform that caters to the needs of both travelers and airlines. It seeks to simplify the flight booking process, optimize seat management, and provide real-time updates on flight availability. The system will offer a range of services and resources aimed at enhancing the travel experience, ensuring accessibility to users worldwide. Regardless of their location or travel preferences, the system aims to deliver a seamless booking experience for all passengers and airlines.
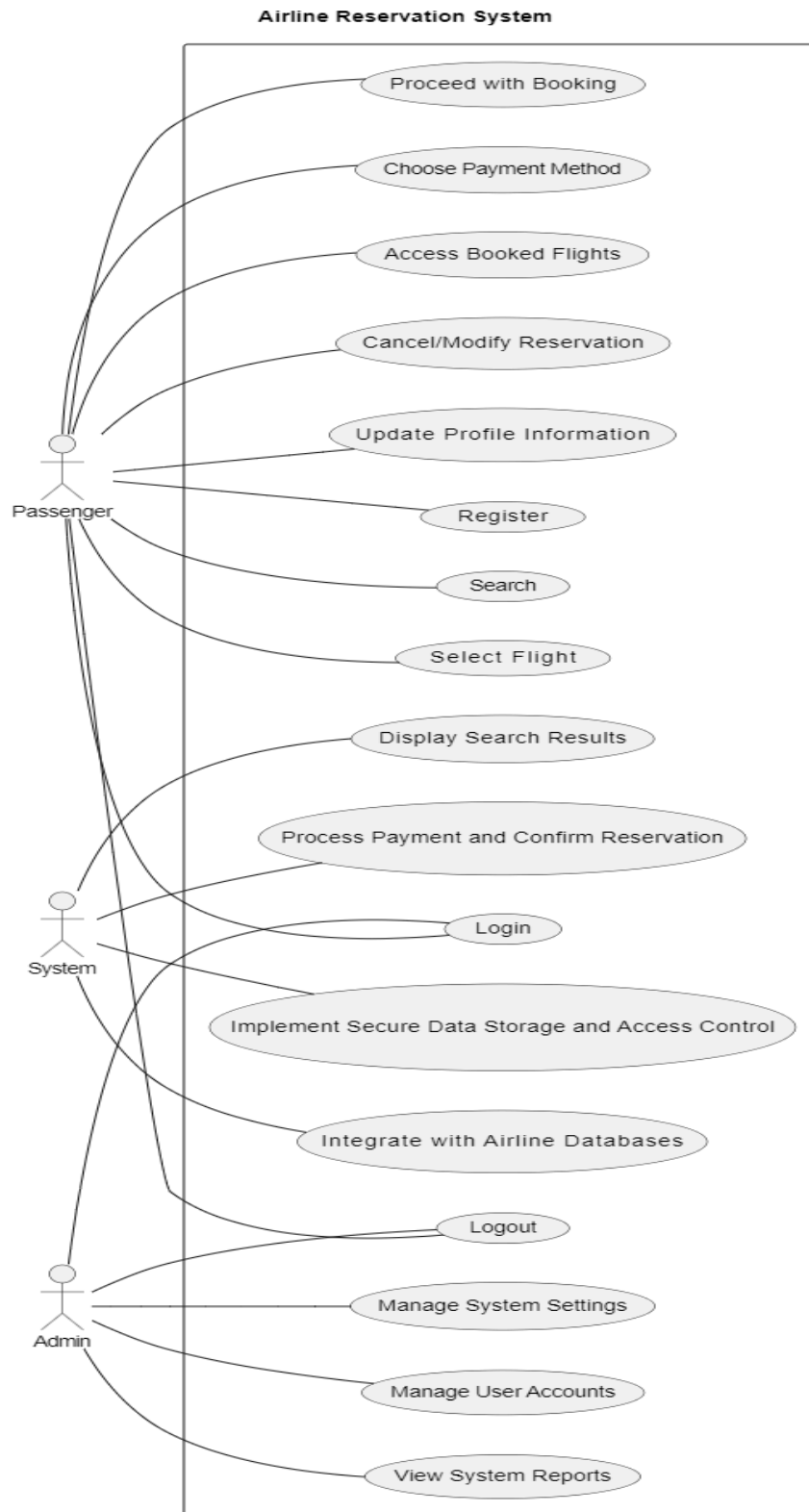
Features:

1. User Registration
2. Real-Time Flight Information Integration
3. Flight Search
4. Search Result Display
5. Flight Booking
6. Secure Payment Processing
7. Booking Management
8. Profile Management
9. Admin User Management
10. Data Security
11. System Reporting
12. System Configuration
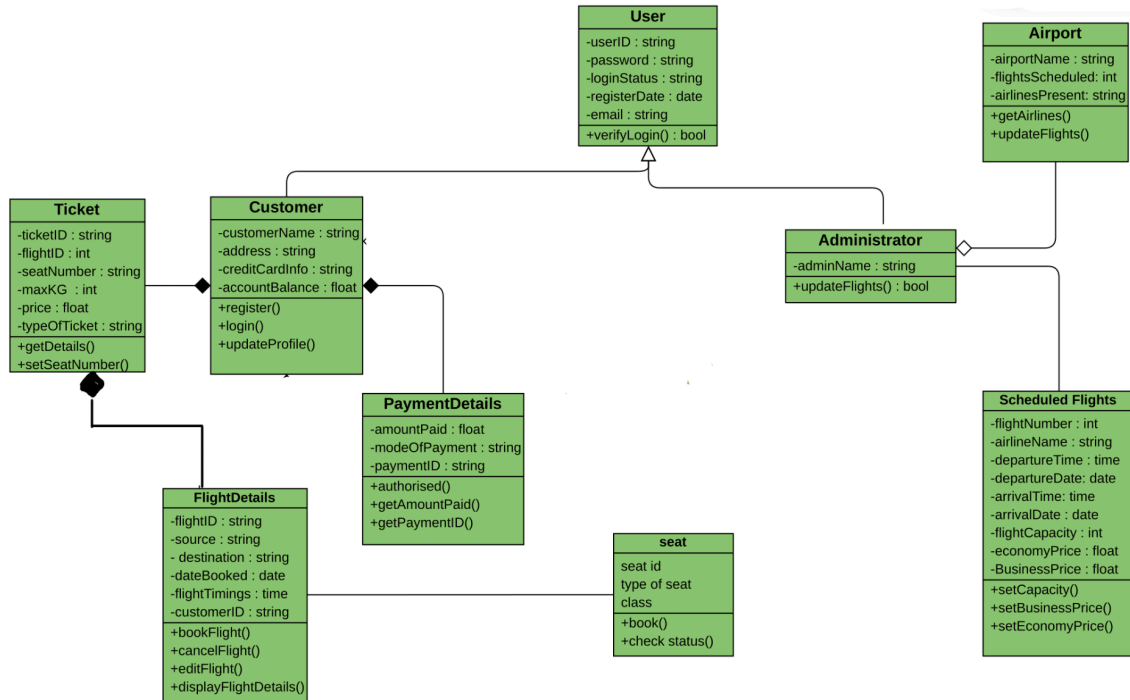
Requirement details with actors (External Entity):

ReqNo	External Entity	Initial Requirement
1	Passenger	Passenger shall Register for an account by providing username, email, password, and additional details (name, contact information).
2	System	System shall Integrate with airline databases to retrieve real-time flight information and availability.
3	Passenger	Passenger shall search for flights by specifying origin, destination, travel dates, and number of passengers.
4	System	System shall display search results including flight details (airline, departure/arrival times, price) and current seat availability.
5	Passenger	Passenger shall select a flight and proceed with booking.
6	Passenger	Passenger shall hoose a preferred payment method and provide secure billing information.
7	System	System shall process the payment securely and confirm the flight reservation upon successful payment.
8	Passenger	Passenger access and view booked flights under their account profile.
9	Passenger	Passenger shall cancel or modify existing reservations subject to airline policy and fees.
10	Passenger	Passenger shall update profile information (contact details, travel preferences).
11	Admin	Admin manage user accounts (passengers and airlines, if applicable) - add, delete, or suspend accounts.
12	System	System shall implement secure data storage practices and access control mechanisms to protect user information.
13	Admin	Admin shall view system reports on booking trends, revenue, and user activity.
14	Admin	Admin shall manage system settings and configurations.

Use Case diagram:

Airline Reservation System

Passenger
- Proceed with Booking
- Choose Payment Method
- Access Booked Flights
- Cancel/Modify Reservation
- Update Profile Information
- Register
- Search
- Select Flight

System
- Display Search Results
- Process Payment and Confirm Reservation
- Login
- Implement Secure Data Storage and Access Control
- Integrate with Airline Databases
- Logout

Admin
- Manage System Settings
- Manage User Accounts
- View System Reports

ClassDiagram(Without Patterns):

Airline Reservation System

ClassDiagram(With Patterns):

Ticket
-ticketID : string
-flightID : int
-seatNumber : string
-maxKG : int
-price : float
-typeOfTicket : string
+getDetails()
+setSeatNumber()

User
-userID : string
-password : string
-loginStatus : string
-registerDate : date
-email : string
+verifyLogin() : bool

Customer
-customerName : string
-address : string
-creditCardInfo : string
-accountBalance : float
+register()
+login()
+updateProfile()

Administrator<<Singelton>>
-Administrator()
-static instance:Administrator*
-static getinstance():Administrator
-Administrator(obj:Administrator&)
-Adminstrator=(obj:Administrator&)
//other data members or methods

PaymentDetails<interface>
+setCardholderName(string name):void
+getCardholderName():string
+clone():PaymentDetails

Scheduled Flights
-flightNumber : int
-airlineName : string
-departureTime : time
-departureDate: date
-arrivalTime: time
-arrivalDate : date
-flightCapacity : int
-economyPrice : float
-BusinessPrice : float
+setCapacity()
+setBusinessPrice()
+setEconomyPrice()

Airport
-airportName : string
-flightsScheduled: int
-airlinesPresent: string
+getAirlines()
+updateFlights()

FlightDetails
-flightID : string
-source : string
- destination : string
-dateBooked : date
-flightTimings : time
-customerID : string
+bookFlight()
+cancelFlight()
+editFlight()
+displayFlightDetails()

seat
seat id
type of seat
class
+book()
+check status()

CreditCardPaymentDetail
+getcardHolderName():string
+getcardNumber():string
+setcardHolderName(string):void
+setcardHolderNumber(string):voi
+clone():PaymentDetails
+// other Methods
-cardHolderName:string
-cardNumber:string
-//other members variable

PayPalPaymentDetails
+setemailaddress(mail:string):void
+getemailaddress():string
+clone():PaymentDetails
-emailaddress:string