Assignment: Design a Flight Booking System

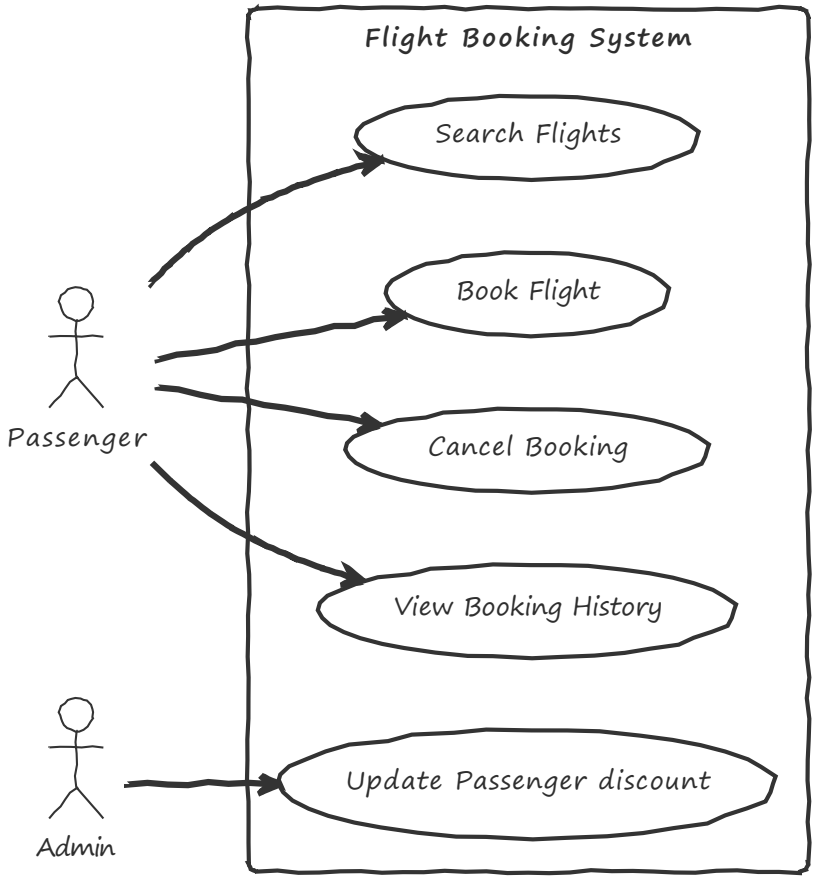
Objective:

The objective of this assignment is to design a skeleton for a Flight Booking System using Object Oriented Analysis and Design, specifically parsing use case diagrams and descriptions in order to create UML diagrams and code.

Task:

Based on the use case diagram and the use case descriptions below, create **class** and **sequence** diagrams and a **Java application skeleton** with all identified fields and methods. In the application skeleton, the methods don't need any logic implemented, they only need to be declared (ie, leave the methods empty). The business analyst working with the requirements has decided that we need the following classes: `Flight`, `Passenger`, `Seat` and `Booking`. Take care to identify all fields, all associations, the type of associations used and the associations' multiplicity by parsing the use cases. Add the fields and methods needed to execute the use cases. If a flight were to be removed, all its seats are also removed.

Add a Main class with a main method which does nothing more than creating one object of each class (this is only to facilitate for the compiler to check all classes).



Grading Criteria:

- Correctly associated all classes, using aggregation, composition and multiplicity as needed
- Added fields and methods with parameters and return types that are needed to execute the use cases
- Used all declared methods in the sequence diagram, covering all use cases
- Correctly translated the diagrams to Java code, including fields for the associations (remember: no logic, ie empty method bodies)
- Added a Main class with a main method which instantiates objects of each class

Submission: Submit your diagrams as images (png files) and your Java source code files (`.java`) according to the instructions on the Canvas assignment page.

Use Case 1: Search Flights

Description: This use case allows users to search for available flights based on the following criteria: departure city, destination city, date and time, class of service and price range.

Actors:

- Passenger (Primary Actor)

Flow of Events:

1. Passenger provides search criteria (departure city, destination city, date, class, price range).
2. System retrieves and displays a list of available flights matching the criteria, showing flight number, departure time (with a time zone), arrival time (with a time zone), seats and their classes, and price (a function of passenger discount, flight and seat).
3. Passenger reviews the list of flights and available seats.

Use Case 2: Book Flight

Description: This use case enables passengers to select a specific seat on a flight and make a booking.

Actors:

- Passenger (Primary Actor)

Flow of Events:

1. Passenger selects a seat on a flight from the list of available flights and their seats. A seat is identified by its seat number, such as "19A".
2. System prompts for passenger details (name, contact information, payment method).
3. Passenger provides details and selects a payment method.
4. A new passenger is created based on this information unless it already exists.
5. System validates the information, reserves the seat on the flight for the passenger, and displays a booking confirmation with booking reference.

Use Case 3: Cancel Booking

Description: This use case allows passengers to cancel an existing flight booking.

Actors:

- Passenger (Primary Actor)

Flow of Events:

1. Passenger requests to cancel a booking by providing the booking reference.
2. System retrieves the booking based on the reference and confirms the cancellation request, displaying booking details such as flight number, departure time, arrival time, seat and its class, and price.
3. System cancels the booking and updates the availability of the flight.

Use Case 4: View Booking History

Description: This use case enables passengers to view their booking history.

Actors:

- Passenger (Primary Actor)

Flow of Events:

1. Passenger requests to view booking history.
2. System retrieves and displays a list of past bookings associated with the passenger, showing booking references, flight numbers, departure times, arrival times, seats and their classes, and prices.

Use Case 5: Update Passenger Discount

Description: This use case enables a company admin to update a Passenger's discount.

This use case doesn't need to be modelled.