FLIGHT BOOKING SYSTEM

- **Discussed Business Problem :**

1. A Flight Booking System will keep a track and maintain booking of records efficiently and safely.
2. Booking details should be easily recoverable and retrievable from the database.
3. Airlines need to constantly process bookings from expectant passengers, they need a means of keeping on top of everything. Therefore, storing the records of flight schedules and passenger details will keep the data organized.
4. The automated flight booking system will allow only authorized users to access the information.
5. The system reduces redundancy by maintaining a centralized database and storing all information only once.

- **Entities Present:**

1. Airport
2. Passenger
3. Flight_Details
4. Flight_Service
5. Seat_Details
6. Flight_Cost
7. Reservation
8. Travel_Class
9. Calendar
10. Payment
11. Payment_Status
12. Service_Offering

- **Relationships between entities:**

1. An airport manages multiple flights. An airport may or may not have a flight landing/takeoff. Hence, it's optional. Whereas for flight details, it is mandatory to have one airport association.
2. A flight service may be offered in a particular class
3. A flight comprises multiple seats. It is mandatory that the flight consists of a seat. For Seat details to exist, it is mandatory that it is linked to at least one flight.
4. A passenger can make multiple reservations for a flight. Passengers may or may

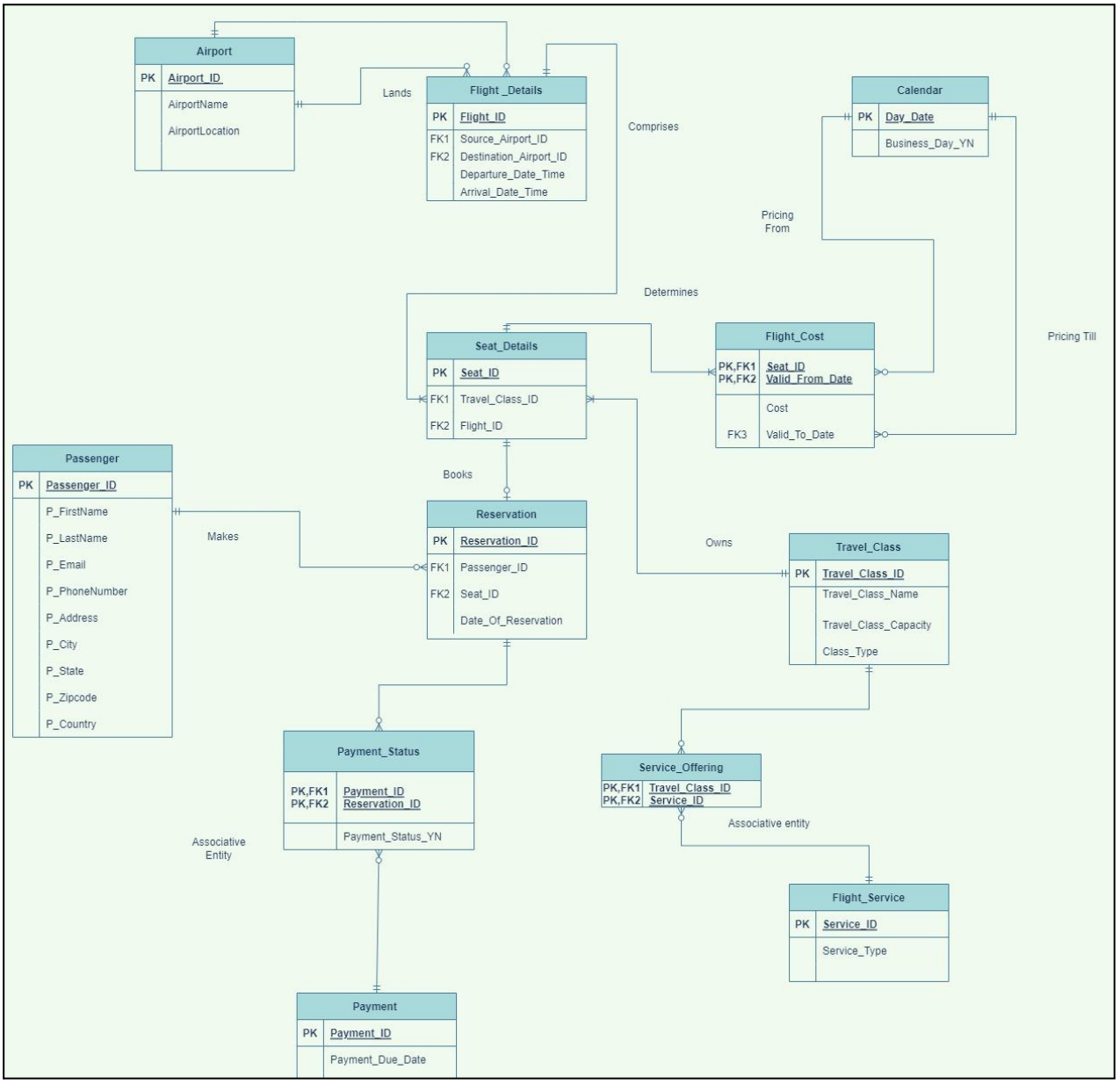
not reserve a seat in the flight. However, it is mandatory that reservation has to be associated with only one passenger.

5. Reservation confirms the Seat Details. It is mandatory that a reservation is linked with a Seat. A seat may or may not be linked with a reservation. One seat can only be linked to one reservation.
6. A travel class should consist of multiple seats. A seat should be linked to only one travel class.
7. A Payment may consist of multiple payment status. Whereas, a payment status should have a Payment associated with it.
8. A Payment Status should have a Reservation. A reservation may consist of multiple payment status.
9. The Flight cost should be determined from the Calendar.
10. The Seat Details should be associated with a Flight cost. The Flight Cost is calculated based on the Seat Details.

- **Key Design:**

1. Each Airport can have multiple flight details associated with its respective source and destination.
2. Flight services like Food, Entertainment, Wifi etc and other exclusive services may be offered to passengers depending on the travel class. Service_Offering is an associative entity which stores Service_ID and Travel_Class_ID.
3. Each Flight mandatorily comprises one or many seats and each seat is associated with a particular Travel_Class_ID. A seat can belong to Business class or economy class etc.
4. Each passenger may or may not make multiple reservations. If a passenger makes a reservation, a seat of the passenger's choice will be assigned. The Seat_Details will point to Travel_Class and Flight_Details. For each seat reserved, a passenger may or may not pay to confirm the seat. A passenger can reserve one seat per Reservation_ID.
5. Payment_Status is an associative entity which holds the information related to particular Reservation_ID and Payment_ID.
6. If the passenger has or hasn’t done the payment it will reflect in payment status and the respective payment due date and amount will be stored in Payment.
7. Flight_Cost can be determined on the basis of Seat_Details and Calendar from which a particular seat belongs to a travel class and the date when the seat is booked respectively.

- ER DIAGRAM :-



- **Reference:**

  https://studylib.net/doc/25306078/flight-managment-system

- **Comparison:**

1. A Passenger can make multiple reservations, but here each Reservation is linked to only one Seat_Details and Seat_Details is the core of the ER diagram, as it links Travel Class, Flight_Details, Flight_Cost.
2. Flight_Cost is determined by multiple factors. The Flight_Cost is dependent on the particular Calendar period, Flight_Details, Travel Class which all are linked together using Seat_Details.
3. We have included a potential enhanced entity - Flight_Service, which will be divided into multiple categories such as Food, Wifi, Entertainment, etc.
4. The Travel Class is also an enhanced entity which is linked to the Flight_Service entity through Service_Offering.
5. We have considered Payment status as an associative entity which would record the status of a particular reservation and payment ID. This would be useful in the scenario when a Reservation is made, but the payment is not confirmed by the passenger. This will help us to determine the confirmation of the ticket.
6. We need source & destination link in flight details, which are referring to the same entity- Airport, but providing us with different information. Hence, we adopted this relationship approach from the reference.

During our first project meeting we were getting to know each other and how we travel to the States. In general we got to know that we traveled through the same airlines i.e Qatar Airlines. So, on the topic of choosing the project idea we landed on going with a flight booking system which can belong to any particular airline.

Then we started researching other flight booking systems, what entities it would contain, and what possible relationships it would contain. So, we picked the above mentioned reference which gave the idea about the entity types and how they can be related.

So, we came up with the first version of ER Diagram which had common entity types like passengers details, flight details, reservation etc. While discussing different entities we thought we can include the calendar which will play a vital role in determining the flight cost. Then we came up with new entities such as Seat Details which became the core of our ER Diagram.

Seat Details - Helped us determine the flight cost along with the calendar as it was linked with flight details and travel class, these are the key details in determining the flight cost in any airlines.

To increase the scope of our ERD we decided to come up with the flight services entity, which can be enhanced depending on the Business Rules in the later stages.

So, there would be many flight services offered to different travel classes. Hence, we have used Service_Offering as an Associativity Entity between them. i.e. one particular service may or may not be offered in one particular class . The flight service depends on the class

which the passenger has opted for while booking.Travel_Class is linked to the Flight_Service entity through Service_Offering. Travel class and flight services can be made enhanced entities in the later stages.

So, we started thinking about the passenger perspective and how his/ her choices would affect storage inside the database.

Based on the reference we inferred that the payment entity conforms to our business rule which is a passenger can reserve a particular seat but may not end up paying for it. So, after a certain period of time if the payment is not completed by the passenger then the reservation is nullified.

We would know the status of this payment through our attribute Payment_Status_YN in our associative entity (Payment_Status).

- **New changes in ERD:**

Initially Service_Offering was linked to Flight_Details and Flight_Service. Now, It has been changed into an Associative Entity between Travel_Class and Flight_Service. A particular service might be exclusive to a class therefore, we generalized the relationship between Flight_Service and Travel_Class through Service_Offering. There is no longer a need to link Flight_Service and Flight_Details.