

Group 1

Huimin Guo, Neha Yadav, Payal Chavan, Yanting Chen, Yuchu Liu

Database Topic

Flight Booking System

Introduction:

In today's fast-paced world, booking a flight is a multi-step process that demands meticulous organization and efficiency. Airlines, catering to the needs of travelers with varying itineraries and preferences, require robust systems to manage reservations reliably. The advent of technological advancements, particularly the introduction of computers, has revolutionized the airline industry. This digital transformation marked a significant shift from traditional, manual methods to more sophisticated, automated record-keeping practices. These developments have not only streamlined the reservation process but also enhanced the overall experience for both airlines and passengers, accommodating the dynamic nature of travel demands with unprecedented precision and ease.

Objectives:

1. To easily restore and retrieve booking information from the database.
2. To avoid data corruption and to maintain data integrity.
3. To reduce data entry errors that affect the system's stored data.
4. The automated flight booking system will only allow access to the information by authorized individuals.
5. To maintain a centralized database to avoid duplication of data.

Purpose:

The primary objective of this project is to develop and sustain a user-friendly interface for flight search and booking, tailored to meet the needs of customers. This system is designed to facilitate users in verifying their bookings and tickets, accessing comprehensive pricing details, and exploring a variety of flight options available at different times on given dates. It meticulously catalogs every critical aspect of travel planning, encompassing origins, destinations, and any layovers or stops along the journey.

To accommodate an extensive range of data, the system is engineered to efficiently store, organize, and manage a substantial volume of information. This includes a comprehensive database encompassing a diverse array of flights. The focus is on ensuring that this wealth of data is not only easily accessible but also well-structured and maintained, providing a seamless and informative experience for users planning their travels.

Business Problems:

1. Information will only be accessible to authorized users via the automated flight booking system.
2. A flight booking system will efficiently and safely retain track of and book records.
3. Booking information ought to be effortlessly retrieved from the database.
4. To manage the frequent and ongoing requests of travelers, airlines require a system that can handle all the information. By keeping track of the flight plans and the personal data of the passengers, they can ensure order and efficiency.
5. By keeping an organized database and saving all data once, the system lowers redundancy.

Final ERD Design:



Data Description:

Entity List:

1. Airplane
2. Airline
3. Flight
4. Airport
5. User
6. Booking
7. Passenger
8. Flight Status
9. Payment
10. Ticket
11. Seats
12. Promotion

Entity structures with relevant attributes:

1. Airplane

Field	Description	Constraints
AirplaneID	unique airplane no.	Primary Key
AirlineID	Identifies the Airline	Foreign Key
Model	Flight model	NOT NULL
Capacity	No. of seats available	NOT NULL
YearOfManufacture	Manufacture year of the airplane	NOT NULL
LastMaintenanceDate	Most recent date of airplane maintenance	NOT NULL

2. Airline

Field	Description	Constraints
AirlineID	unique airline no.	Primary Key

AirlineName	Identifies the Airplane	NOT NULL
ContactInfo	Airline contact info	NOT NULL
Rating	Airline rating given by passengers	NOT NULL

3. Flight

Field	Description	Constraints
FlightID	unique flight no.	Primary Key
AirlineID	Identifies the Airline	Foreign Key
AirplaneID	Identifies the airplane	Foreign Key
ArrivalAirportID	Arrival airport no. of a flight	Foreign Key
DepartureAirportID	Departure airport no. of a flight	Foreign Key
DepartureDateTime	Departure time of the flight	NOT NULL
ArrivalDateTime	Arrival time of the flight	NOT NULL
AvailableSeats	No. of seats available	NOT NULL
Price	Total fare of the flight	NOT NULL

4. Airport

Field	Description	Constraints
AirportID	unique airport no.	Primary Key
AirportName	Identifies the name of the airport	NOT NULL

City	City belonging to airport	NOT NULL
State	State belonging to airport	NOT NULL
Country	Country of the airport	NOT NULL
Code	Specific airport code	NOT NULL

5. User

Field	Description	Constraints
UserID	unique user no.	Primary Key
FirstName	First name of the user	NOT NULL
LastName	Last name of the user	NOT NULL
Email	Email address of the user	NOT NULL
Password	Password of the user	NOT NULL
Phone	Contact info of the user	NOT NULL

6. Booking

Field	Description	Constraints
BookingID	Stores unique row number	Primary Key
UserID	User ID from “User” table	Foreign Key
FlightID	Flight model	Foreign Key
BookingDateTime	stores the booking date and time	NOT NULL

Status	Booking Status- Confirmed/Canceled	NOT NULL
--------	---------------------------------------	----------

7. Passenger

Field	Description	Constraints
PassengerID	unique passenger no.	Primary Key
FirstName	Passenger's first name	NOT NULL
LastName	Passenger's last name	NOT NULL
Age	Age calculated by birthdate	NOT NULL
Gender	Passenger's gender	NOT NULL
BirthDate	Passenger's birthdate	NOT NULL

8. FlightStatus

Field	Description	Constraints
FlightStatusID	unique flight status no.	Primary Key
FlightID	Flight ID from "Flight" table	Foreign Key
Status	Flight is on time/delayed/canceled	NOT NULL
StatusDate	Stores date of flight status	NOT NULL

9. Payment

Field	Description	Constraints
PaymentID	unique payment no.	Primary Key

BookingID	Booking ID from “Booking” table	Foreign Key
Amount	Total amount paid	NOT NULL
PaymentDateTime	Date and time of the payment made	NOT NULL
PaymentMethod	Type of payment	NOT NULL
Status	Payment failed/successful/pending	NOT NULL

10. Ticket

Field	Description	Constraints
TicketID	unique flight status no.	Primary Key
BookingID	Booking ID from “Booking” table	Foreign Key
PassengerID	Passenger ID from “Passenger” table	Foreign Key
BoardingPassURL	Boarding Pass of the passenger	NOT NULL
SeatNumber	Seat no. assigned to the passenger	NOT NULL

11. Seats

Field	Description	Constraints
SeatID	unique seat no.	Primary Key
FlightID	Flight ID from “Flight” table	Foreign Key
SeatNumber	Seat no. of passenger	NOT NULL

IsBooked	Status of seat booking	NOT NULL
ClassType	Seats class	NOT NULL

12. Promotion

Field	Description	Constraints
PromotionID	unique promotion no.	Primary Key
AirlineID	Airline ID from “Airline” table	Foreign Key
DiscountPrice	Discount amount	NOT NULL
StartDate	Stores start date of promotion	NOT NULL
EndDate	Stores end date of promotion	NOT NULL
PromoCode	Promo code label	NOT NULL

Relationships between entities:

1. An airport manages multiple flights. Hence, there is a one-to-many relationship from Airport→Flight. As flights must be associated with airport, it is a mandatory relationship. Each flight has one departure and one arrival airport. So, there is a many-to-one relationship from Flight→Airport.
2. There are several seats on a flight. A seat is a requirement for the flight. Seat information must be associated with at least one flight in order for them to exist. Hence, there is a one-to-many relationship from Flight→Seats, and many-to-one relationship from Seats→Flights.
3. Each user may or may not make multiple bookings. Also, each booking is made by one user. Hence, there is one-to-many relationship from User→Booking, and many-to-one relationship from Booking→User.
4. A booking can have multiple tickets, whereas each ticket is associated with one booking. As a result, there is a one-to-many relationship from Booking→Ticket, and many-to-one relationship from Ticket→Booking.

5. A user can make a booking with multiple payments. Each payment is made for one booking. Accordingly, there is a one-to-many relationship from Booking→Payment, and many-to-one relationship from Payment→Booking.
6. Each ticket is issued to one passenger. Also, passengers may or may not have multiple tickets. Hence, there is a one-to-many relationship from Passenger→Ticket, and many-to-one relationship from Ticket→Passenger.
7. A flight can have multiple bookings, and each booking can be made for one flight. Therefore, there is a one-to-many relationship from Flight→Booking, and many-to-one relationship from Booking→Flight.
8. Each flight is associated with a flight status, and vice-versa. Hence, there is a one-to-one relationship between Flight→FlightStatus and FlightStatus→Flight.
9. Each airplane is owned by one airline. An Airline owns multiple airplanes. Thus, there is a one-to-many relationship between Airline→Airplane, and many-to-one relationship between Airplane→Airline.
10. An airline can offer multiple promotions. Also, each promotion is offered by one airline. Hence, there is a one-to-many relationship between Airline→Promotion and many-to-one relationship between Promotion→Airline.

Cardinality:

Serial no.	Entities	Cardinality
1.	Airline & Airplane	1: M
2.	Booking & Payment	1: M
3	Seats & Flight	M: 1
4.	Airline & Promotion	1: M
5.	Flight & FlightStatus	1: 1
6.	Flight & Airport	M: 1
7.	Flight & Booking	1: M
8.	Booking & Ticket	1: M
9.	User & Booking	1: M
10.	Passenger & Ticket	1: M

Key Design Decisions:

1. **Flight and Airport Association:** Multiple flight details will be associated with a single airport, serving as either the point of departure or the destination. This centralized linkage ensures that each airport in the database serves as a hub of information for all flights arriving at or departing from its location.
2. **Dynamic Pricing Mechanism:** The cost of a flight will be dynamically calculated based on the details of the seat (such as class and location within the aircraft) and the date of booking. This approach allows for flexible pricing strategies that can adapt to demand and booking timelines, optimizing revenue for the airline.
3. **Payment Status Tracking:** The payment status for each passenger will be clearly defined within the system, indicating whether their payment is pending, complete, or overdue. The 'Payment' entity will encapsulate crucial details, including the payment amount and the due date, to ensure efficient financial management and customer communication.
4. **FlightStatus Association:** The 'FlightStatus' entity will function as an associative entity, maintaining a direct correlation with the 'FlightID' to track real-time statuses of flights. This entity will contain information pertinent to flight schedules, such as delays, cancellations, and other operational details that are essential for both the airline's workflow and the passengers' travel plans.
5. **Promotional Offers Management:** Airlines will have the capability to offer multiple promotions simultaneously, each with its own set of terms and validity period. This design decision allows for a robust promotional strategy, enabling airlines to attract a diverse customer base with various offers and discounts.