Milestone 1 for SkyRoutes

Overview

SkyRoutes is a flight booking database system designed to facilitate the searching, booking, and management of flight tickets for travelers globally. It aims to provide a user-friendly interface that integrates flight schedules, ticket pricing, and user profiles to enhance the booking and boarding experience.

Team Member

- Mingjian Li - ml8347@nyu.edu
- Fuxian Gong - fg1118@nyu.edu
- Yifan Hu - yh5435@nyu.edu
- Junjie Wang - jw8180@nyu.edu

Key Entities and Attributes

1. User: user_id (PK), name, email (Unique), phone, passport_number (Unique)
2. Passenger: passenger_id (PK), @booking_id (FK references Booking), name, passport_number, dob (Date of Birth)
3. Airport: airport_id (PK), airport_name, city, country, iata_code (Unique), icao_code (Unique)
4. Aircraft: aircraft_id (PK), model, total_seats, @airline_id (FK references Airline)
5. Flight: flight_id (PK), flight_number (Unique), @departure_airport_id (FK references Airport), @arrival_airport_id (FK references Airport), departure_time, arrival_time, @aircraft_id (FK references Aircraft), @airline_id (FK references Airline)
6. Seat: seat_id (PK), @flight_id (FK references Flight), seat_number, class_type (Economy, Business, First), price
7. Booking: booking_id (PK), @user_id (FK references User), @flight_id (FK references Flight), @seat_id (FK references Seat, Unique), booking_time, status (Paid, Unpaid, Cancelled), @payment_id (FK references Payment)
8. Payment: payment_id (PK), @booking_id (FK references Booking), payment_method (Credit Card, PayPal, Alipay), payment_status (Completed, Pending, Failed)
9. Airline: airline_id (PK), airline_name (Unique), country
10. Crew: crew_id (PK), @flight_id (FK references Flight), name, role (Pilot, Co-Pilot, Flight Attendant)
11. Luggage: luggage_id (PK), @booking_id (FK references Booking), weight, dimensions
12. Check-in: checkin_id (PK), @booking_id (FK references Booking), @seat_id (FK references Seat), checkin_time

13. Boarding Pass: boarding_pass_id (PK), @passenger_id (FK references Passenger), @flight_id (FK references Flight), seat_number, boarding_gate, boarding_time
14. Flight Status: flight_status_id (PK), @flight_id (FK references Flight), status (On-Time, Delayed, Cancelled), last_updated
15. Baggage Claim: baggage_claim_id (PK), @airport_id (FK references Airport), @flight_id (FK references Flight), baggage_belt_number

Relationships

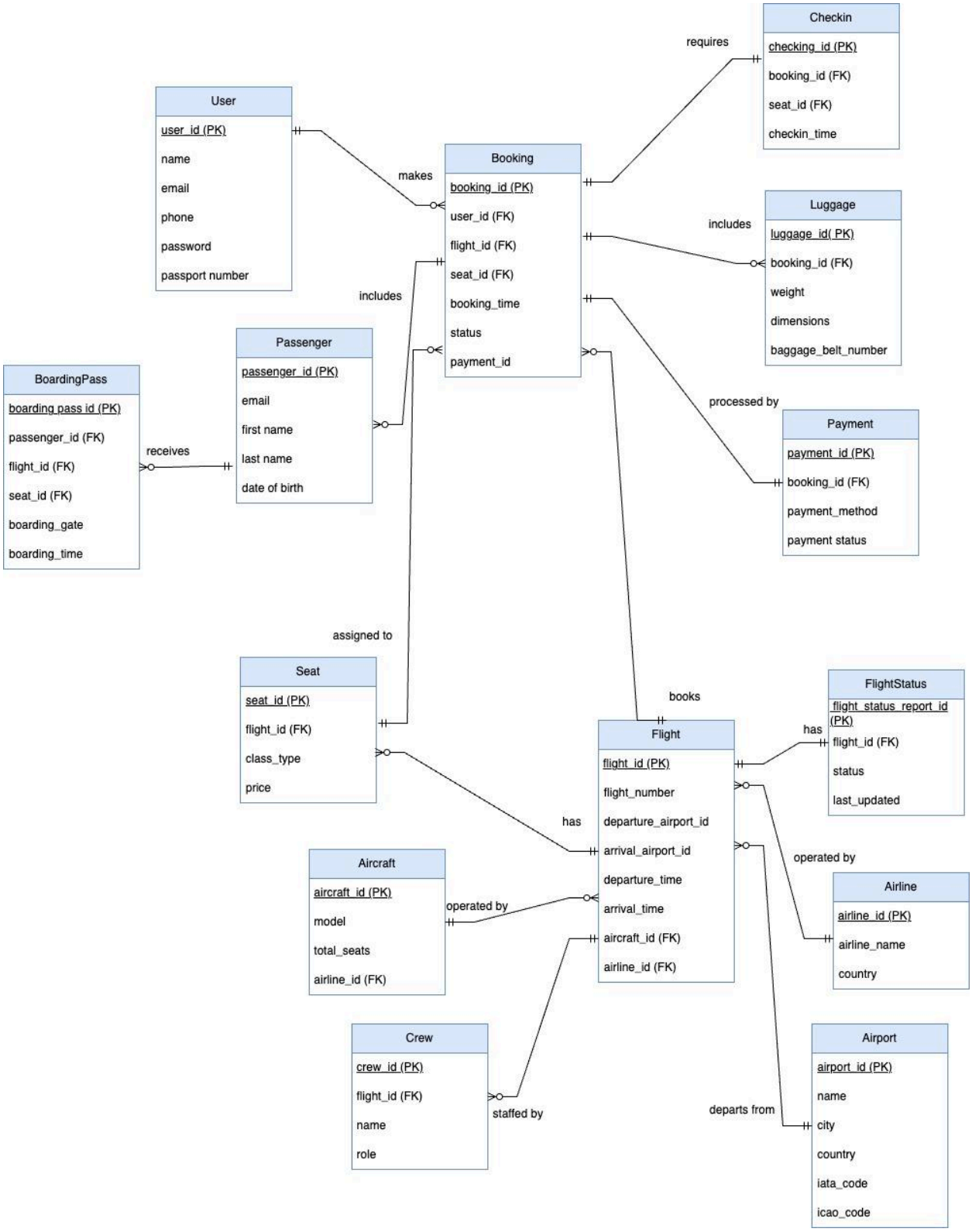
- A **User** can make multiple **Bookings**, but each **Booking** belongs to only one **User** (1:N).
- A **Booking** can have multiple **Passengers**, ensuring that a single reservation can cover multiple travelers (1:N).
- Each **Booking** has exactly **one Payment**, but a **Payment** is linked to only one **Booking** (1:1).
- A **Booking** is associated with a **Flight**, but multiple **Bookings** can be linked to the same **Flight** (N:1).
- Each **Booking** is associated with a **Seat**, but multiple **Bookings** can be linked to the same **Seat** (N:1).
- Each **Flight** departs from and arrives at an **Airport** (N:1).
- Each **Flight** is operated by a single **Aircraft**, but one **Aircraft** may operate multiple **Flights** (N:1).
- A **Flight** contains multiple **Seats**, but each **Seat** is linked to only one **Flight** (1:N).
- Each **Flight** is associated with one **Airline**, but an **Airline** operates multiple **Flights** (N:1).
- Each **Flight** has multiple **Crew Members**, ensuring safe operation (1:N).
- A **Booking** may include multiple **Luggage items**, but each **Luggage item** is linked to only one **Booking** (1:N).
- A **Booking** has exactly one **Check-in record**, ensuring travelers complete the process (1:1).
- A **Passenger** has multiple **Boarding Passes**, as they may have multiple flights in an itinerary (1:N).
- A **Flight** has a single **Flight Status**, which updates in real-time (1:1).
- A **Flight** may have multiple **Baggage Claim points**, especially in large airports (1:N).

Business rule

1. **A user must register** with a valid email, phone number, and passport number before making a booking.
2. **Each booking must be linked to a valid seat on a flight** to prevent overbooking and seat duplication.
3. **A payment is required to confirm a booking**, and unpaid bookings may be automatically canceled after a certain time limit.
4. **A user can only book available seats**, ensuring that only vacant seats are assigned to passengers.
5. **Each flight must have at least one crew member assigned**, ensuring regulatory compliance and operational efficiency.
6. **Check-in must be completed before receiving a boarding pass**, ensuring that all passengers go through proper verification before boarding.
7. **Once a booking is confirmed, the seat is marked as reserved**, preventing multiple bookings for the same seat.
8. **Each booking may include multiple passengers**, ensuring that group reservations are managed efficiently.
9. **Passengers must check-in before their flight's departure time**, ensuring a smooth boarding process.
10. **Each flight has a real-time status update**, providing passengers with accurate information on delays and cancellations.
11. **A baggage claim area must be assigned for each flight**, ensuring proper baggage retrieval at the destination.
12. **A flight cannot be scheduled without a valid departure and arrival airport**, maintaining the validity of flight records.
13. **A passenger can hold multiple boarding passes if they have multiple flight segments**, ensuring smooth travel for multi-leg journeys.
14. **A crew member can only be assigned to one flight at a time**, preventing scheduling conflicts.
15. **Each luggage item must be assigned to a valid booking**, ensuring that baggage is properly tracked.

Entity-Relationship Diagram (ERD)

(On next page)


User
user_id (PK)
name
email
phone
password
passport number
makes
Booking
booking_id (PK)
user_id (FK)
flight_id (FK)
seat_id (FK)
booking_time
status
payment_id
requires
Checkin
checking_id (PK)
booking_id (FK)
seat_id (FK)
checkin_time
includes
Luggage
luggage_id(PK)
booking_id (FK)
weight
dimensions
baggage_belt_number
includes
Passenger
passenger_id (PK)
email
first name
last name
date of birth
BoardingPass
boarding pass id (PK)
passenger_id (FK)
flight_id (FK)
seat_id (FK)
boarding_gate
boarding_time
receives
processed by
Payment
payment_id (PK)
booking_id (FK)
payment_method
payment status
assigned to
Seat
seat_id (PK)
flight_id (FK)
class_type
price
books
Flight
flight_id (PK)
flight_number
departure_airport_id
arrival_airport_id
departure_time
arrival_time
aircraft_id (FK)
airline_id (FK)
has
FlightStatus
flight_status_report_id (PK)
flight_id (FK)
status
last_updated
has
operated by
Airline
airline_id (PK)
airline_name
country
Aircraft
aircraft_id (PK)
model
total_seats
airline_id (FK)
operated by
Crew
crew_id (PK)
flight_id (FK)
name
role
staffed by
departs from
Airport
airport_id (PK)
name
city
country
iata_code
icao_code

Relational Model (Schema Statements)

We have implemented normalization in our models and verified that our models are compatible with the 3NF standard.

1. User(_user_id_, name, email, phone, passport_number)
2. Passenger(_passenger_id_, @booking_id, name, passport_number, dob)
3. Airport(_airport_id_, airport_name, city, country, iata_code, icao_code)
4. Aircraft(_aircraft_id_, model, total_seats, @airline_id)
5. Flight(_flight_id_, flight_number, @departure_airport_id, @arrival_airport_id, departure_time, arrival_time, @aircraft_id, @airline_id)
6. Seat(_seat_id_, @flight_id, seat_number, class_type, price)
7. Booking(_booking_id_, @user_id, @flight_id, @seat_id, booking_time, status, @payment_id)
8. Payment(_payment_id_, @booking_id, payment_method, payment_status)
9. Airline(_airline_id_, airline_name, country)
10. Crew(_crew_id_, @flight_id, name, role)
11. Luggage(_luggage_id_, @booking_id, weight, dimensions)
12. Check-in(_checkin_id_, @booking_id, @seat_id, checkin_time)
13. BoardingPass(_boarding_pass_id_, @passenger_id, @flight_id, seat_number, boarding_gate, boarding_time)
14. FlightStatus(_flight_status_id_, @flight_id, status, last_updated)
15. BaggageClaim(_baggage_claim_id_, @airport_id, @flight_id, baggage_belt_number)

Assumptions and Justifications

1. **Each flight is associated with one aircraft**, ensuring accurate seat allocation and preventing conflicts in scheduling.
2. **Seats are uniquely identified by flight_id and seat_number**, preventing duplicate assignments and allowing efficient seat selection.
3. **Each booking is assigned a seat**, making it a **one-to-one** relationship, preventing overbooking of specific seats.
4. **Payments are required for booking confirmation**, ensuring revenue collection and preventing invalid reservations.
5. **Every flight must have a departure and arrival airport**, ensuring a valid flight route and avoiding incorrect scheduling.
6. **Normalization is applied to eliminate redundancy and maintain data integrity**, improving database efficiency and consistency.

7. **Each airline operates multiple flights**, enabling scalability and accommodating airline operations.
8. **Each flight must have at least one crew member assigned**, ensuring regulatory compliance and passenger safety.
9. **Passengers must check in before receiving a boarding pass**, ensuring security checks and flight readiness.
10. **Each flight has an updated status**, allowing passengers to receive real-time updates on delays, cancellations, and schedule changes.
11. **Each flight has a designated baggage claim area upon arrival**, ensuring baggage is properly sorted and reducing lost baggage incidents.
12. **A booking may include multiple passengers**, ensuring that families and groups can book under a single reservation.
13. **A flight may have multiple baggage claim points**, especially in large airports where multiple baggage belts are used for one flight.