Data Requirements

9.1 Data Models:

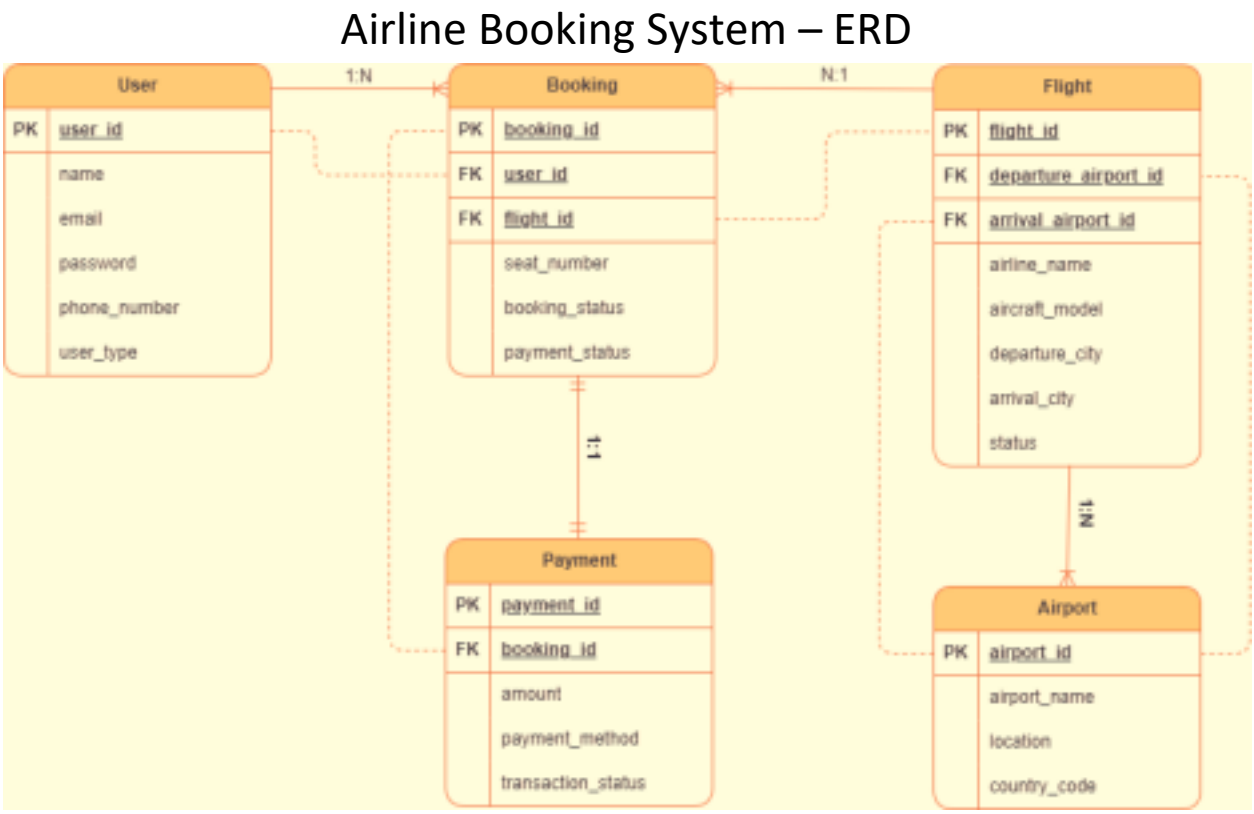

Figure 9.1.1

9.2 Database Requirements

Tables and Relationships

The database consists of the following

tables: Entities and Attributes

User

- user_id (PK)
- name
- email
- password
- phone_number
- user_type (Passenger/Admin)

Flight

- flight_id (PK)

- airline_name
- aircraft_model (New, replaces Aircraft entity)
- departure_city
- arrival_city
- departure_airport_id (FK)
- arrival_airport_id (FK)
- departure_time
- arrival_time
- status (On-time, Delayed, Cancelled)

Booking

- booking_id (PK)
- user_id (FK)
- flight_id (FK)
- seat_number
- booking_status (Confirmed, Pending, Cancelled)
- payment_status (Paid, Unpaid)

Payment

- payment_id (PK)
- booking_id (FK)
- amount
- payment_method (Credit Card, PayPal, etc.)
- transaction_status (Success, Failed)

Airport

- airport_id (PK)
- airport_name
- location
- country_code

Relationships

- User to Booking: 1:N → a user can have multiple bookings, but each booking belongs to one user.
- Flight to Booking: 1:N → a flight can have multiple bookings, but each booking belongs to one flight.
- Booking to Payment: 1:1 → each booking has one payment record.
- Flight to Airport: 1:N for departure_airport_id and 1:N for arrival_airport_id.

9.3 Data Storage and Retrieval

Data Storage

- The system uses a NoSQL database such as MongoDB to ensure flexible data management, scalability, and efficient document-based queries.
- Tables are designed with primary keys (PK) to uniquely identify records and foreign keys (FK) to maintain referential integrity between related tables.
- Indexes are applied to frequently queried columns (user_id, flight_id, booking_id) to improve search performance and optimize query execution.
- Data is stored in a normalized structure to minimize redundancy and enhance scalability.

Data Retrieval

User Queries

Users can:

- Search for flights by specifying departure and arrival cities, dates, and preferred airlines.
- View their booking history with details on past and upcoming flights.
- Check booking status (Confirmed, Pending, or Cancelled).

Admin Queries

Administrators can:

· Update flight statuses (On-time, Delayed, or Cancelled) to reflect real-time changes. · Manage payments and refunds, ensuring successful transactions and handling disputes.

Booking Process

1. Flight Selection: Users browse available flights based on search filters.
2. Seat Selection & Confirmation: Users choose a preferred seat and confirm the booking.
3. Payment Processing: The system records payment details and updates the booking status.
4. Final Confirmation: A successful payment updates the booking to “Confirmed,” while a failed transaction keeps it as “Pending.”

Optional

· Caching could be used to store frequently accessed flight schedules for faster lookups.· Stored Procedures could be implemented for complex queries like flight availability checks to reduce query execution time.

· Data Archiving policies can be set up to handle old booking records, keeping the database efficient.