YatraSathi

Business Requirements for Online Ticket Booking System (DBMS Project)

Members:-

Aayush Gupta - 2023014
Harsh Garg - 2023228
Shaman Ranjan - 2023498
Sanchit Maini - 2023480

Project Overview

The online ticket booking system will allow users to search, book, and manage tickets for different modes of transportation, including trains, buses, and flights. The platform should provide real-time seat availability, pricing, user authentication, payment processing, and cancellation/refund policies.

Business Requirements

1. **User Registration & Authentication**
    - Users should be able to sign up, log in, and manage their profiles.
    - Authentication should be secure (hashed passwords, OTP verification).
2. **Search & Filtering of Tickets**
    - Users should be able to search for available tickets based on source, destination, date, and transport type (train/bus/flight).
    - Filters should allow sorting by price, duration, departure time, etc.
3. **Seat Selection & Booking**
    - Users should be able to view seat maps and select specific seats.
    - The system should prevent overbooking by dynamically updating seat availability.
4. **Pricing & Discounts**
    - The system should manage different pricing models for peak and off-peak hours.
    - Promo codes and discounts should be applicable based on user type (student, senior citizen, etc.).
5. **Payment Gateway Integration**

    - Users should be able to pay via multiple payment methods (credit/debit cards, UPI, net banking, wallets).
    - Payment confirmation should trigger an email/SMS notification.
6. **Booking Confirmation & Ticket Generation**
    - After successful payment, the system should generate a unique PNR (Passenger Name Record).
    - Users should receive a digital ticket (PDF or QR code).
7. **Cancellation & Refunds**
    - Users should be able to cancel bookings based on predefined refund policies.
    - Partial refunds should be processed based on cancellation timing.
8. **Admin Dashboard**
    - Admins should be able to manage transport schedules, pricing, discounts, and user complaints.
    - Admins should have access to revenue reports and user statistics.
9. **Feedback & Customer Support**
    - Users should be able to provide ratings and feedback for their travel experience.
    - Live chat or email-based customer support should be available.

The admin details will be prior stored in the system, and only one admin is assigned to the website, who can login with the access privileges. The schedule of the transport including time, locations of departure and arrival, along with the costs can be updated only via the admin dashboard.

To use the website, every customer will have to create an account by giving the basic details of name, Phone number, age etc. Upon this, the info will be added to the system's database and a unique Customer ID will be assigned to him/her.

The customer can view the menu via the website and filter them by category . A person can view the number of tickets available for a particular trip, and can purchase them only if available.

After a person chooses the number of tickets, he/she will be needed to select a preferred one out of the available ones. Only once a person makes the payment, the available seats are changed in the db and subsequently in the frontend. Also, the seats are now shown as booked.

The booking is created and a unique booking ID is assigned along with a payment completion ID. This information can be viewed by the user in their account.

There can be multiple bookings (tickets) placed from a single customer ID itself. After this, the customer goes to the payment option in which he has the option to choose his payment mode. After the ticket confirmation is received, the customer can provide feedback regarding his booking experience. The feedback can further be viewed only by the admin.

The ticket prices have 3 slabs. The age group of 18-60 will fall under the base category, while the price will be 20% lesser for the kids and the senior citizens.

The status of the transport can only be one of the following:-

In journey / Completed / Cancelled / Not started

Furthermore, the prices can also change according to the demand and peak / non-peak hours. This can be changed only by the admin.

The system also should allow cancellation and refunds. The cancellation is done according to the following rule:-

<1day : 75% refund

<6 hr : 25% refund

<1 hr : Non refundable

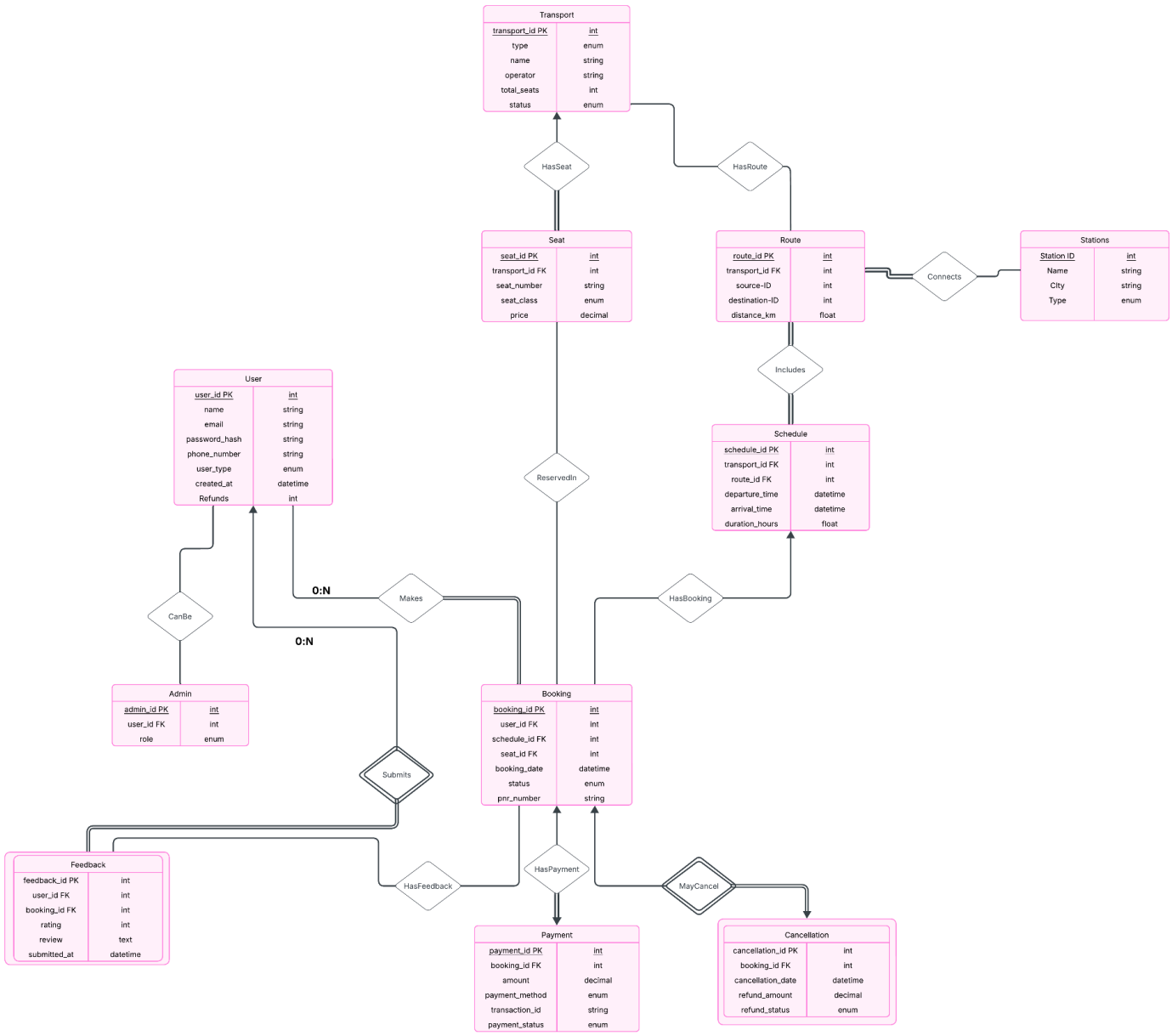
When a person cancels the ticket, the ticket count and the seat are made available again. Also the customer's record now shows Rs X in the "Refunds" section.

Why we used database for this project

A database is essential for **YatraSathi** as it enables structured data management, efficient querying, and secure storage of user profiles, bookings, payments, and transport schedules. It ensures data integrity through relationships and constraints while preventing double bookings via concurrency control. With SQL queries, users can quickly search, filter, and retrieve information. Additionally, authentication mechanisms like password hashing enhance security, and the database allows admins to manage transport schedules, pricing, and refunds dynamically. Overall, a database provides scalability, reliability, and seamless transaction handling for an efficient online ticket booking system.

ER Model

Link:
https://lucid.app/lucidchart/6837b133-ea9a-493b-b913-9ff8a42d4cd3/edit?viewport_loc=-67%2C-502%2C5493%2C2770%2C0_0&invitationId=inv_7576e2b1-cdcb-4fe9-be83-9b559759a579


Transport
transport_id PK int
type enum
name string
operator string
total_seats int
status enum
HasSeat
HasRoute
Seat
seat_id PK int
transport_id FK int
seat_number string
seat_class enum
price decimal
Route
route_id PK int
transport_id FK int
source-ID int
destination-ID int
distance_km float
Connects
Stations
Station ID int
Name string
City string
Type enum
Includes
Schedule
schedule_id PK int
transport_id FK int
route_id FK int
departure_time datetime
arrival_time datetime
duration_hours float
ReservedIn
User
user_id PK int
name string
email string
password_hash string
phone_number string
user_type enum
created_at datetime
Refunds int
CanBe
0:N
Makes
0:N
HasBooking
Admin
admin_id PK int
user_id FK int
role enum
Booking
booking_id PK int
user_id FK int
schedule_id FK int
seat_id FK int
booking_date datetime
status enum
pnr_number string
Submits
Feedback
feedback_id PK int
user_id FK int
booking_id FK int
rating int
review text
submitted_at datetime
HasFeedback
HasPayment
MayCancel
Payment
payment_id PK int
booking_id FK int
amount decimal
payment_method enum
transaction_id string
payment_status enum
Cancellation
cancellation_id PK int
booking_id FK int
cancellation_date datetime
refund_amount decimal
refund_status enum

REFERENCES USED

1. Lecture Slides
2. Class Notes
3. E-R slides from Sudarshan's book...