

Tour Booking System (YallaGo)

1. Overview

YallaGo is an online tour booking platform that allows users to browse, book, and manage travel experiences. The system provides a seamless way for travelers to explore destinations, avail offers, and make secure payments, while tour operators can manage tours and bookings efficiently.

2. Objectives

- Provide a user-friendly platform for travelers to book tours.
- Enable tour operators to manage tours, bookings, and discounts.
- Integrate secure payment processing for hassle-free transactions.
- Allow users to leave reviews and ratings for better decision-making.

3. Scope

- User registration, authentication, and profile management.
- Tour browsing, filtering, and booking.
- Payment processing and invoicing.
- Offers and discount management.
- Review and rating system.
- Notification system (email for bookings and offers).

➤ Stakeholder Analysis – Key Stakeholders & Their Needs

1. Travelers (Users):

- Search and book tours easily.
- View available offers and discounts.
- Make secure payments via **PayPal**.
- Receive booking confirmation and notifications.
- Leave reviews and ratings for tours.

2. Admin (Tour Operators):

- Add, update, and manage tours and destinations.
- Set offers and discounts on tours.
- Manage user bookings and cancellations.
- Track payments and booking status.

3. Payment Providers (PayPal):

- Process online payments securely.
- Ensure fraud detection and transaction security.

➤ User Stories & Use Cases

▪ User Stories

- **As a user**, I want to browse available tours based on location and date, so I can choose a suitable trip.
- **As a user**, I want to confirm my booking by making an online payment, so I can secure my spot.
- **As a user**, I want to manage my bookings, so I can organize my schedule.
- **As a user**, I want to track my past bookings, so I can review my travel history.
- **As an admin**, I want to add a new tour with detailed information, so I can attract more tourists.

- **As an admin**, I want to monitor tour reviews and ratings, so I can ensure high-quality services.
- **As an admin**, I want to manage user bookings and cancellations, so I can ensure smooth operations.
- **As an admin**, I want to set offers and discounts, so I can attract more customers.

➤ **Functional Requirements:**

1. User Management:

- User registration, login, and profile management.
- Admin role to manage tours, bookings, and offers.

2. Tour & Destination Management:

- View, search, and filter tours by destination, price, and date.
- Add, update, or remove tours and destinations.

3. Booking System:

- Select a tour and specify the number of people.
- Automatic seat availability check.
- Booking confirmation and cancellation options.

4. Payment Processing:

- Secure online payments (credit card, PayPal, etc.).
- Generate invoices and payment receipts.

5. Offers:

- Apply discount codes or promotions to tours.

6. Reviews:

- Users can leave ratings and comments for tours.

7. Notification System:

- Email notifications for booking confirmations, cancellations, and offers.

➤ Non-functional Requirements – System Quality Attributes

1. Performance:

- Fast response time (<3s per page).
- Scalable to handle high traffic loads.

2. Security:

- Secure authentication (hashed passwords, 2FA).
- Encrypted transactions for payments (SSL/TLS).

3. Usability:

- Intuitive UI for easy navigation.
- Mobile-friendly design.

4. Reliability:

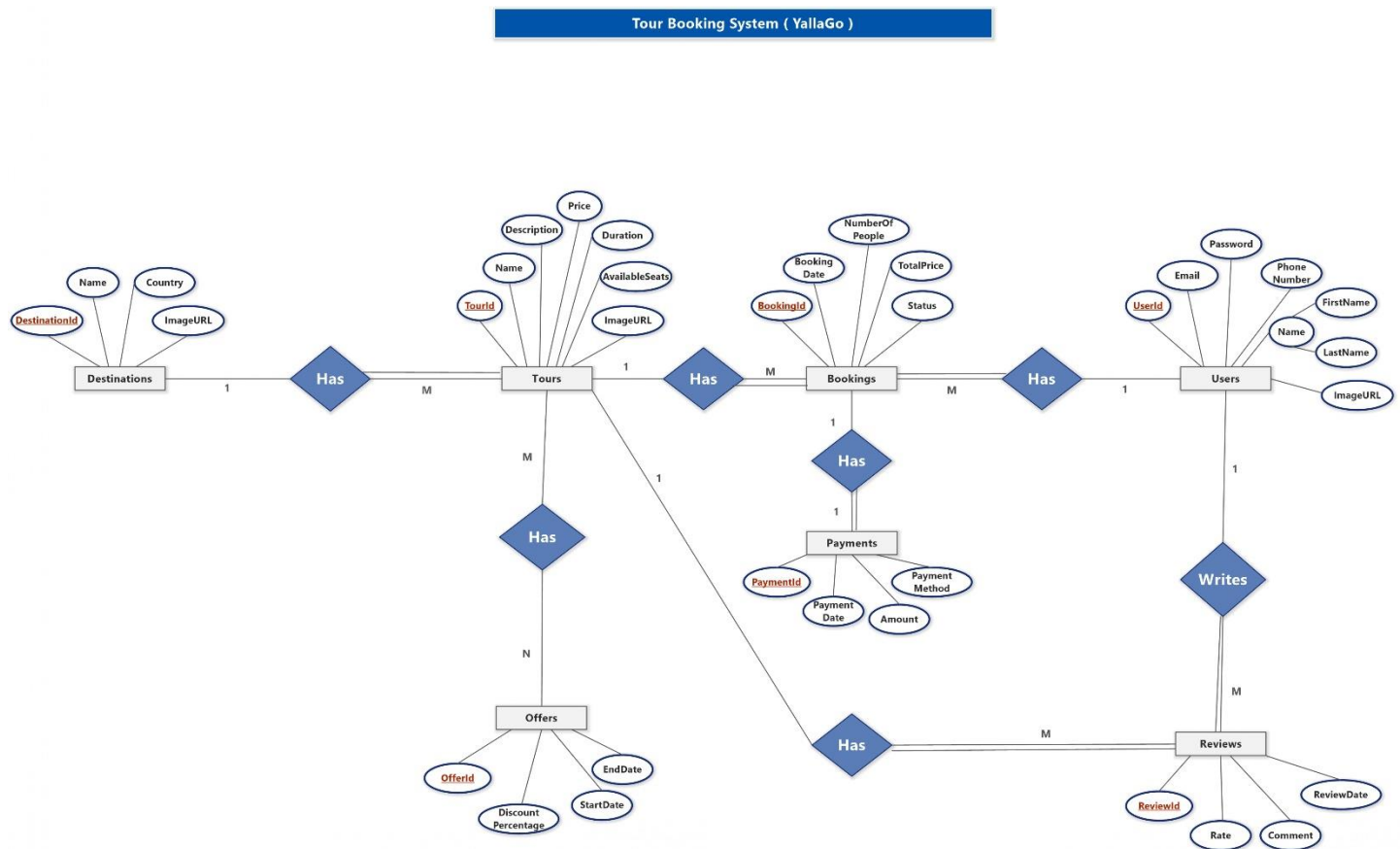
- 99.9% uptime for critical booking features.
- Automated backups to prevent data loss.

➤ Technology Stack:

- **Backend & Frontend:** ASP.NET Core MVC.
- **Database:** SQL Server with Entity Framework Core.
- **Authentication:** Identity with Cookie-based authentication.
- **Payment Integration:** PayPal API.
- **Styling & UI:** HTML/CSS, Bootstrap and jQuery.

➤ Database Design & Data Modeling

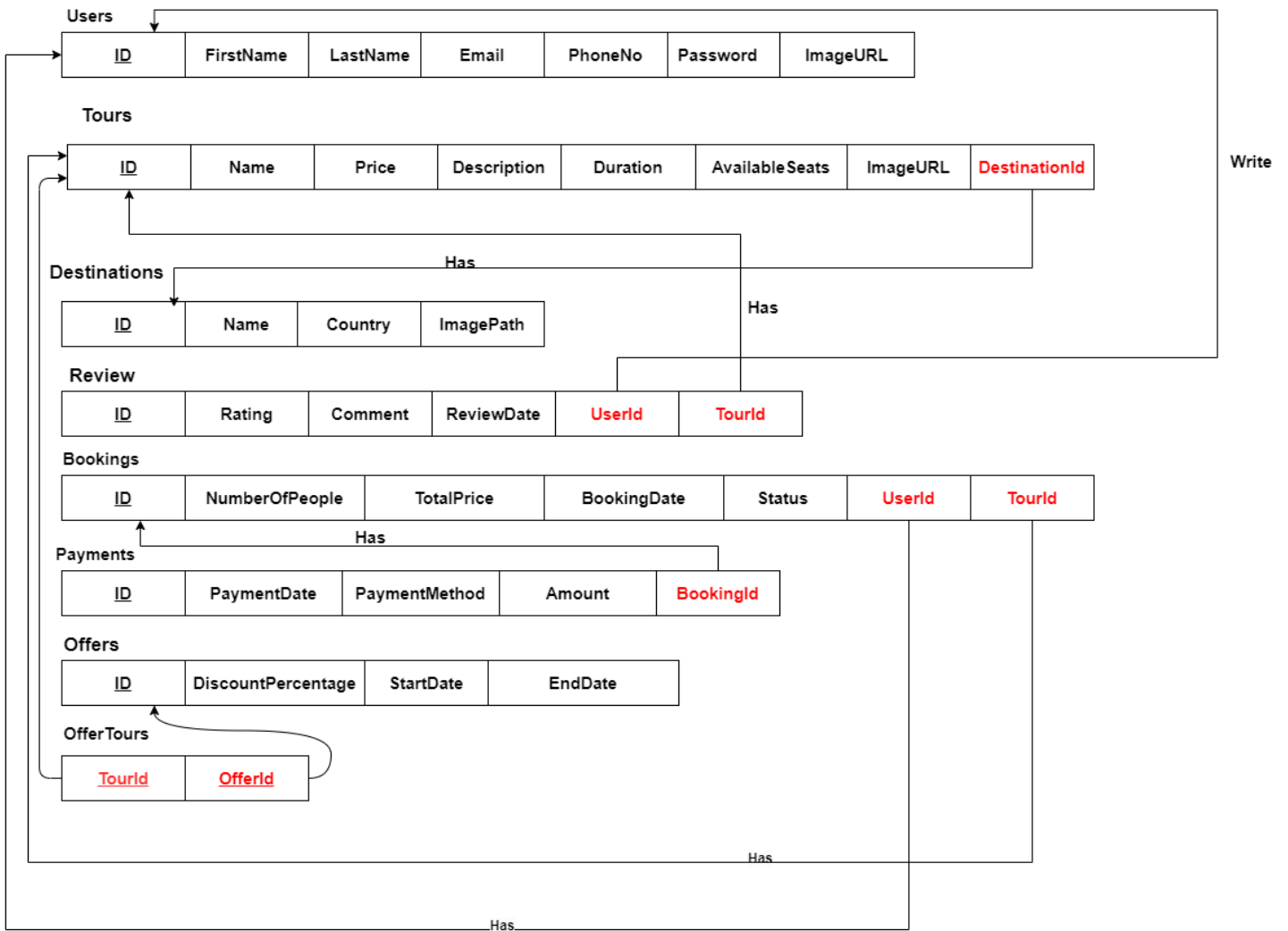
- **ER Diagram (Entity-Relationship Diagram)** The ERD visually represents the database structure, defining entities, attributes, and relationships. It includes key entities such as **Users, Tours, Destinations, Bookings, Payments, Offers, and Reviews**, with relationships ensuring data integrity and efficient retrieval.



- **Logical to Physical Mapping**

- **Database Schema Mapping:**

This section illustrates how the entity-relationship model is translated into database tables, including primary keys, foreign keys, and relationships.



➤ Activity diagram:

The diagram illustrates the **trip booking workflow**, detailing the steps a user follows from searching for trips to completing a successful payment.

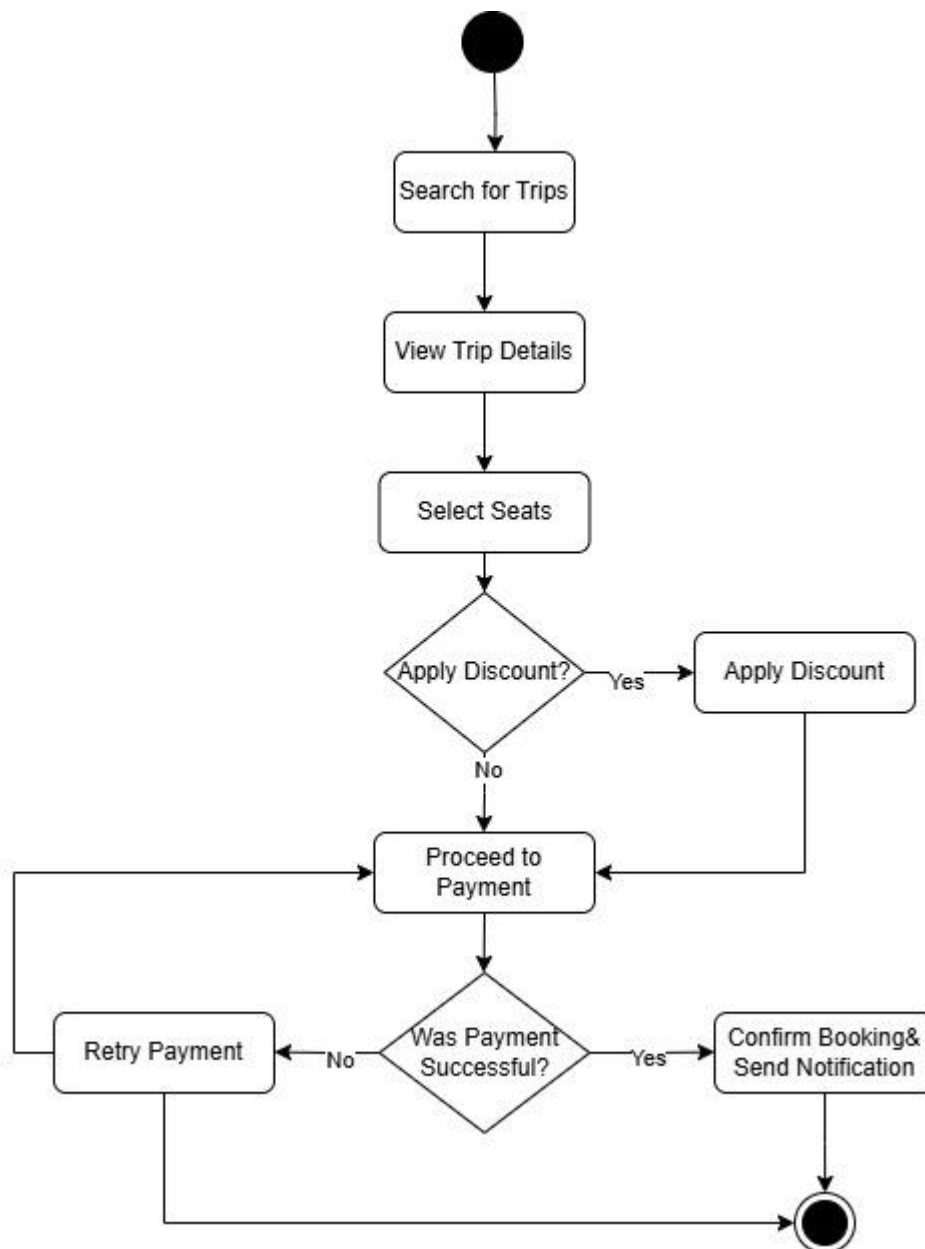
1. User Actions:

- Search for trips.
- View trip details.
- Select seats.
- Apply for a discount (if applicable).

2. Payment Process:

- If a discount is applied, it is processed before proceeding to payment.
- The user makes a payment.
- If the payment is successful, the booking is confirmed, and a notification is sent.
- If the payment fails, the user is prompted to retry.

This ensures a smooth and structured flow for **booking trips online** while handling potential payment failures.



➤ **Sequence Diagram:**

This sequence diagram illustrates the interactions between different components in the **YallaGo** system for **user registration, tour booking, payment processing, and admin management**.

▪ **Key Flows:**

1. User Registration & Authentication

- New user registers and their details are stored.
- Existing users log in, and credentials are validated.

2. Tour Search & Selection

- Users search for tours and apply filters (date, price, etc.).
- Filtered tours are retrieved and displayed.
- Users select a specific tour to view details.

3. Booking Process

- Users initiate a booking, and availability is checked.
- Once confirmed, booking details are stored.

4. Payment Processing

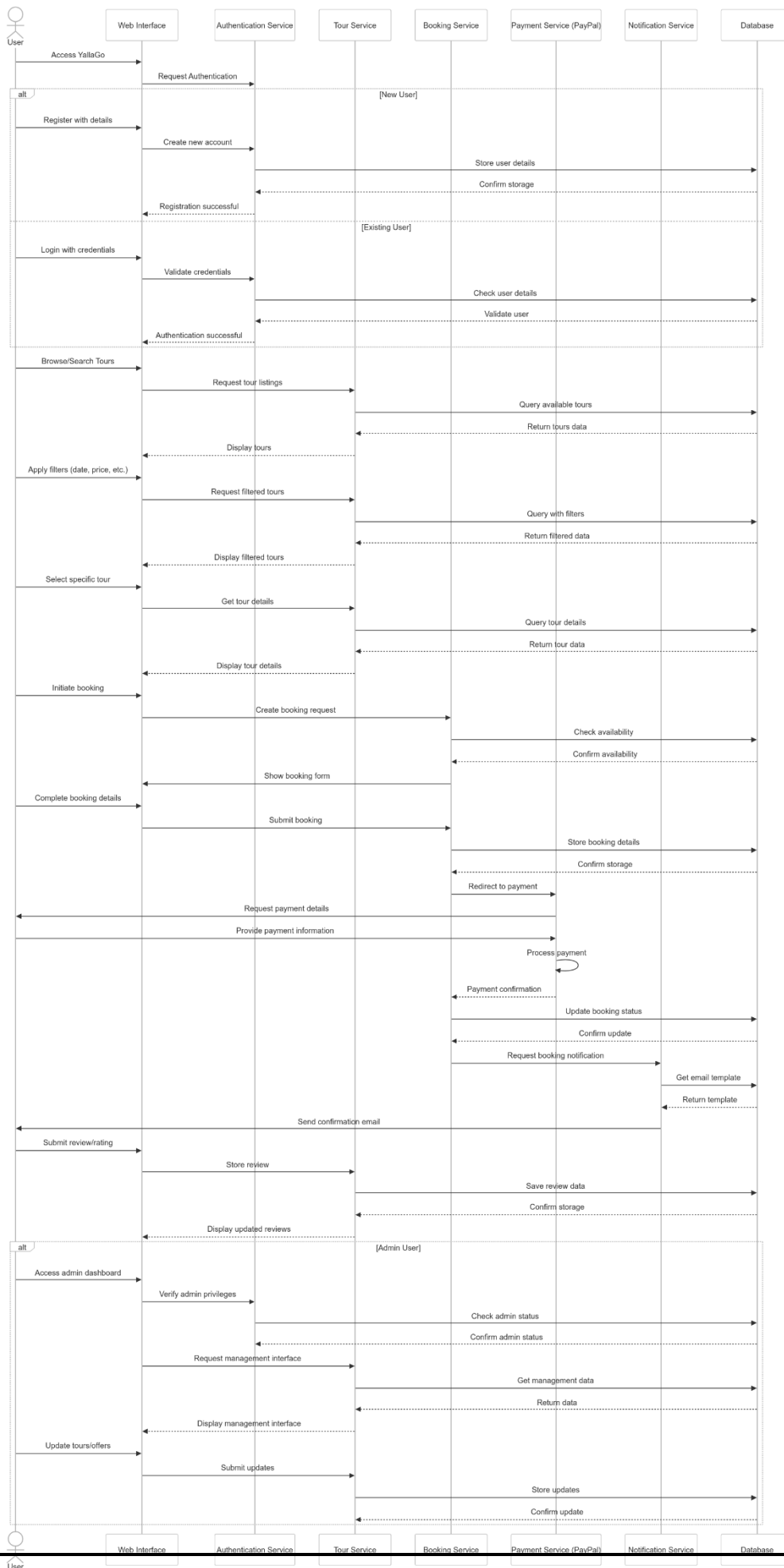
- Users provide payment details via PayPal.
- Upon successful payment, the booking status is updated, and a notification is sent.

5. Review Submission

- Users submit a review, which is stored and displayed.

6. Admin Management

- Admins log in, verify privileges, and manage tours/offers.
- Updates are stored and confirmed.



➤ Use Case Diagram:

1 Actors (System Users)

1. **User**– Searches for tours, books trips, and makes payments.
2. **Admin** – Adds and manages tours, monitors activities, and oversees content.
3. **Payment System** – Processes online transactions securely.

2 Use Cases

1. Book a Tour

- **Actors:** User, Payment System
- **Steps:**
 1. The user searches for available tours.
 2. Select a suitable tour.
 3. Enter payment details.
 4. Receives booking confirmation.

2. Add a New Tour

- **Actor:** Admin
- **Steps:**
 1. The admin logs into the system.
 2. Enter tour details (title, date, price, location, max capacity).
 3. Publishes the tour on the platform.

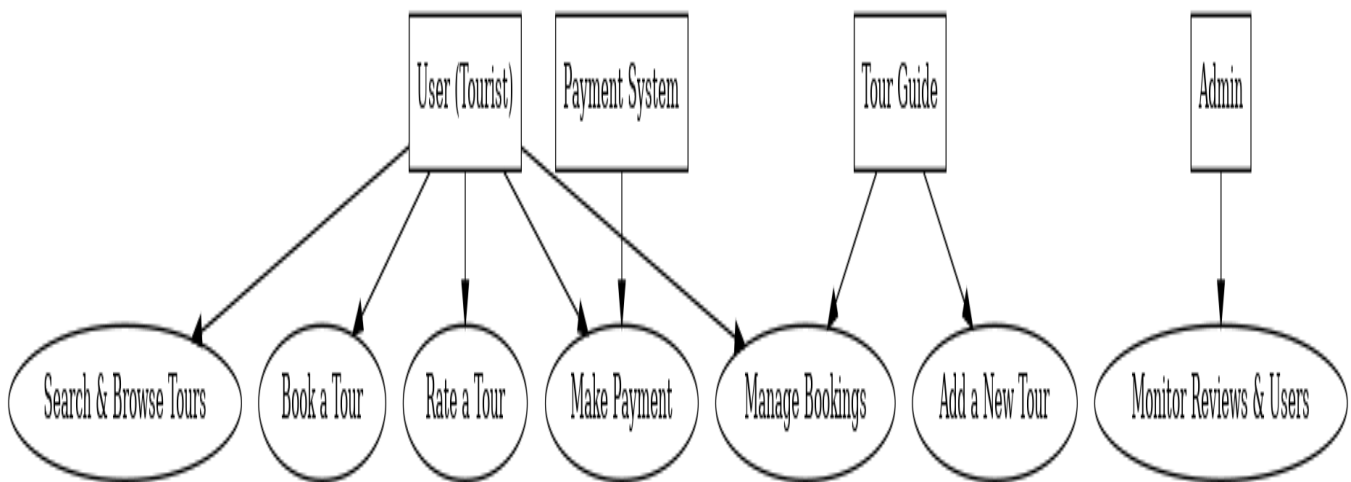
3. Manage Bookings

- **Actors:** User, Admin
- **Steps:**
 1. Users view their upcoming bookings.
 2. Users can cancel bookings before a specified deadline.

3. Admin can manage bookings and cancellations.

4. Rate a Tour

- **Actor:** User
- **Steps:**
 1. After completing a tour, the user gets a notification to rate the experience.
 2. The user submits a review and rating.
 3. The review appears on the tour's page for future users.



- **Technical company:** AST
- **Group code:** ONL2_SWD5_S2
- **Team members:**

Name	Phone Number
الزهراء عبد الله عرفة ذكي	01113531079
رقية علاء عبد الرافع وهدان	0 114 857 7738
سارة شعبان نصر الدين الشاعر	01205821041
مريم مصطفى محمود عبد العظيم	0109 033 8947