Hasta Travel Online Car Rental System - Requirement Analysis

Objective

To design a web-based car rental information system for Hasta Travel that automates the process of vehicle booking, fleet management, and customer service while supporting operational efficiency and strategic decision-making.

Functional Requirements

- User Registration & Authentication

Customers and staff can register and log in securely. Role-based access: admin, customer, operations staff.

- Vehicle Browsing & Search

Search vehicles by type, price, brand, availability, or location. Filter results (e.g., electric, SUV, manual/automatic).

- Online Booking & Payment

Real-time booking system with date/time selection. Integration with payment gateways (e.g., PayPal, Stripe).

- Rental Management

Track active, upcoming, and completed rentals. Set rules for late return, cancellations, and refunds.

- Fleet Management

Admin can add, update, or remove vehicle listings. Track vehicle condition, maintenance, availability.

- Customer Feedback & Support

Collect ratings and reviews after rentals. Provide chat or ticketing system for support.

- Report Generation

Generate usage statistics, financial summaries, and customer reports. Admin dashboard for KPIs.

- Discounts & Loyalty System

Apply promo codes and discounts. Track customer history and reward repeat customers.

- Notification System

Send emails/SMS for booking confirmations, reminders, or alerts.

- Location Services

Integrate Google Maps API to show car pickup/drop-off points.

Non-Functional Requirements

- Performance

Support at least 1,000 concurrent users with <3s page load time.

- Scalability

Must accommodate new features (e.g., car subscription plans) and more users.

- Security

Use HTTPS, input validation, and role-based access. Encrypt sensitive user and payment data (PCI DSS compliance).

- Usability

Mobile-first responsive UI. Simple, intuitive navigation for all age groups.

- Availability

Ensure 24/7 access with minimal downtime (<0.5% monthly).

- Maintainability

Use modular codebase (MVC architecture) for easy updates.

- Portability

Accessible via web browsers and mobile devices (Android/iOS).

- Interoperability

Integrate with external APIs (payment, map, vehicle tracking, CRM).

- Compliance

Adhere to local laws (e.g., driver's license validation, rental terms). GDPR compliance for user data privacy in applicable regions.

- Agility

Development should follow Agile methods (Scrum sprints, CI/CD pipelines).