



UE22CS341A : Software Engineering

SOFTWARE REQUIREMENTS SPECIFICATION (SRS) **REPORT**

TRAVEL AGENCY MANAGEMENT SYSTEM

Team:

Shreya M Hegde - PES1UG22CS573

Shreya Mittal - PES1UG22CS575

Introduction:

1. Purpose:

The purpose of this document is to provide an in-depth overview of the requirements and specifications for the Travel Agency Database Management System. This system is designed to automate and streamline the booking process for road trips, providing an easy interface for customers and improving the agency's management of bookings, vehicle inventory, payments, and driver assignments.

2. Scope:

The Travel Agency Database Management System will manage vehicle bookings by integrating customer details, agency information, vehicle availability, payment status, and driver assignments. The system will automate manual processes, ensuring data accuracy, preventing double bookings, and improving customer satisfaction.

3. Definitions, Acronyms, and Abbreviations:

- **TADMS:** Travel Agency Database Management System
- **DBMS:** Database Management System
- **CRUD:** Create, Read, Update, Delete

4. References:

- "Automated Travel Management Systems: A Comparative Study" - J. Smith, A. Brown
- "A Study on the Impact of Digital Transformation in Travel Agencies" - M. Johnson
- "Design and Implementation of Online Booking Systems for Travel Agencies" - L. Wang, R. Kumar
- "Customer Relationship Management in the Travel Industry: A Case Study" - P. Gonzalez, T. Evans
- "The Role of Technology in Modernizing Travel Agencies" - S. Lee

5. Overview:

This document outlines the requirements of the TADMS project, including its functionalities, user roles, and technical specifications. It covers both functional and non-functional requirements, as well as the system architecture and design approach.

System Overview:

The Travel Agency Database Management System is designed to automate the process of booking vehicles for road trips. The system will manage bookings by handling the interactions between the agency, customers, drivers, and vehicles. Payments for each booking will also be tracked, ensuring accurate payment statuses for both customers and the agency.

Overall Description:

1. Product Perspective:

The system serves as a stand-alone platform specifically tailored for travel agencies to manage vehicle bookings. It will provide seamless integration of different modules like car selection, booking management, driver assignment, and payment processing, along with customer management.

2. Product Functions:

- **Customer Management:** Allows customers to register and manage their profiles.

- **Booking Management:** Manages trip bookings and their associated data such as customers, drivers, vehicles, and payment.
- **Vehicle Management:** Keeps track of the vehicle inventory, including availability status and vehicle details.
- **Driver Management:** Manages drivers and their assignments for trips.
- **Payment Tracking:** Manages payment information and statuses related to bookings.

3. User Classes and Characteristics:

- **Administrator:** Manages vehicle inventory, driver assignments, and system configurations.
- **Customer:** Registers on the platform, browses vehicles, and makes bookings.
- **Driver:** Assigned to road trip bookings and manages their availability.

4. Operating Environment:

The system will be a web-based platform, running on modern web browsers (Chrome, Firefox, Safari) with a MySQL database on the backend.

5. Design and Implementation Constraints:

- The database will be implemented using MySQL.
- Frontend development will use React and JavaScript for responsiveness.
- The system must ensure real-time booking and availability status.

6. Assumptions and Dependencies:

- The users have a stable internet connection.
- Drivers and vehicles are assigned manually by the administrator or automatically based on availability.
- Customers can only book available vehicles.

Functional Requirements:

1. Customer Management:

- The system shall allow customers to register with their details (FirstName, LastName, Email, Phone, Address, DateOfBirth, Nationality).
- The system shall allow customers to view and manage their bookings.

2. Vehicle Management:

- The system shall allow the administrator to add vehicles with attributes such as VehicleType, LicensePlate, Capacity, Model, MakeYear, InsuranceNumber, and AvailabilityStatus.
- The system shall update vehicle availability after each booking or trip.

3. Booking Management:

- The system shall allow customers to select and book a vehicle for a specific date and time.
- The system shall prevent double bookings by checking availability before confirmation.

4. Driver Assignment:

- The system shall allow administrators to manage drivers' profiles (DriverID, FirstName, LastName, Phone, LicenseNumber, AssignedVehicleID, AvailabilityStatus).
- Drivers shall be automatically assigned to trips based on availability or manually assigned by the administrator.

5. Payment Management:

- The system shall allow customers to make payments using various payment methods (paymentMethod).
- The system shall track payment status (paymentStatus) for each booking (e.g., Paid, Pending).

6. Notification System:

- The system shall send booking confirmation to customers via email or SMS.

Non-Functional Requirements:

1. Performance Requirements:

- The system should respond to booking requests within 2 seconds.
- The system should handle up to 1,000 concurrent users.

2. Security Requirements:

- The system must implement user authentication (password encryption).
- Customer data, including payment information, should be securely stored and protected against breaches.

3. Usability Requirements:

- The user interface should be simple and intuitive for both customers and administrators.
- Customers should be able to complete bookings with minimal steps.

4. Reliability Requirements:

- The system should have an uptime of 99.9%.
- Data recovery processes should ensure minimal downtime in case of failures.

5. Scalability:

- The system must be scalable to handle increased loads by adding more servers as needed.

System Models:

1. ER Diagram:

A diagram showing relationships between key entities:

- Agency (AgencyID, AgencyName, Address, Phone, etc.)
- Booking (BookingID, CustomerID, VehicleID, TripDate, etc.)
- Vehicle (VehicleID, Type, LicensePlate, Capacity, etc.)
- Customer (CustomerID, Name, Email, Phone, etc.)
- Driver (DriverID, Name, Phone, LicenseNumber, etc.)
- Payment (PaymentID, PaymentStatus, Amt, etc)

2. Data Flow Diagrams (DFD):

- Customer selects a vehicle -> System checks availability -> Booking is confirmed.
- Administrator adds a vehicle -> System updates the inventory.

3. Use Case Diagrams: Use cases for customer registration, vehicle booking, payment processing, and driver assignment.

System Interfaces:

1. User Interfaces:

- **Customer Interface:** Web interface allowing customers to browse vehicles, make bookings, and complete payments.
- **Administrator Interface:** Dashboard for managing vehicles, bookings, payments, and driver assignments.

2. Hardware Interfaces: No specific hardware interfaces are required as the system is web-based.

3. Software Interfaces:

- The backend will use MySQL for database management.
- Integration with payment gateways for handling customer payments.

4. Communication Interfaces: The system will use HTTP/HTTPS protocols for web-based communications.

Database Design:

1. ER Model:

Entities and their attributes:

- **Agency** (AgencyID, AgencyName, Address, Phone, Email, LicenseNumber, EstablishedDate)
- **Booking/Trip** (BookingID, CustomerID, AgencyID, VehicleID, DriverID, TripDate, BookingDate, StartLocation, EndLocation, TotalCost, PaymentStatus)

- **Vehicle** (VehicleID, VehicleType, LicensePlate, Capacity, Model, AvailabilityStatus)
- **Customer** (CustomerID, FirstName, LastName, Email, Phone, Address, DateOfBirth, Nationality)
- **Driver** (DriverID, FirstName, LastName, Phone, LicenseNumber, ExperienceYears, AssignedVehicleID, AvailabilityStatus)
- **Payment** (PaymentID, Amt, PaymentStatus, PaymentMethod, Dues, PaymentDate)

2. Relational Schema:

The schema defines the tables and their relationships:

- **Agency Table:** Stores agency details.
- **Booking Table:** Stores trip booking details.
- **Vehicle Table:** Contains vehicle information.
- **Customer Table:** Stores customer details.
- **Driver Table:** Stores driver profiles.
- **Payment Table:** Stores payment transactions.

3. CRUD Operations:

- **Create:** Register customers, book trips, add vehicles, assign drivers.
- **Read:** Retrieve available vehicles, booking details, payment status.
- **Update:** Modify booking details, vehicle availability, payment status.
- **Delete:** Remove bookings, cancelled payments.

System Evolution:

1. Potential Extensions:

- Integration with third-party travel services (e.g., hotels, flight bookings).
- Addition of luxury vehicles and VIP services.

2. **Adaptability:** The system can be extended to support more advanced features like loyalty programs, real-time GPS tracking, or more complex trip packages.

Glossary:

- **Booking:** The process of reserving a vehicle for a road trip.
- **Driver Assignment:** Allocating a driver to a confirmed booking.
- **Payment Status:** Indicates whether a booking has been paid or is still pending.

Appendices:

- Additional system diagrams (ER Diagram, Data Dictionary, Schema Diagram, CRUD Matrix)