Car Washing Management System

Car Washing Management System Introduction

The Car Washing Management System is a digital platform designed to simplify car wash bookings and service tracking. It enables users to book appointments online and view their service history easily. For car wash owners, the system offers efficient management of customers and services.

This platform aims to improve service efficiency and enhance the overall customer experience by allowing admins to manage bookings, customers, and service history seamlessly.

User Roles and Functional Overview

Customers (Users)

Book car wash slots, and view service history.

Employees

Complete or cancel services, search bookings by customer or ID, and track their service history.

Admins

Manage bookings, register users and employees, update details, and oversee service statuses.

User Functional Requirements

Secure Registration and Login

Users can register and log in using JWT authentication for secure access.

Booking Services

Users can book car wash slots by selecting date, time, and service type.

View Booking History

Users have access to their previous booking records for reference.

Employee Functional Requirements

Service Completion and Cancellation

Employees can complete or cancel scheduled car wash services as needed.

Search and Filter Bookings

Ability to search by customer name or booking ID and filter by service status such as requested, scheduled, completed, or canceled.

View Personal Service History

Employees can track the number of services they have completed for performance monitoring.

Admin Functional Requirements

Booking and User Management

Admins can schedule bookings, register new users and employees, and update their details.

Service Control

Admins have authority to complete or cancel scheduled services.

Advanced Search

Capability to search by customer name, booking ID, or service status.

Non-Functional Requirements: Security and Usability

Security

- JWT authentication for secure, stateless sessions
- Encryption of sensitive data like passwords and tokens
- Role-based access control to restrict unauthorized actions

Usability

- Clean, responsive, and user-friendly desktop interface
- error messages guide users effectively

Technology Stack

Frontend

React.js for building dynamic and responsive user interfaces.

Backend

Java with Spring Boot microservices architecture and Spring Security for JWT-based authentication.

Database

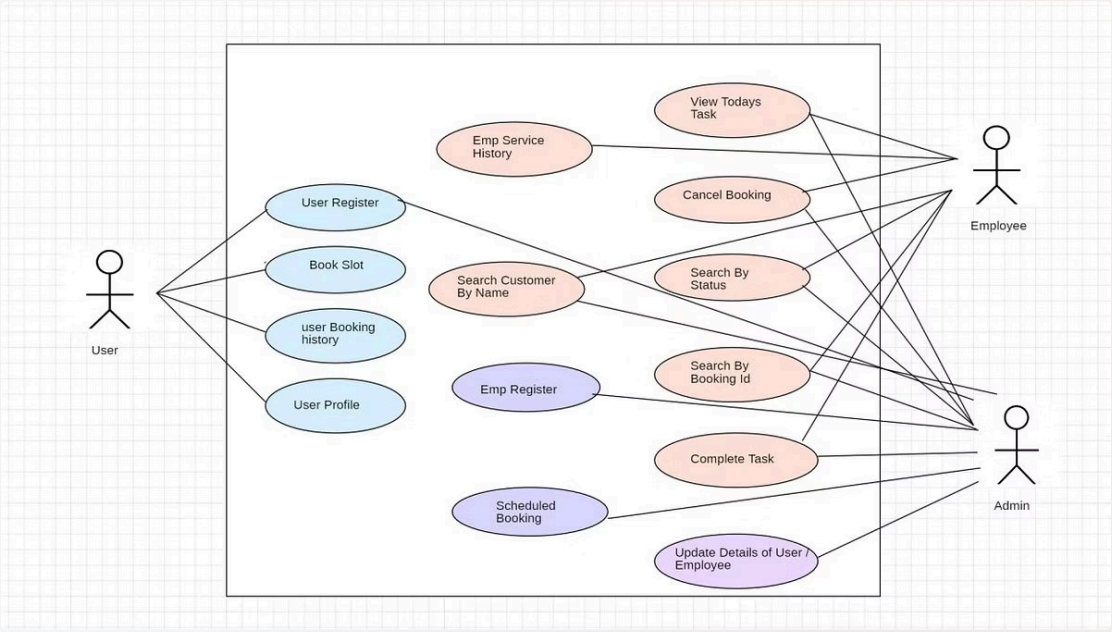
MySQL for data storage.

Testing

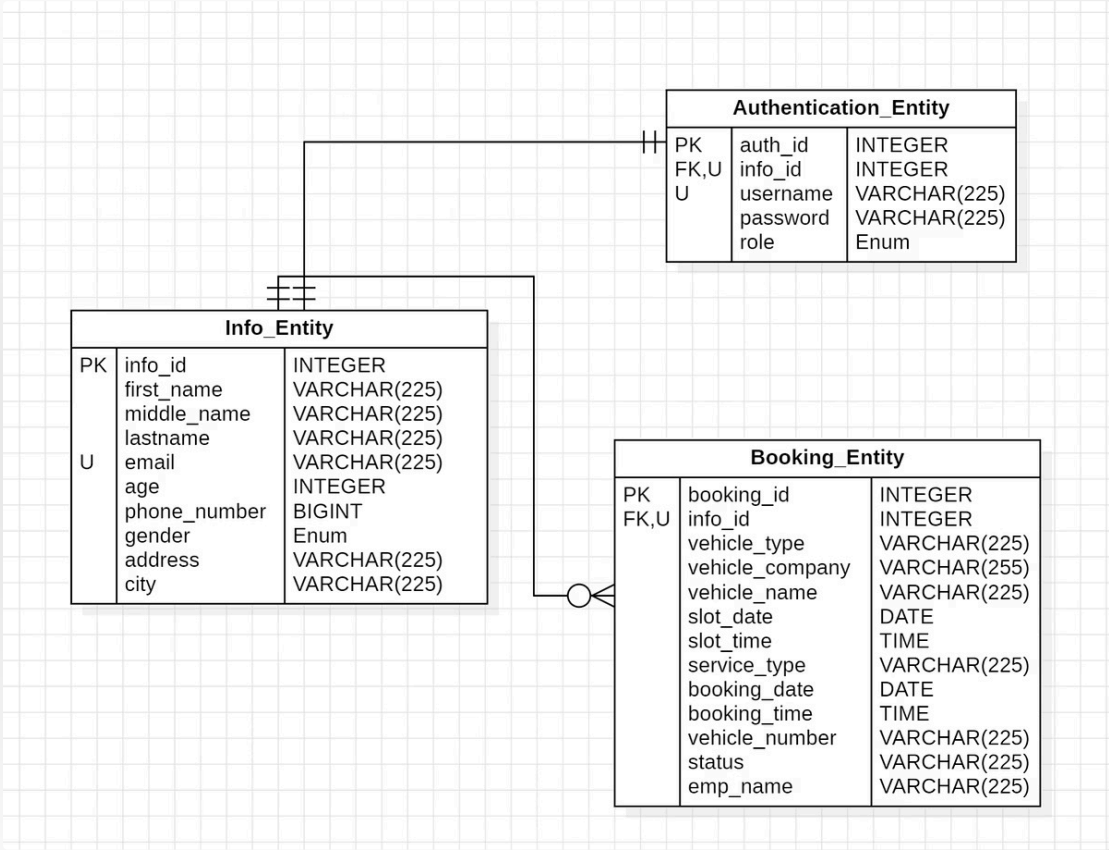
API Documentation

Swagger used for API documentation and testing.

Use Case Diagram



Data Base Diagram



Thank You