UTILITY VEHICLE RENTAL SYSTEM

17.12.2023

IRADUKUNDA PACIFIQUE

Overview

The Utility Vehicle Rental System (UVRS) is a comprehensive platform designed to facilitate the seamless rental of utility vehicles. It serves as a bridge between vehicle owners and individuals or businesses in need of utility vehicles for specific tasks or projects. UVRS provides an online marketplace where users can register their utility vehicles, search for available options, and efficiently engage in the rental process. The system aims to streamline the entire lifecycle of vehicle rental, from registration to booking and payment, fostering a user-friendly and efficient experience.

Goals

1. Efficient Vehicle Rental Process:
    a. Enable vehicle owners to easily register their utility vehicles on the platform.
    b. Provide a user-friendly interface for renters to search, filter, and select suitable vehicles based on their specific needs.
2. User Authentication and Security:
    a. Implement a robust user authentication system to ensure the security and privacy of user information.
    b. Employ secure and encrypted payment methods to safeguard financial transactions.
3. Scalability and Performance:
    a. Design the system to be scalable, accommodating a growing number of users and vehicles.
    b. Optimize performance to ensure quick response times and a smooth user experience.
4. Intuitive User Interface:
    a. Create an intuitive and visually appealing user interface for both vehicle owners and renters.
    b. Prioritize a user-centric design that enhances the overall usability of the UVRS platform.
5. Documentation and Support:

a. Develop comprehensive user documentation to guide users through the registration, rental, and payment processes.

b. Establish a customer support system to address queries, issues, and provide assistance when needed.

Project Requirements

1. Introduction:

The Utility Vehicle Rental System (UVRS) is an online platform connecting utility vehicle owners with individuals/businesses seeking to rent such vehicles.

2. Features:

- User Registration and Authentication:
  - Vehicle owners can create accounts.
  - Users can log in using secure authentication methods (e.g., email/password, JWT).
- Vehicle Registration:
  - Vehicle owners can register their utility vehicles.
  - Include details such as make, model, year, photos, and rental pricing.
- Search and Filtering:
  - Renters can search for available vehicles based on criteria (location, type, availability).
  - Implement advanced filtering options.
- Performance:
  - Ensure system responsiveness even during peak usage.
  - Loading times for pages and search results should be optimized.
- Security:
  - Implement secure data transmission (SSL).
  - Protect user data through encryption and secure storage.
- Compatibility:
  - Ensure compatibility with major web browsers (Chrome, Firefox, Safari).

    - Responsive design for various devices (desktop, tablet, mobile).
- Scalability:
    - Design the system to handle a growing number of users and vehicles.
    - Optimize database queries for scalability.

4. User Roles:

- Vehicle Owner:
    - Can register, manage, and update vehicle listings.
    - Receive and respond to booking requests.

5. Use Cases:

- User Registration:
    - New users can create accounts with valid information.
- Vehicle Registration:
    - Owners can add details and images of their vehicles.
- Booking Process:
    - Renters can search for vehicles and complete the booking process.

6. Constraints:

- Technology:
    - Use Spring Boot for the backend and React for the frontend.
- Time:
    - Development budget and timeline constraints.

Project Plan

1. Project Overview:

- Objective: Develop and launch the Utility Vehicle Rental System (UVRS) platform.
- Scope: Registration, vehicle listing, search, filtering, sorting, and authorization

2. Timeline:

- Phase 1: Planning and Design (1 day)
  - Define detailed requirements.
  - Finalize UI/UX design.
  - Establish the technology stack.
- Phase 2: Backend Development (1/2 weeks)
  - Set up Spring Boot backend.
  - Implement user registration and authentication.
  - Develop vehicle registration and listing features.
- Phase 3: Frontend Development (4 weeks)
  - Create React web app.
  - Implement user interfaces for registration, search, and booking.

Database Schema

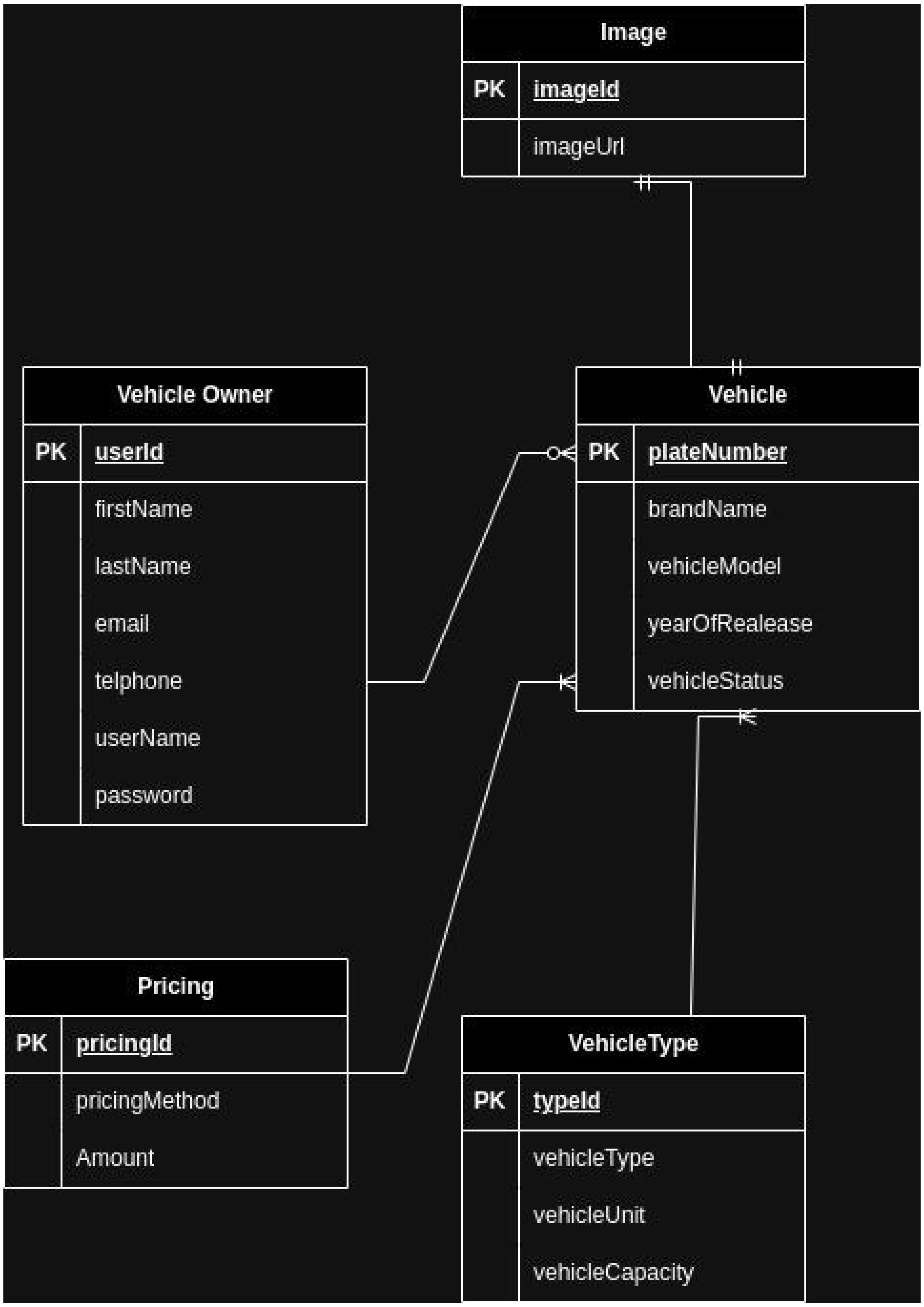

Image	
PK	imageId
	imageUrl

Vehicle Owner	
PK	userId
	firstName
	lastName
	email
	telphone
	userName
	password

Vehicle	
PK	plateNumber
	brandName
	vehicleModel
	yearOfRealease
	vehicleStatus

Pricing	
PK	pricingId
	pricingMethod
	Amount

VehicleType	
PK	typeId
	vehicleType
	vehicleUnit
	vehicleCapacity

User Documentation

Welcome to UVRS!

Thank you for choosing the Utility Vehicle Rental System (UVRS) for your utility vehicle rental needs. This user guide will provide you with step-by-step instructions on how to make the most of the UVRS platform.

Table of Contents:

---

1. Getting Started

1.1 Creating an Account

To begin using UVRS, follow these simple steps:

Navigate to the UVRS website.

Click on the "Sign Up"button.

Fill in the required information, including your email address and a secure password.

1.2 Logging In

Already have an account? Here's how to log in:

Visit the UVRS website.

Click on the "Log In" button.

Enter your registered email address and password.

Click "Log In" to access your account.

---

2. Vehicle Rental Process

2.1 Vehicle Search

Looking for the perfect utility vehicle? Follow these steps:

Visit the UVRS website.

Click on the "Search" or "Find Vehicles" option.

Use filters to narrow down your search based on location, type, and availability.

Browse through the available vehicles and click on a listing for more details.

Technical Documentation

1. System Architecture:

1.1 Overview:

The UVRS system follows a microservices architecture where frontend and backend components communicate through RESTful APIs. The components include the React-based frontend, Spring Boot-based backend services, and a relational database.

1.2 Components:

- Frontend:
  - Developed using React.js.
- Backend:
  - Built on the Spring Boot framework.
  - Implements a set of RESTful APIs.
- Database:
  - MySQL is used as the relational database.

1.3 Communication:

The frontend communicates with the backend through RESTful APIs. Data flows between components using HTTP.

2. Backend (Spring Boot):

2.1 Technology Stack:

- Java 17
- Spring Boot 1.5.6
- Spring Data JPA
- Spring security
- JWT for authentication

2.2 API Endpoints:

The backend exposes various endpoints, including:

- /api/users for user-related operations.
- /api/vehicles for vehicle-related operations.

2.3 Authentication:

User authentication is implemented using JWT tokens. Upon successful login, the backend issues a token that is sent with subsequent requests for authorization.

2.4 Database Schema:

The database schema consists of entities such as User, Vehicle, and Pricing, Relationships are established to maintain data integrity.

2.5 Pagination, Filtering, Sorting:

APIs support pagination, filtering, and sorting through query parameters.

3. Frontend (React):

3.1 Technology Stack:

- React.js
- Axios for API communication
- Bootstrap

3.2 User Interfaces:

The frontend provides user interfaces for:

- User registration and login.
- Vehicle search and listing.
- Booking and reservation processes.