**KPIT Technologies Limited**



**SYNOPSIS**

**ON**

**“Vehicle Availability System”**

**GLA University , Mathura**

Submitted By : Raj Gupta, Pragyan Prakhar, Sumit Mandal

**Title of the Project:**

**Vehicle Availability System –** A Web-Based Application for Managing Vehicle Status and Accessibility

**Objective:**

The Vehicle Availability System aims to provide a secure, scalable, and user-friendly web platform for real-time tracking and management of vehicles. It features role-based access control to restrict critical operations like adding or updating vehicles to authorized users such as dealers or admins. Built using Core Java, Hibernate, and PostgreSQL, it ensures efficient data handling, while a ReactJS frontend delivers a smooth user experience. The system also incorporates session management and validation to maintain data integrity and prevent unauthorized access.

**Scope:**

TheVehicle Availability System application will encompass the following functionalities:

* **User Authentication**: Secure signup, login, and logout using HTTP sessions to manage user state and protect access..
* **Role-Based Access Control:** · Restricts operations like adding, updating, or removing vehicles to authorized users such as dealers and admins.
* **User Portal**:Allows users to view real-time vehicle availability and search based on specific criteria.
* **Admin/Dealer Portal:** Enables authorized personnel to manage vehicle listings, update availability status, and perform administrative tasks.
* **Vehicle Management:Add, update, and delete vehicle details including model, price, fuel type, and mileage.**

**Methodology:**

The project will utilize the MERN stack:

* **Frontend**: React.js is used to build a responsive and interactive user interface for users and dealers.
* **Backend**: Core Java with Spring MVC architecture handles business logic and defines RESTful APIs for communication between frontend and backend.
* **Database**: PostgreSQL is used to store and manage vehicle details, user information, and availability status.
* **ORM**: Hibernate is implemented for efficient database operations and to map Java objects to relational tables.
* **Authentication:** HTTP Sessions are used to manage user login sessions securely, with role-based access control to distinguish between regular users, dealers, and admins.

**Proposed System:**

The Vehicle Availability System will provide an intuitive interface for users to view real-time availability of vehicles and explore vehicle details. Dealers and admins will have access to a secure portal where they can add, update, or remove vehicle entries. The system ensures efficient vehicle tracking and management, role-based access control, and seamless data handling through a secure backend and responsive frontend interface.

**Features:**

· **Secure User Authentication**: Login and signup system using HTTP sessions with role-based access control.

· **Role-Based Dashboard**: Separate interfaces for dealers and admins to manage vehicle data.

· **Vehicle Management**: Add, update, and delete vehicle information including model, price, and availability.

· **Real-Time Availability**: Display live status of vehicles to users for efficient decision-making.

· **Database Integration:** Persistent storage and retrieval using PostgreSQL with Hibernate ORM.

· **RESTful APIs**: Robust backend developed in Core Java exposing secure and scalable APIs.

· **Responsive Frontend**: User-friendly interface built with React.js, compatible across devices.

· **Data Validation & Security**:Input validation and session management to prevent unauthorized access.

**Implementation Plan:**

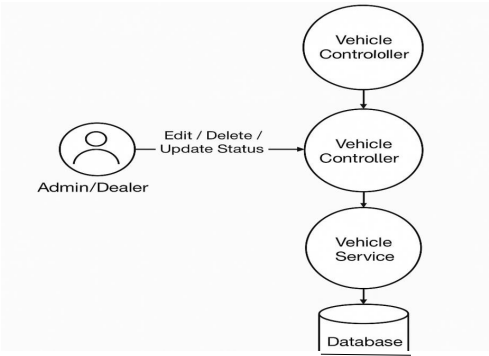
· **Planning Phase**:

* Defined core project requirements such as vehicle listing, user roles (dealer/user), and vehicle availability status.
* Designed wireframes and user interface layouts for login/signup, vehicle view, and dealer dashboard.

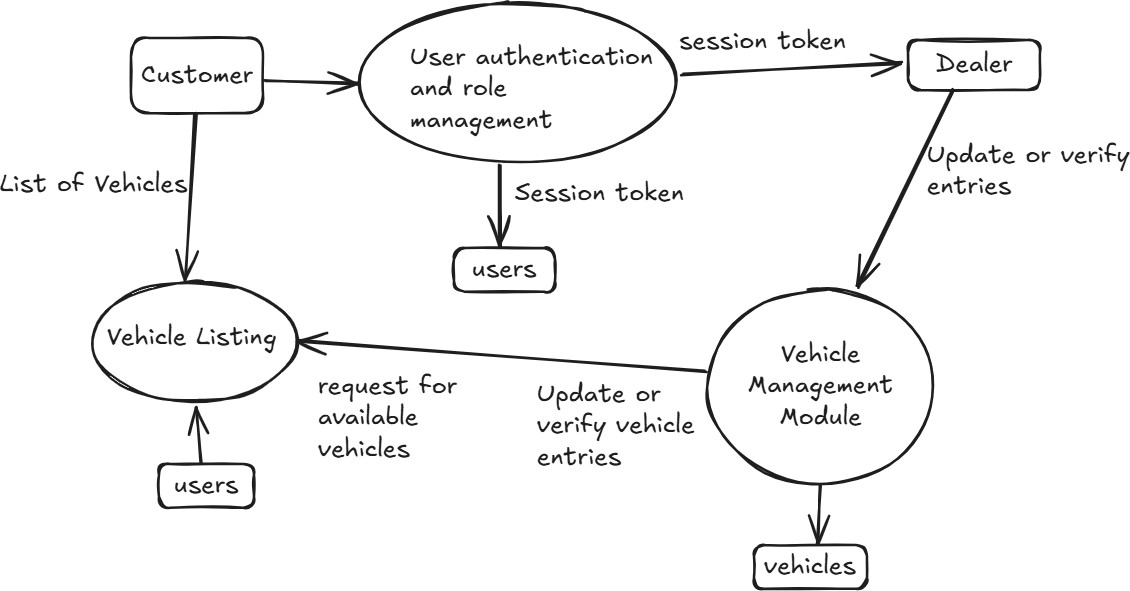
· **Development Phase**:

* Set up a Java-based backend using Core Java, Hibernate (JPA), PostgreSQL, and Maven.
* Implemented user authentication and role-based access control using HTTP sessions.
* Developed REST APIs for vehicle listing, login/signup, and user management.
* Integrated the backend with a React-based frontend

**Collaboration Diagram :** This Collaboration Diagram shows how Admin/Dealer interacts through UI to edit, delete, or update vehicle status, and how the backend layers (controller → service→ DAO) communicate to perform those actions in the PostgreSQL database.



**Data Flow Diagram:**



**Team Member:**

|  |  |
| --- | --- |
| NAME | SUPERSET ID |
| Raj Gupta | 5384344 |
| Pragyan Prakhar | 5468143 |
| Sumit Mandal | 5387731 |

#### Resources Required:

* **Software**: Visual Studio Code, PostgreSQL, PgAdmin, Apache Maven, Hibernate ORM, Postman, React.js, Git, GitHub
* **Hardware**: Personal computers with stable internet connection.

**References:**

• JAVA SE Documentation: <https://docs.oracle.com/en/java/>

• Hibernate ORM Documentation: <https://hibernate.org/orm/documentation/>

• PostgreSQL Documentation: https://[www.postgresql.org/docs/](http://www.postgresql.org/docs/)

• Maven Documentation: <https://maven.apache.org/guides/index.html>

• React Documentation: [https://reactjs.org/docs/getting-started.html](https://reactjs.org/docs/getting-started.html" \t "_new)

**Expected Outcomes:**

The Vehicle Availability System seeks to deliver a reliable and efficient platform for managing and viewing vehicle availability, aligned with the specified goals and scope of the project. Users will benefit from secure login and role-based access, an easy-to-use interface for dealers to manage vehicle listings, and a seamless experience for end-users to browse available vehicles.

**Conclusion:** The Vehicle Availability System is a streamlined solution for managing vehicle data using Core Java, Hibernate, and PostgreSQL. With role-based access for dealers and users, it ensures secure and efficient inventory control. The system simplifies vehicle management and lays a strong foundation for future enhancements like booking, feedback, and cloud deployment.