

Car Rental Management System – Synopsis

1. Introduction to the Project

The Car Rental Management System is a microservices-based solution designed to streamline vehicle rental operations. It provides an efficient way to manage vehicle bookings, track transactions, monitor vehicle kilometres, and handle customer data. The system ensures smooth interactions between customers and rental agencies by automating processes and reducing manual intervention.

2. Problem Statement

Traditional car rental services rely on manual record-keeping, leading to inefficiencies, booking conflicts, and a lack of real-time vehicle availability. Additionally, tracking vehicle usage and handling transactions manually can result in errors and delays. This project aims to develop a digital system that addresses these challenges by implementing a well-structured, normalized database and an intuitive user interface.

3. Objectives

- To develop a microservices-based architecture for managing car rentals efficiently.
- To enable seamless booking and transaction processing.
- To implement vehicle tracking for better maintenance and utilization.
- To maintain a database of customers and provide loyalty-based discounts.
- To ensure data consistency and avoid redundancy through proper normalization.

4. Scope of the Project

The Car Rental Management System will cater to the following functionalities:

- **Vehicle Management:** Adding, updating, and tracking vehicle details.
- **Booking Management:** Enabling customers to book vehicles with specific rental durations.
- **Transaction Management:** Handling payments, invoices, and refunds.
- **Vehicle Kilometers Tracking:** Monitoring distance traveled for maintenance purposes.
- **Customer Management:** Storing customer details and tracking rental history.
- **Owner Management:** Offering cars for rental purposes.

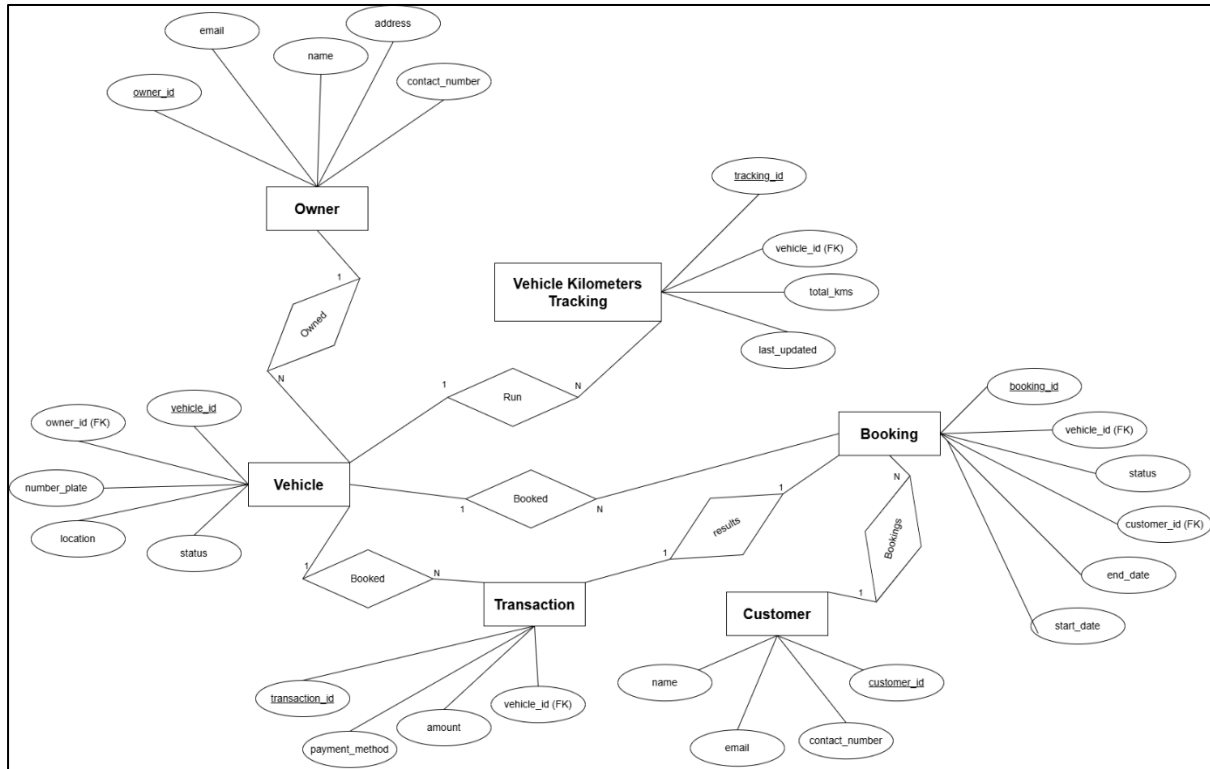
5. Technologies Used

- **Backend:** Node.js / Spring Boot (Java)
- **Frontend:** React.js / Angular
- **Database:** MySQL / PostgreSQL (Normalized for optimal performance)
- **Microservices Communication:** RESTful APIs / gRPC
- **Cloud Services:** AWS / Firebase (for hosting and authentication)
- **Version Control:** GitHub

6. Expected Outcome

- A fully functional car rental system that allows seamless booking, payment processing, and vehicle tracking.
- A well-structured ER diagram ensuring data integrity and normalization.
- Efficient microservices-based architecture ensuring scalability and modular development.
- Enhanced customer experience through real-time vehicle availability and discount-based incentives.
- A deployable system with cloud hosting support.

7. ER Diagram: -



GitHub Link: -

<https://github.com/Dhaval8/Microservices-Architecture-Project>