GARAGE MANAGEMENT SYSTEM

Project Design Phase

Solution Architecture

DATE	2.11.2025
TEAM ID	NM2025TMID02770
PROJECT NAME	Garage Management system
MAXIMUM MARKS	4 MARKS

Solution Architecture:

Goals of the Architecture:

• Provide an efficient digital platform for garage operations.

• Ensure accurate tracking of customers, vehicles, and service requests.

• Maintain secure and reliable data flow between different modules.

• Enhance automation in job card generation, billing, and inventory management.

• Improve customer satisfaction with timely notifications and transparent billing.

Key Components:

• Customer Module – Manages customer and vehicle details.

• Service Module – Handles job cards, repairs, and maintenance tracking.

• Inventory Module – Tracks spare parts availability and updates stock automatically.

• Billing Module – Generates invoices and records payment details.

• Notification Module – Sends alerts to customers for service completion or due maintenance.

• Admin Panel – Controls access, manages user roles, and generates system reports.

Development Phases:

1. Design database for customers, vehicles, and service records.

2. Develop modules for customer registration and job card creation.

3. Integrate inventory and billing systems.

4. Implement notification system for updates and reminders.

5. Test the system for functionality, accuracy, and performance.

6. Deploy the application for real-time garage operations.

⬚ Solution Architecture Description:

The Garage Management System architecture is built to automate and streamline the daily operations of automotive repair facilities. It follows a modular approach connecting various functional units like customer management, job tracking, billing, and inventory control. Data flows seamlessly between these modules to ensure consistency and accuracy. The backend manages secure data storage, while the frontend provides an intuitive interface for garage staff and customers. The architecture supports role-based access, ensuring that only authorized users can perform critical actions like billing or inventory updates. The integration of automation in notifications and data synchronization improves efficiency and reduces manual errors. This design ensures scalability, enabling the system to handle multiple garages and a growing customer base in the future.

Example - Solution Architecture Diagram:

[Architecture Diagram Placeholder – Illustrates interaction among modules: Customer Module ↔ Service Module ↔ Inventory ↔ Billing ↔ Notification ↔ Admin Panel]

References:

• https://aws.amazon.com/architecture/
• https://www.geeksforgeeks.org/software-architecture/
• https://www.visual-paradigm.com/guide/architecture/what-is-system-architecture/
• https://www.atlassian.com/software-development