

Project Design Phase

Problem – Solution Fit Template

Date	01 NOV 2025
Team ID	NM2025TMID04625
Project Name	Garage Management System
Maximum Marks	4 Marks

Data Flow Diagrams:

A **Data Flow Diagram (DFD)** is a visual representation of how information moves through a system. It illustrates the flow of data between external entities, processes, and data stores, helping developers and stakeholders understand the system's functionality at a glance.

In the project “*Garage Management System*”, the DFD demonstrates how service requests, vehicle details, and billing information move through various stages of the system.

The diagram shows how **customers** submit service requests, which are recorded in the **service database**. The **manager** assigns mechanics for each job, and once the work is completed, **billing details** are generated and stored. Finally, **notifications** are sent to customers about service completion and payments.

This DFD helps ensure clear understanding of how vehicle servicing, inventory management, and billing operations are integrated within the system.

Example (Context-Level DFD):

- **External Entities:** Customer, Manager, Mechanic, Billing Department
- **Processes:** Customer Registration, Job Assignment, Service Update, Payment Processing
- **Data Stores:** Customer Database, Vehicle Details, Service Records, Billing Information

User Stories:

User stories describe what different users expect from the system in simple, goal-oriented language. In this project, they capture essential functionalities such as vehicle registration, job management, and billing — ensuring smooth garage operations and customer satisfaction.

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance Criteria	Priority	Release
Customer	Vehicle Registration	USN-1	As a customer, I want to register my vehicle and service request online so that I can	The system should allow customers to register their vehicle with	High	Sprint-1

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance Criteria	Priority	Release
			easily book a repair or maintenance service.	details and choose a service type.		
Garage Manager	Job Assignment	USN-2	As a manager, I want to assign service jobs to mechanics based on availability and skill.	The system should list available mechanics and assign them to pending service jobs.	High	Sprint-1
Mechanic	Service Update	USN-3	As a mechanic, I want to update the service status (in progress, completed) so that the manager and customer are informed.	The system should allow mechanics to update the job status in real-time.	Medium	Sprint-2
Billing Staff	Payment & Invoice	USN-4	As a billing staff, I want to generate service bills automatically based on job details.	The system should calculate service costs and generate a printable invoice.	High	Sprint-2
Admin	Reporting	USN-5	As an admin, I want to view reports on completed services, pending jobs, and overall revenue.	The system should generate daily and monthly reports with summary statistics.	Medium	Sprint-3

Functional Requirements:

Following are the **functional requirements** of the proposed *Garage Management System*.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	Customer Registration	Customers can register with name, contact, and vehicle details.
FR-2	Vehicle Information Management	System allows storing vehicle details such as model, number, and service history.
FR-3	Service Booking	Customers can book a service appointment through the portal or in person.
FR-4	Job Assignment	Admin/Manager can assign jobs to available mechanics based on specialization.
FR-5	Inventory Management	Track available spare parts, update stock after each service, and generate purchase alerts.
FR-6	Billing & Payment	Automatically calculate charges, generate invoices, and record payments.
FR-7	Service Status Update	Mechanics can update the progress of ongoing jobs.
FR-8	Notifications	System sends SMS/email updates on service completion and pending payments.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-9	Feedback & Reporting	Customers can provide feedback, and the manager can generate daily/monthly reports.

Non-Functional Requirements:

Following are the **non-functional requirements** of the *Garage Management System*.

NFR No.	Non-Functional Requirement	Description
NFR-1	Usability	The system interface should be user-friendly for both staff and customers.
NFR-2	Security	Only authorized personnel can access job assignments, payments, and reports.
NFR-3	Reliability	The system must accurately maintain service records and job histories.
NFR-4	Performance	System should handle multiple customer and service requests simultaneously without delay.
NFR-5	Availability	The application should be accessible 24/7 for booking and status tracking.
NFR-6	Scalability	The system should support the addition of new service types, users, and garages as it grows.
NFR-7	Maintainability	The software should be easy to update and maintain with minimal downtime.
NFR-8	Data Backup	All records should be backed up daily to prevent data loss.

Technical Architecture:

The proposed *Garage Management System* will be developed as a **web-based application** that helps manage customer registrations, vehicle service tracking, mechanic job assignments, and billing in an efficient and digital manner.

The architecture includes **three major layers** —

1. **Presentation Layer (Frontend)** for user interaction,
2. **Application Layer (Backend)** for business logic and data processing, and
3. **Database Layer (Storage)** for managing data securely.

External APIs such as payment gateways and notification services are integrated to enhance functionality. The system is deployed on a cloud-based infrastructure for scalability and accessibility.

Example: Centralized garage management platform accessible to customers, managers, and mechanics via web and mobile devices.

Reference:

<https://developer.ibm.com/patterns/ai-powered-backend-system-for-order-processing-during-pandemics/>

S.No	Component	Description	Technology
1	User Interface	Customers, managers, and mechanics interact through a responsive web portal.	HTML5, CSS3, Bootstrap 5, JavaScript
2	Application Logic – 1	Handles customer registration and service booking workflows.	Node.js / Express.js
3	Application Logic – 2	Assigns service jobs to mechanics and tracks job progress.	RESTful APIs
4	Application Logic – 3	Generates automated invoices and sends status notifications.	Python (Flask) / Twilio API
5	Database	Stores details of customers, vehicles, services, and billing records.	MySQL / PostgreSQL
6	Cloud Database	Cloud-hosted database for high availability and data backup.	AWS RDS / Firebase
7	File Storage	Stores service receipts, reports, and customer feedback files.	AWS S3 / Cloud Storage
8	External API – 1	SMS and email notification integration for service updates.	Twilio / SendGrid API
9	External API – 2	Payment gateway for online bill payments.	Razorpay / PayPal API
10	Machine Learning Model	Predictive maintenance suggestion (optional future enhancement).	TensorFlow / Scikit-learn
11	Infrastructure (Server / Cloud)	Hosted and managed on scalable cloud services.	AWS EC2 / Google Cloud Platform

Table – 2: Application Characteristics

S.No	Characteristics	Description	Technology
1	Open-Source Frameworks	Uses open-source frameworks for flexibility and cost-effectiveness.	Node.js, Bootstrap, React
2	Security Implementations	Implements role-based access control and encrypted data storage.	JWT Authentication, HTTPS
3	Scalable Architecture	Easily expandable for multiple garage branches and users.	Cloud Load Balancing, Microservices
4	Availability	System hosted on a cloud server ensures 24/7 uptime.	AWS Cloud / Azure
5	Performance	Optimized database queries and API caching for faster response.	Redis / Indexed DB Queries
6	Maintainability	Modular structure enables easy updates and maintenance.	MVC Framework (Express / React)
7	Integration	Supports third-party APIs for payments and communication.	REST / JSON APIs