REQUIREMENT ANALYSIS

Objectives:

Customer Journey Map

Data Flow Diagram

Solution Requirements

Technology Stack

Customer Journey Map

1. Booking Phase:
   - Customer searches for garage services online or via phone.
   - Checks availability, books appointment through portal or front desk.
   - Receives confirmation and cost estimate.
2. Drop-off Phase:
   - Customer drops off vehicle at garage.
   - Service advisor inspects vehicle, confirms issues, updates estimate if needed.
3. Service Phase:
   - Mechanic performs inspection and service.
   - If additional work is needed, customer is notified for approval.
   - Real-time updates sent to customer via SMS/email.
4. Billing & Payment Phase:
   - Customer receives final invoice with breakdown of services and parts.
   - Payments processed digitally or at front desk.
   - Receipt provided to customer.
5. Feedback & Follow-up Phase:
   - Customer receives reminder for feedback and rating.
   - Garage collects feedback to improve services.

6.Awareness & Research:

-Customer realizes their car needs servicing or repair.

-Searches online or calls to compare services, prices, and availability.

-Pain Point: Lack of clear information about services and pricing.

System Needs:

- Easy-to-navigate website or portal listing services, base pricing, certifications, and reviews.
- FAQ section to answer common questions.

7.Vehicle Drop-off & Check-in:

-Customer drops vehicle at the garage.
-Service Advisor inspects vehicle, explains issues, updates estimates if needed.

-Pain Point: Fear of unexpected charges or delays.

System needs::

- Digital check-in with service checklist.
- Electronic approval for any extra work beyond the initial estimate
- Customer gets a copy of the approved work order.

Insights:

By mapping this journey, you can:

- Identify where delays or confusion happens.
- Prioritize which steps to automate.
- Design the system to keep the customer informed and build trust.
- Reduce calls asking for status updates.
- Add Customer Emotions: Anxious → Confident → Relieved → Satisfied.
- Identify “Moments of Truth”: E.g., when giving approvals for extra work — these are points where trust can be won or lost.
- Highlight Pain vs. Delight: Show where you can turn friction into a good experience (like real-time tracking).

Purpose:

A clear Customer Journey Map helps you see the entire experience from the customer’s eyes — so you can design features that build trust, save time, and remove friction. It also helps the team prioritize automation, transparency, and clear communication.

Data Flow Diagram

A Data Flow Diagram (DFD) is a visual tool used to show how data moves through a system.

It helps you see:

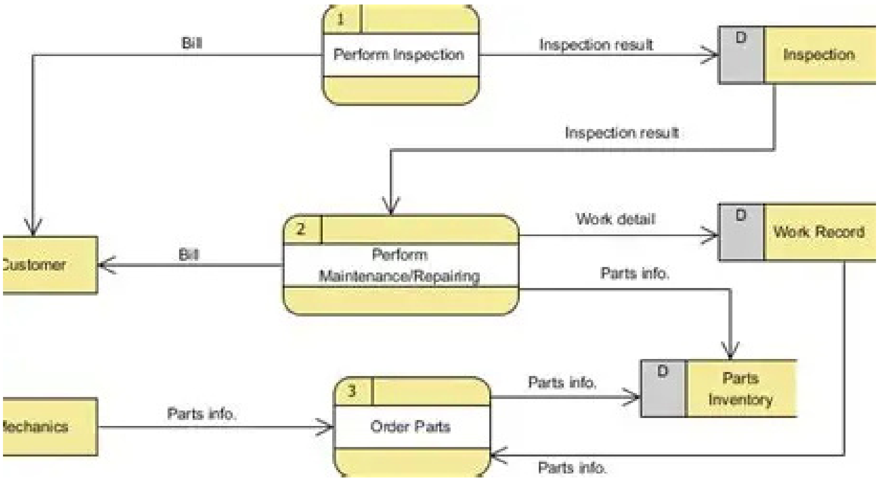
- Where data starts (inputs)
- How it gets processed (processes)
- Where it goes next (outputs)
- What stores the data (data stores)

A DFD does not show how the system works technically (like database tables or code logic). Instead, it focuses on what data is needed, where it goes, and who interacts with it

In Garage Management System, a DFD helps:

- Identify all entities (e.g., Customer, Service Advisor, Mechanic, Billing System).
- Show how information flows between people and systems (e.g., booking, approvals, payments).
- Clarify where automation can reduce manual tasks.
- Highlight where data must be validated to avoid errors like missed fields or null values.

Diagram:



Entities:

- Customer

- Service Advisor

- Mechanic

- Garage Management System (GMS)

- Billing System

- Inventory System

Data Flows:

1. Customer -> GMS: Booking Request
2. GMS -> Service Advisor: Appointment Details
3. Service Advisor -> Mechanic: Work Order
4. Mechanic -> GMS: Service Updates
5. GMS -> Customer: Service Status Notifications
6. Mechanic -> Inventory System: Parts Request
7. Inventory System -> Mechanic: Parts Issued
8. GMS -> Billing System: Service & Parts Charges
9. Billing System -> Customer: Invoice
10. Customer -> GMS: Feedback

This flow ensures real-time updates, inventory checks, transparent billing, and feedback collection.

Solution Requirements:

Goal:

To develop an integrated, user-friendly system that streamlines garage operations — from appointment booking to billing and feedback — while ensuring efficiency, transparency, trust, and data accuracy.

1. Appointment & Scheduling

- Provide an online and offline booking module with a real-time calendar view to prevent double-booking.
- Allow customers to choose available dates/times and receive immediate confirmation.
- Send automated reminders (SMS/Email) for upcoming appointments.

2. Customer & Vehicle Management

- Maintain detailed customer profiles (contact info, feedback history).
- Track vehicle history: past services, parts replaced, warranty details.
- Allow quick lookup of vehicle details for repeat customers.

3. Service Management

- Generate digital work orders for each appointment.
- Enable mechanics to update job status in real-time (e.g., “In Progress”, “Waiting for Approval”, “Completed”).

- Support approval workflows for extra repairs or unexpected work.
- Link parts usage with the inventory system to auto-update stock levels.

4. Inventory Management
    - Track spare parts availability, usage, and reorder levels.
    - Alert staff when stock is low to prevent service delays.
    - Generate reports for fast-moving and slow-moving parts.
5. Billing & Payment
    - Create clear, itemized invoices showing labor, parts, taxes, and discounts.
    - Integrate with payment gateways for digital payments.
    - Allow partial payments if needed (advance and balance).
    - Generate receipts automatically and store billing history.
6. Notifications & Updates
    - Notify customers about booking confirmations, status updates, approvals required, and completion.
    - Keep mechanics and service advisors informed about approvals in real-time.
    - Send reminders for feedback and future services.
7. Feedback & Quality Monitoring
    - Collect customer feedback after each service.
    - Store feedback for analyzing staff performance and service quality.

- User-friendly customer booking portal with calendar view.

- Automated reminders and status updates via SMS/email.

- Role-based access for Service Advisors, Mechanics, and Managers.

- Digital work orders and approval workflows for additional repairs.

- Integration with Inventory System to manage parts stock.

- Secure billing and payment processing with accurate cost breakdown.

- Feedback collection module for customer ratings and comments.

- Dashboards and reports for daily operations, revenue, and staff performance.

- Mobile-friendly interface for mechanics to update status on the go.

Proposed Technology Stack:

1. Frontend (User Interface):

The frontend should be user-friendly, responsive, and easy to use for customers, service advisors, and mechanics.

- Web Technologies:
  - HTML5, CSS3, JavaScript — Standard web building blocks.
  - Framework: React.js (or Angular, Vue.js) for a dynamic and interactive user interface.
  - Mobile Responsive Design: Use frameworks like Bootstrap or Tailwind CSS for mobile-friendly layouts.
  - Optional: Salesforce Lightning Web Components (LWC) if you're building on Salesforce.
- Customer Portal:
  - Online appointment booking.
  - Real-time service status tracking.
  - Payment and invoice view.

2.Backend (Server-Side Processing):

The backend handles business logic, data processing, and secure transactions.

- Platform:
  - Salesforce Platform (for CRM and process automation) — or
  - Node.js + Express.js (if building custom).
- Language:
  - Apex (for Salesforce logic, triggers, and flows) or
  - JavaScript (Node.js) or Python (Django/Flask) for custom solutions.
- Authentication:
  - Role-based access control, user management.
- Notifications:
  - Integration with email/SMS gateways via APIs.

3.Database

The database securely stores customer records, vehicle history, appointments, parts inventory, and billing details.

- CRM Database:
    - Salesforce Standard and Custom Objects for structured data.
- Custom Build:
    - Relational Database: MySQL or PostgreSQL for strong data integrity.
    - Optional NoSQL: MongoDB for flexibility with feedback or logs.

- Frontend: HTML, CSS, JavaScript, React (or Salesforce Lightning Web Components)

- Backend: Salesforce Platform / Apex for business logic and automation

- Database: Salesforce Objects and Relationships

- Integration: REST APIs for payment gateways and inventory systems

- Notifications: Salesforce Flows, Process Builder, or third-party SMS/Email services

- Reporting: Salesforce Reports and Dashboards

- Security: Role Hierarchies and Permission Sets