

Project Design Phase

Problem–SolutionFitTemplate

Date	01NOV2025	
TeamID	NM2025TMID03923	
ProjectName	Garage Management System	
MaximumMarks	4Marks	

Problem–SolutionFitTemplate:

The Problem–SolutionFit means identifying the real pain points of garage owners and customers, and ensuring that the proposed Garage Management System (GMS) effectively addresses those problems. It helps developers and entrepreneurs design practical solutions that improve service efficiency, customer experience, and business operations.

Purpose:

- Solve complex problems in garage operations such as manual record-keeping, delayed communication, and disorganized service tracking.
- Increase efficiency by automating repetitive tasks like customer registration, billing, and vehicle service scheduling.
- Improve customer satisfaction by providing real-time updates and transparent billing.
- Strengthen business performance by maintaining accurate data, reports, and performance analytics.
- Help garage owners make informed decisions using data-driven insights and service tracking modules.

Template:

The project “Garage Management System” addresses major challenges in traditional garages where manual work leads to inefficiency, data loss, and customer dissatisfaction. By developing an automated and centralized platform, garages can handle operations digitally—covering customer details, vehicle information, service history, mechanic assignments, and billing.

The solution introduces modules for:

- Customer & Vehicle Management
- Service Booking & Status Tracking
- Inventory & Spare Parts Management
- Billing & Invoice Automation
- Feedback & Report Generation

This approach enhances accuracy, accountability, and productivity. Garage owners can monitor ongoing services, track parts usage, and generate instant invoices—all from one dashboard.

Customers benefit from better communication, faster service, and reliable updates.

The successful implementation of the Garage Management System will modernize garage

workflows, promoted data integrity, and build trust through efficient and transparent operations.

References:

1. <https://www.ideahackers.network/problem-solution-fit-canvas/>
2. <https://medium.com/@epicantus/problem-solution-fit-canvas-aa3dd59cb4fe>

Proposed Solution Template:

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Most garages still rely on manual methods for maintaining vehicle
2.	Idea/Solution Description	The proposed Garage Management System is a digital platform
3.	Novelty/Uniqueness	The system provides an all-in-one solution for small and medium
4.	Social Impact/Customer Satisfaction	It enhances transparency and customer trust by sending service
5.	Business Model (Revenue Model)	The Garage Management System can be offered as a subscription
6.	Scalability of the Solution	The system can be extended to include advanced analytics, mobile

Reference: Internal Design Concept created using Figma.

Solution Description:

The Garage Management System (GMS) is a completed digital solution developed to simplify and automate daily garage operations. It helps manage customer details, vehicle history, job cards, mechanic schedules, and billing in one unified platform.

The system enables garage owners to:

- Maintain detailed service histories for each vehicle
- Track spare parts usage and stock levels
- Generate invoices automatically after service completion
- Notify customers about service status through SMS or email

By reducing manual record-keeping and improving workflow coordination, the Garage Management System ensures faster service delivery, better accountability, and improved customer satisfaction. This solution not only modernizes traditional garage operations but also paves the way for digital transformation in the automobile service industry.

Solution Architecture:

Goals of the Architecture:

- Develop a centralized digital system for managing garage operations.
- Maintain accurate data for customers, vehicles, and service records.
- Streamline mechanic assignments, inventory tracking, and billing.
- Enhance transparency and customer satisfaction through automation.

Key Components:

- Customer Module: Stores customer and vehicle information.
- Service Management Module: Tracks ongoing repairs, job cards, and mechanic tasks.
- Inventory Module: Manages spare parts, availability, and stock alerts.
- Billing & Reports Module: Automates invoices and generates financial reports.

- UserInterface(WebPortal):Forgaragestaff,mechanics,andcustomerstointeractwiththesystem.

DevelopmentPhases:

- Designdatabasestructureforcustomer,vehicle,service,andbillingdata.
- Buildweb-baseduserinterfacefordataentryandservicetracking.
- Implementserviceschedulingandmechanicassignmentworkflows.
- Integratebillingandinventoryautomation.
- Testsystemwithsampledataandrefinebasedonfeedback.

The Garage Management System (GMS) architecture is designed to automate and streamline every aspect of garage operations. It integrates customer management, vehicle service tracking, inventory control, and billing into a single unified platform.

The architecture follows a modular approach, ensuring scalability and easy maintenance. The frontend web interface allows staff and customers to access real-time information, while the backend database stores and processes all operational data.

Each module interacts seamlessly through defined data relationships:-

The Customer Module links to Service Records for tracking vehicle history.

- The Inventory Module updates automatically when spare parts are used during a repair.
- The Billing Module generates detailed invoices once a service is marked complete.

This digital workflow eliminates manual paperwork, minimizes errors, and provides real-time insights into service progress and business performance. The architecture ensures data integrity, efficient resource utilization, and enhanced decision-making for garage owners.

Example–SolutionArchitectureDiagram:

Figure 1: Architecture and Data Flow of the Garage Management System (Frontend Web Application → Backend Server → Database → Reports Dashboard)

Reference:

1. <https://aws.amazon.com/architecture/>
2. <https://www.ibm.com/cloud/architecture>

