Garage Management System

Phase 2: Requirement Analysis

COLLEGE: Andhra University College of Engineering
TEAM ID : LTVIP2025TMID31547
TEAM SIZE: 1
TEAM LEADER: KEERTHI REDDDY
MAIL: reddykeerthi648@gmail.com
Roll:322506402237
Date: [JUNE 2025]

1. Customer Journey Map

The customer journey map outlines the end-to-end experience a customer goes through when interacting with the Garage Management System. It highlights customer interactions, touchpoints, expectations, and the system's response at each phase.

Stage	Customer Action	Touchpoints	Customer Experience	System Interaction
Awareness	Learns about garage via ads/web search	Website, Social Media, Google	Curious, looking for reliable service	Homepage viewed, SEO-triggered
Booking	Books an appointment for service	Mobile App / Website / Call	Easy, fast booking desired	Appointment form submission, confirmation
Arrival & Check-in	Brings vehicle to garage	Front Desk / Reception	Quick check-in and expected wait time clarity	System logs vehicle and service request
Service	Vehicle is serviced	Mechanic updates via system/app	Wants real-time updates and transparency	Service status updated in real-time
Payment	Makes payment post-service	App / POS / Invoice email	Needs clear billing and smooth payment process	Invoice generated, payment processed
Feedback	Provides rating and comments	App / Website / SMS link	Opportunity to improve loyalty	Feedback stored for analytics
Return /Repeat	Returns for next service	App reminder / Email / SMS	Seeks reliable follow-up and reminders	Notification scheduled and sent

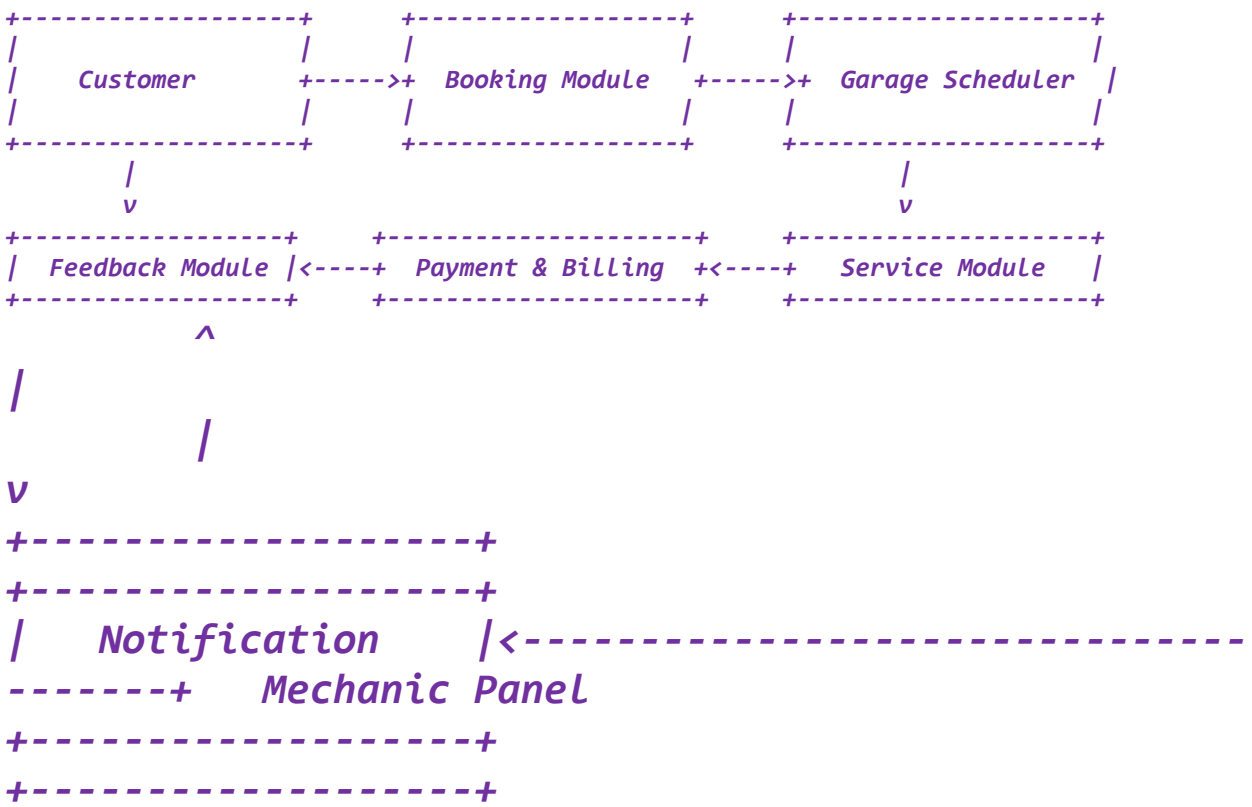
2.Data Flow Diagram (DFD) – Level 1

The DFD illustrates how data flows through the system, showing interactions between the user and major system modules.

Entities and Processes:

- **Customer:** Initiates appointments, provides feedback, makes payments.
- **Booking Module:** Captures booking details and schedules service.
- **Garage Scheduler:** Assigns jobs to available mechanics.
- **Service Module:** Handles the actual service updates and logs.
- **Payment & Billing:** Processes payment and generates invoices.
- **Feedback Module:** Captures ratings and comments.
- **Notification System:** Sends reminders and status updates.
- **Mechanic Panel:** Allows mechanics to view and update their tasks.

DFD Representation (Textual):



3. Solution Requirements

A. Functional Requirements

- **Customer Interface:**
    - Book, view, reschedule, or cancel appointments.
    - View service history and payment records.
    - Receive SMS/email alerts for upcoming services.
    - Submit feedback and ratings post-service.
- **Admin Interface:**
    - Add/edit/delete service categories.
    - Assign jobs to mechanics and manage availability.
    - View analytics dashboards (service trends, customer feedback).
- **Mechanic Interface:**
    - View assigned jobs for the day.
    - Update status (in progress, completed).
    - View service instructions and vehicle history.
- **Billing & Payment:**
    - Auto-generate invoices upon service completion.
    - Accept online payments (UPI, credit/debit cards).
    - Record offline payments (cash/cheque).
- **Notifications:**
    - Reminders for upcoming appointments.
    - Status updates during service progress.
    - Payment confirmation and feedback requests.

B. Non-Functional Requirements

- **Usability:** Easy-to-use UI for all user types.
- **Security:** Role-based access control, secure payment gateway.
- **Performance:** System should respond within 2 seconds for any operation.
- **Scalability:** Capable of supporting multiple garage locations.
- **Availability:** 99.9% uptime with minimal maintenance windows.

---

4. Technology Stack

Layer	Technology Choices
Frontend (UI)	React.js / Angular for web; Flutter/React Native for mobile
Backend (API)	Node.js with Express.js / Django / Spring Boot
Database	MySQL / PostgreSQL for relational data
Authentication	Firebase Auth / JWT-based login / OAuth2
Payment Gateway	Razorpay / Stripe / PayPal
Notification System	Twilio (SMS), SendGrid (Email), Firebase Cloud Messaging
Deployment	AWS (EC2, RDS), Azure, Heroku, or Firebase Hosting
Version Control	Git + GitHub/GitLab for source control
Monitoring & Logs	New Relic / LogRocket / Sentry for error tracking
Analytics	Google Analytics / Custom-built admin dashboard

This technology stack ensures a secure, scalable, and responsive garage management solution catering to both customers and internal staff.
