Requirement Analysis phase

Functional Requirements

Date	25 October 2025
Team ID	NM2025TMID02695
Project Name	Garage Management system
Maximum Mark	4 Marks

1. Introduction

The Functional Requirements define the features and business logic of the Garage Management System (GMS). The goal is to ensure that the Salesforce solution aligns with operational needs, covering customer handling, appointment scheduling, service tracking, billing, and feedback collection.

Each requirement here will be implemented primarily using Salesforce features such as **Custom Objects, Validation Rules, Flows, and Relationships**, without code unless necessary.

2. Functional Requirements Overview

2.1 Customer and Appointment Management

Req ID	Requirement Description	Implementation Detail
FR-1.1	The system must record each customer's full name, phone number, and email (Gmail).	Implemented via *Customer Details* custom object with required Phone and Email fields.
FR-1.2	Each appointment must record a unique Vehicle Number Plate for identification.	Create a *Vehicle Number Plate* field (Text(10), Unique, Required) on *Appointment* object.
FR-1.3	Appointments must be linked to Customers.	Establish a **Lookup Relationship** from Appointment → Customer Details.
FR-1.4	Appointment Date and Service Type must be stored for each booking.	Add *Date* and *Picklist (Service Type)* fields with values like "Repair", "Maintenance", "Oil Change".

FR-1.5	System must prevent double-booking for the same vehicle on the same date.	Enforced using a Validation Rule combining Vehicle Number Plate and Appointment Date.

2.2 Service Execution and Tracking

Req ID	Requirement Description	Implementation Detail
FR-2.1	Each *Service Record* must be linked to an existing *Appointment*.	Required **Lookup Relationship** (Service Record → Appointment).
FR-2.2	Track service progress through stages.	Use a **Picklist field: Service Status** with values: “Received”, “In Progress”, “Completed”.
FR-2.3	Record quality verification before billing.	Add a *Quality Check Status (Checkbox)* on Service Record.
FR-2.4	Prevent “Completed” status unless Quality Check = TRUE.	Use a **Validation Rule** to ensure QA is done before marking service complete.
FR-2.5	Store Technician Name and Assigned Mechanic.	Add *Lookup (User)* field to assign work to a Mechanic.

2.3 Billing and Feedback

Req ID	Requirement Description	Implementation Detail
FR-3.1	Each billing record must be linked to a *Service Record*.	Create a **Lookup Relationship** (Billing → Service Record).
FR-3.2	Enforce a *Rating for Service* input (1–5).	Use *Text(1)* field with **Validation Rule** to restrict range between 1–5.
FR-3.3	Track *Payment Status* using a Picklist.	Picklist values: “Pending”, “Paid”, “Cancelled”.
FR-3.4	Automatically calculate final service amount on Appointment.	Create *Currency (Read-Only)* field on Appointment; populate via **Flow**.

FR-3.5	Prevent billing creation unless service is completed.	Validation Rule using Service Status = “Completed” condition.
FR-3.6	Capture feedback comments from customers.	Create a *Long Text Area* field for optional comments.

2.4 Reporting and Dashboards

Req ID	Requirement Description	Implementation Detail
FR-4.1	Manager must view total services completed per day.	Create a **Custom Report Type** for Service Records.
FR-4.2	Dashboard showing service count, average rating, and revenue.	Build a **Manager Dashboard** folder with restricted access.

3. Dependencies

- Customer Details object must exist before Appointment object (for lookup).
- Appointment object must exist before Service Record and Billing objects.
- Flows depend on object relationships and validation rules.

4. Acceptance Criteria

- All functional requirements must be verifiable within Salesforce.
- System must prevent inconsistent or incomplete data.
- All business logic must be declarative (no Apex unless Flow is insufficient).