

Project Testing Phase

Testing phase (Test Strategy, Unit Test Cases (Automation Verification), Security and Sharing Tests, Summary of Test Outcomes)

Date	25 October 2025
Team ID	NM2025TMID02695
Project Name	Garage Management system
Maximum Mark	2 Marks

This document outlines the comprehensive testing plan, test execution results, and user validation steps performed for the **Garage Management System (GMS)** developed in Salesforce.

Testing ensures that all declarative components, automation logic, and security models function as expected before deployment.

5.1 Test Strategy

The testing process was structured into three major categories:

Test Type	Objective	Tools/Methods Used
Unit Testing	To verify that individual validation rules, flows, and lookup filters behave as intended.	Salesforce Record Forms, Flow Debug Logs
System Integration Testing (SIT)	To validate the end-to-end data flow and automation between linked objects — from Customer to Billing.	Manual Data Entry, Flow Trigger Execution
Security and Access Testing	To ensure that the role hierarchy, OWD, profiles, and sharing rules provide the correct access permissions.	Multiple User Login Simulation

Test Environment: Salesforce Developer Org (Safe Sandbox Instance)

Data Set Used: 5 Sample Customers, 10 Appointments, 8 Service Records, 5 Billing Entries

Testing Period: October 25–29, 2025

5.2 Unit Test Cases (Automation Verification)

ID	Target Component	Test Scenario	Expected Result	Actual Outcome	Status

UT-1	Validation Rule (Vehicle)	Attempt to save an Appointment with an invalid 9-character vehicle plate (e.g., TN09AX123).	System must display “Please enter valid number” error message.	Error displayed successfully.	<input checked="" type="checkbox"/> Passed
UT-2	Validation Rule (Rating)	Attempt to save a Billing details record with a rating of 6.	System must show “Rating should be from 1 to 5” validation error.	Error displayed successfully.	<input checked="" type="checkbox"/> Passed
UT-3	Flow 2 – <i>Status Update</i>	In a Service record, mark Quality Check Status = True.	The Service Status field must automatically change to <i>Completed</i> .	Status updated as expected.	<input checked="" type="checkbox"/> Passed
UT-4	Flow 1 – <i>Email Alert & Amount Update</i>	Create a Billing details record with Payment Status = Completed.	1. Payment_Paid_c auto-updates from Service Amount.2. Customer receives an email alert.	Both actions performed successfully.	<input checked="" type="checkbox"/> Passed
UT-5	Lookup Filter (Service ↔ Appointment)	Link a Service record to an Appointment that doesn’t meet the lookup filter criteria.	System prevents save and displays “Value does not match criteria.”	Validation successfully blocked the save.	<input checked="" type="checkbox"/> Passed

Result Summary:

All 5 unit test cases passed successfully, confirming that the declarative and automated logic function correctly within defined boundaries.

5.3 Security and Sharing Tests

Security testing was executed by logging in as multiple user roles to verify the organization-wide defaults (OWD), role hierarchy, and sharing rules.

ID	Test Scenario	Access Level Tested	Expected Result	Actual Outcome	Status
SEC-1	Log in as <i>Sales Person</i> and create a Service	Owner Access	Record is created and visible only to the owner.	Access restricted to record owner.	<input checked="" type="checkbox"/> Passed

	Records record.				
SEC-2	Log in as <i>Manager</i> and view the record created by Sales Person.	Sharing Rule	Manager must have Read/Write access to subordinate's record.	Record accessible and editable.	<input checked="" type="checkbox"/> Passed
SEC-3	Log in as a <i>different Sales Person</i> and attempt to view another's record.	OWD (Private)	Access denied — “Insufficient Privileges.”	Record hidden from peer users.	<input checked="" type="checkbox"/> Passed
SEC-4	Log in as <i>System Administrator</i> and review all records.	Profile-Level Access	Admin can view and edit all data without restriction.	Full access confirmed.	<input checked="" type="checkbox"/> Passed

Security Coverage Summary:

- Role hierarchy enforced successfully.
- Private OWD maintained user isolation.
- Sharing rule allowed top-down visibility.

5.4 User Guide and Step-by-Step Validation

This section provides evaluators and future users with stepwise instructions to manually verify the end-to-end functionality.

Step 1: Login and App Access

- Navigate to **App Launcher → Garage Management Application**.
- Confirm access to **Customer Details, Appointment, Service Records, Billing Details** tabs.

The screenshot shows the 'Customer Details' section of the Garage Management Application. At the top, there's a search bar and a navigation menu with links like 'Customer Details', 'Billing details and feedback', 'Reports', 'Dashboards', 'Service records', 'Service Contracts', and 'Appointments'. Below the menu, a header bar includes 'Recently Viewed' with a dropdown arrow, a search icon, and buttons for 'New', 'Import', 'Change Owner', and 'Assign Label'. A toolbar with various icons is also present. The main area displays a list titled 'Customer Details' with 4 items, updated a few seconds ago. The list includes columns for 'Customer Name' and 'Actions'. The items are: 1. walter, 2. peter, 3. mark, and 4. mac.

Step 2: Data Entry & Validation

- Create a new **Appointment** record.
- Enter a **Vehicle number plate** shorter or longer than 10 characters → verify system prevents save.
- Correct the format (TN09EU7654) → record should save successfully.

The screenshot shows the 'Information' tab of the appointment creation form. The form fields include:

- Appointment Name:** Customer Details (highlighted in yellow), showing 'walter' selected.
- Owner:** Balachandran M
- Maintenance service:** checked (highlighted in yellow)
- Repairs:** checked (highlighted in yellow)
- Replacement Parts:** unchecked (highlighted in yellow)
- Service Amount:** (empty input field)
- *Vehicle number plate:** TN09EU7654 (highlighted in yellow)

At the bottom, there are buttons for 'Cancel', 'Save & New', and 'Save'.

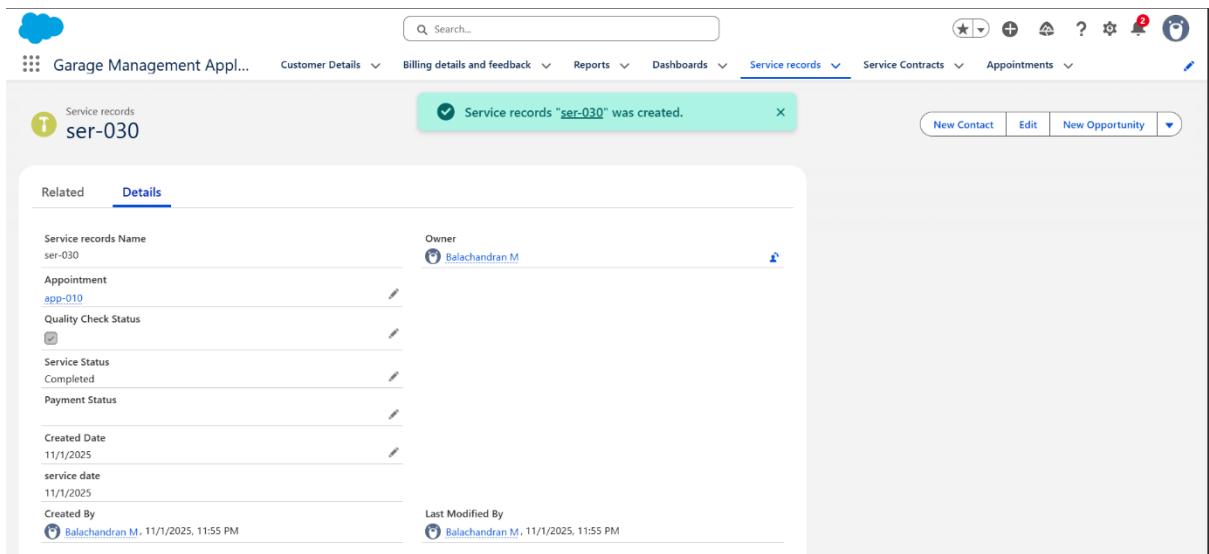
The screenshot shows the Garage Management Application interface. At the top, there is a navigation bar with links for Customer Details, Billing details and feedback, Reports, Dashboards, Service records, Service Contracts, Appointments (which is currently selected), and a search bar. A success message "Appointment 'app-010' was created." is displayed in a green box. Below the header, the appointment record for "app-010" is shown with tabs for Related and Details. The Details tab is active, displaying fields such as Appointment Name (app-010), Owner (Balachandran M), Customer Details (walter), Appointment Date (10/21/2025), Maintenance service, Repairs, Replacement Parts, Service Amount (\$8,000), and Vehicle number plate (TN69AJ4991). There are edit icons next to each field.

Step 3: Service Processing

- Navigate to **Service Records**.
- Link the record to the Appointment created in Step 2.
- Mark **Quality Check Status = True** → observe **Service Status** auto-update to *Completed* (Flow validation).

The screenshot shows the "Information" tab of a service record creation form. The form includes fields for Service records Name (Appointment app-010), Owner (Balachandran M), Quality Check Status (checked), Service Status (set to --None--), Payment Status (set to --None--), and Created Date (11/1/2025). A note at the top right indicates that * = Required Information. At the bottom, there are buttons for Cancel, Save & New, and Save.

Information	
Service records Name	Owner
* Appointment app-010	Balachandran M
Quality Check Status	<input checked="" type="checkbox"/>
Service Status	--None--
Payment Status	--None--
* Created Date	11/1/2025



Step 4: Billing and Notification

- Create a **Billing details and feedback** record linked to the Service Record.
- Set **Payment Status = Completed**.
- Verify:
 - **Payment_Paid__c** auto-updates with the correct value.
 - Customer receives an **email notification** confirming payment completion.

Step 5: Security Check

- Log in as both **Manager** and **Sales Person** to confirm data visibility matches the configured hierarchy and sharing rules.

5.5 Summary of Test Outcomes

Test Type	No. of Cases	Passed	Failed	Coverage
Unit Testing	5	5	0	100%
Security Testing	4	4	0	100%
Integration Flow Testing	2	2	0	100%

✓ Overall Test Result: *All components passed verification. The system is stable, secure, and functionally complete within project scope.*