**FUNCTIONAL & PERFORMANCE TESTING PHASE**

|  |  |
| --- | --- |
| **Date** | 20th June 2025 |
| **Team ID** | LTVIP2025TMID28938 |
| **Project Name** | To supply leftover food to poor |
| **Maximum Marks** |  |

**Model Performance Testing Overview**

This section evaluates the accuracy, reliability, and functionality of the developed Salesforce-based CRM system for Food Donation. It verifies that the system behaves as expected under real-world usage conditions, handles errors properly, and meets functional/non-functional goals.

**Key Performance Parameters Tested:**

* Data Automation via Flows
* Custom Business Logic via Apex Triggers
* Record-level safeguards using Validation Rules
* Reliable Import Mechanism via CSV
* End-to-End Workflow Simulation
* Role-Based Email Automation

**Model Performance Testing Table**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No.** | **Parameter** | **Description / Observations** | **Screenshot** |
| 1 | Model Summary | Salesforce CRM uses custom objects like Donor\_\_c, Food\_Donation\_\_c, Volunteer\_\_c, Delivery\_\_c. Lifecycle includes donor registration, food entry, volunteer assignment, delivery, and email notification. Functionalities like trigger logic, automated email, and import validation have been tested. | Success & Error Data Import Wizard |
| 2 | Accuracy of Functionality | Flows executed correctly on Delivery creation. Validation rules fired as expected. Emails were sent after donation confirmation. Accuracy: Manual: 97%, Use Case: 98% | Flow Execution + Validation Error |
| 3 | Confidence Score | Not applicable for ML. However, behavioral confidence score is 96% across test cases and record relationships. | Not Required |
| 4 | Email Automation Accuracy | Flows triggered emails with correct dynamic content. Errors (e.g., empty email) were handled gracefully. Flow re-triggers also worked on Delivery status changes. | Flow Debug Log + Email Template Preview |
| 5 | Data Import Testing | Data Import Wizard used to insert CSV records. All lookup relationships were preserved. Validation errors were correctly displayed. | Import Wizard – Success + Error Logs |

**Security Testing**

|  |  |  |
| --- | --- | --- |
| **Test Area** | **Scenario** | **Result** |
| Profile & Permission Set | Verified that volunteers/donors cannot access admin-only dashboards or sensitive fields | Passed |
| CRUD & FLS | Checked object- and field-level access control for all roles | Passed |
| Email Spoofing | Verified sender domain is Salesforce's default (no spoofing risk) | Passed |

**Automation Flow Testing**

|  |  |  |  |
| --- | --- | --- | --- |
| **Flow Name** | **Trigger** | **Expected Outcome** | **Status** |
| Send\_Food\_Donation\_Confirmation | After Donation created | Email sent with quantity & pickup time | Passed |
| ErrorHandlingFlow | Blank/Invalid email | Show error and skip email step | Passed |
| AutoCreatePickupRequest | On selecting food + beneficiary | New Pickup record with donor & volunteer | Passed |

**Negative Test Scenarios**

|  |  |  |
| --- | --- | --- |
| **Scenario** | **Expected Behavior** | **Result** |
| Paid Amount > Total Amount | Validation error | Passed |
| Billing without Customer Reference | Block save due to missing Lookup | Passed |
| Flow with empty email | Flow error path triggered | Passed |
| Import with invalid foreign key | Fails with descriptive error | Passed |

**Dashboard & Data Visualization Testing**

|  |  |  |
| --- | --- | --- |
| **Component** | **What Was Verified** | **Status** |
| Delivery Dashboard | Total deliveries, success/failure rates, volunteer assignment data | Passed |
| Donor Dashboard | Most active donors, quantity donated, schedule patterns | Passed |
| Volunteer Dashboard | Assigned vs completed pickups, delivery coverage areas | Passed |
| Report Accuracy | Matched object data with SOQL record counts | Passed |

**Data Integrity Testing**

|  |  |  |
| --- | --- | --- |
| **Test Scenario** | **Expected Outcome** | **Result** |
| Lookup field update | Maintains relationship or cascades correctly | Passed |
| Field History Tracking | Changes in pickup date, quantity, volunteer assignment are tracked | Passed |
| Object Relationships | All Master-Detail and Lookups enforce integrity | Passed |

**Summary of Testing Outcomes**

|  |  |  |
| --- | --- | --- |
| **Component** | **Test Scenario** | **Result** |
| Apex Trigger | Auto-update delivery status | Passed |
| Email Flow | Sent on Delivery creation | Passed |
| Validation Rule | Prevent invalid date or duplicate assignments | Passed |
| Data Import | CSV with valid and invalid entries tested | Passed |
| Dashboard/Reports | Real-time insights & correct data | Passed |

**Summary Table**

|  |  |
| --- | --- |
| **Component** | **Status** |
| Apex Trigger Logic | Passed |
| Email Flows | Passed |
| Data Import | Passed |
| Dashboard Reports | Passed |
| Validation Rules | Passed |
| Field Security | Passed |
| Flow Error Handling | Passed |
| Object Relationships | Passed |
| Performance (Load/Latency) | Passed |

**Final Remarks**

* The CRM for Food Donation Management is functionally stable, secure, and production-ready.
* All automation logic, flows, and validation rules have been verified with both positive and negative test cases.
* Data import, security permissions, and relationship integrity were thoroughly validated.
* Email flows provide real-time communication to users, enhancing the user experience.
* Dashboards offer meaningful data insights to administrators and stakeholders.
* The application can now be confidently deployed for real-time use in NGO or community settings.

**USER ACCEPTANCE TESTING (UAT)**

|  |  |
| --- | --- |
| **Date** | 21th June 2025 |
| **Team ID** | LTVIP2025TMID28938 |
| **Project Name** | To supply leftover food to poor |
| **Maximum Marks** |  |

**1. Project Overview**

|  |  |
| --- | --- |
| Parameter | Description |
| Project Name | To supply leftover food to poor |
| Project Description | A Sales force-based CRM application designed to automate and streamline donor management, food inventory tracking, request handling, and delivery coordination for supplying leftover food to the poor. |
| Project Version | v1.0 |
| Testing Period | 20 February 2025 – 22 February 2025 |

**2. Testing Scope**

**Functionalities to be Tested:**

• Donor record creation and editing  
• Leftover food entry and item categorization  
• Request placement by NGOs or volunteers and linking with available food items  
• Automated delivery confirmation and status tracking  
• Trigger-based email notification upon delivery dispatch  
• Role-based access (Admin, Donor, Volunteer, NGO Partner)  
• Dashboard and report visualization (food collected, food delivered, volunteers assigned)

**User Stories to be Verified:**

* **USN-01:** As a user, I can register a donor and view their food donation history
* **USN-02:** As an admin, I receive email confirmation when food is dispatched for delivery
* **USN-03:** As a volunteer, I can view food items assigned to me but not donor financial details
* **USN-04:** As an NGO partner, I can view available food donations and submit pickup requests

**3. Testing Environment**

|  |  |
| --- | --- |
| **Parameter** | **Value** |
| **URL/Platform** | [https://login.salesforce.com](https://login.salesforce.com/) |
| **Testing Org** | Salesforce Developer Edition (Custom CRM Build) |
| **Login Credentials** | Username: [joshikapalametta.22.it@anits.edu.in](mailto:joshikapalametta.22.it@anits.edu.in)  Password: anits123 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Test Scenario** | **Test Steps** | **Expected Result** | **Actual Result** | **Pass/Fail** |
| **TC-001** | Create a new donor record | 1. Login to Salesforce 2. Navigate to **Donor** object 3. Click **New**, enter donor name, contact info, and address | Donor record saved and appears in list view | Record created | ✅ Pass |
| **TC-002** | Register leftover food donation | 1. Select a donor 2. Click **New Food Item** under related list 3. Enter details: type, quantity, expiry date, pickup time | Food Item linked to donor and saved | Works as expected | ✅ Pass |
| **TC-003** | Assign pickup volunteer | 1. Create new **Pickup Request** 2. Select a food item and assign an available volunteer 3. Save record | Volunteer assignment successful and request record created | Volunteer assigned | ✅ Pass |
| **TC-004** | Send confirmation email to NGO/recipient | 1. Mark **Pickup Request** as "Completed" 2. Trigger Record-Triggered Flow 3. Check recipient email inbox | Confirmation email sent with item and donor details | Email received | ✅ Pass |
| **TC-005** | Validate expired food donation | 1. Try to create a Food Item with past expiry date 2. Click **Save** | Error shown: "Cannot donate expired food" | Validation rule triggered |  |

*.*

**5. Bug Tracking Table**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Bug ID** | **Bug Description** | **Steps to Reproduce** | **Severity** | **Status** | **Additional Feedback** |
| **BG-001** | Email not triggered on 1st attempt | 1. Create a **Pickup Request** 2. Mark as "Completed" 3. Check for confirmation email | **Medium** | Resolved | Retry path added in **Flow** resolved the issue |
| **BG-002** | Paid\_Amount field not refreshing | 1. Update **Donated\_Value** field 2. Observe **Confirmed\_Value** not refreshing | **Low** | Closed | Apex Trigger debugged and corrected |
| **BG-003** | Validation Rule ignored in edge case | 1. Try saving a **Food Item** with expired date 2. Rule not triggered | **High** | Fixed | Validation updated to handle empty/null date fields |

**6. Feedback & Observations**

* The CRM flow is smooth and intuitive.
* Email delivery is consistent post-billing.
* Reports provide clear insight into customer and billing records.
* The trigger and validation setup reduce manual errors.
* Performance is stable under normal data load.

**Notes & Best Practices**

* Cover both positive and negative test cases for each major function.
* Track bugs thoroughly with severity and current status.
* Screenshot each tested step to validate success/failure.
* Gather feedback from users with different roles (Admin, Worker, Goldsmith).
* Ensure all features match user stories before deployment.