**REQUIREMENT ANALYSIS**

|  |  |
| --- | --- |
| **Date** | 17th June 2025 |
| **Team ID** | LTVIP2025TMID28938 |
| **Project Name** | To supply leftover food to poor |
| **Maximum Marks** |  |

**CUSTOMER JOURNEY MAP**

This map meticulously illustrates the step-by-step process .It critically highlights the points of interaction with the proposed CRM system, detailing how each system interaction not only supports the customer's action but also streamlines the store's operations and enhances data capture for future engagement.

**Customer Journey & CRM Interaction – Leftover Food Supply Management System**

|  |  |  |
| --- | --- | --- |
| **Step** | **User Action (Scenario & Intent)** | **System Interaction (CRM Role & Data Capture)** |
| **1. Restaurant/Caterer offers leftover food** | A restaurant, caterer, or individual informs the NGO/organization about leftover food availability through phone, web form, or mobile app.   **Intent:** Donate food to avoid waste. | **Donor\_\_c record is created or retrieved**   The system records donor details (name, contact info, location, food type, quantity, expiry time). Existing donor records are retrieved for regular contributors. |
| **2. Food details are entered** | Donor gives information about the food—what type it is, how much, when it was cooked, and expiry estimate.   **Intent:** Provide food safety and pickup details. | **Food\_Item\_\_c record created**   The food donation details are stored in a custom object with fields like food type, quantity, packaging, expiry time, and status (Pending, Picked, Delivered). |
| **3. Pickup is requested/scheduled** | The donor confirms a time for the food to be picked up by the organization’s volunteers.   **Intent:** Ensure timely pickup before spoilage. | **Pickup\_Request\_\_c record created**   The CRM logs the pickup request, linking it with the Donor and Food Item. It assigns a volunteer based on availability and location. Validation ensures the pickup time is before expiry. |
| **4. Volunteer collects food** | A volunteer reaches the donor’s location and picks up the food.   **Intent:** Safely transport food to the distribution point. | **Pickup\_Status\_\_c updated**   The volunteer updates the status (Picked Up) via mobile app or portal. System logs timestamp and location. |
| **5. Food is delivered to needy location** | The volunteer reaches the slum, orphanage, or roadside area and distributes the food.   **Intent:** Ensure food reaches the poor in time. | **Delivery\_\_c record created**   Delivery details are logged including location, time, and feedback. Status is updated to “Delivered”. |
| **6. Confirmation & gratitude sent** | Donor expects confirmation that the food was successfully delivered.   **Intent:** Get closure and feel appreciated. | **Flow sends confirmation email/SMS**   A Salesforce Flow sends an automated thank-you email/SMS with delivery confirmation and optional photo/report, enhancing donor satisfaction and encouraging repeat donatio |

**DATA FLOW DIAGRAM (DFD)**

**Level 0 – Context Diagram (High-Level View)**

**Entities (External Stakeholders):**

* **Donor** (e.g., restaurant, individual, caterer)
* **Volunteer** (pickup/delivery personnel)
* **Beneficiary** (poor/hungry recipients)
* **Admin/System Coordinator**

**Single Process:**

* **CRM for Food Supply Management**

**Data Flows:**

* Donor submits food details → System
* Volunteer receives pickup assignment → System
* System sends pickup/delivery instructions → Volunteer
* System sends confirmation message → Donor
* System logs delivery details → Admin

**Diagram (Level 0 – Conceptual Textual View):**

scss

CopyEdit

[Donor] ───> (CRM for Food Supply Management) ───> [Volunteer]

↑ ↓

[Confirmation Email/SMS] [Delivery Status]

**Level 1 – Detailed DFD**

This breaks down the internal processes within the CRM system.

**Processes:**

1. **Capture Donor Details** → creates/updates **Donor\_\_c** record
2. **Log Food Item** → creates a new **Food\_Item\_\_c** record
3. **Schedule Pickup** → creates **Pickup\_Request\_\_c** and assigns a **Volunteer\_\_c**
4. **Log Delivery** → updates **Delivery\_\_c** record with beneficiary details
5. **Send Notifications** → triggered email/SMS sent using Flow

**Data Stores:**

* Donor\_\_c
* Food\_Item\_\_c
* Pickup\_Request\_\_c
* Delivery\_\_c
* Volunteer\_\_c

**Data Flow (Level 1 – Textual Format):**

|  |  |  |
| --- | --- | --- |
| **Step** | **Process Description** | **Data Flow** |
| 1 | Donor registers or returns | Donor\_\_c record created/retrieved |
| 2 | Donor enters food details | Food\_Item\_\_c linked to Donor\_\_c |
| 3 | Pickup is scheduled | Pickup\_Request\_\_c assigned to Volunteer\_\_c |
| 4 | Volunteer picks and delivers food | Delivery\_\_c created/updated |
| 5 | Confirmation sent | Email/SMS sent to Donor from Flow using Donor\_\_c email |

**SOLUTION REQUIREMENTS**

**Functional Requirements**

These are the specific capabilities the system must support to ensure efficient food collection, volunteer coordination, and distribution to beneficiaries.

|  |  |  |
| --- | --- | --- |
| **FR No.** | **Functional Requirement (Epic)** | **Sub Requirement (Story / Sub-Task)** |
| FR-1 | Donor Registration | - Register via form  - Register using Gmail or LinkedIn |
| FR-2 | User Confirmation | - Send confirmation email  - Support OTP for account activation |
| FR-3 | Food Donation Management | - Create Food\_Donation\_\_c records  - Categorize food by type (perishable/non-perishable)  - Track expiry, weight, and condition |
| FR-4 | Pickup Scheduling | - Create Pickup\_\_c records  - Assign volunteers  - Link pickups to donations and donors |
| FR-5 | Delivery Automation | - Generate Delivery\_\_c records from Pickup\_\_c  - Auto-calculate total food delivered |
| FR-6 | Confirmation and Email Alert | - Trigger confirmation emails post-delivery  - Auto-fetch donor email via Lookup and use Flow |
| FR-7 | Role-Based Access Control | - Define roles (Admin, Donor, Volunteer)  - Use Permission Sets to control visibility and actions |
| FR-8 | Reporting & Dashboards | - Create donation summary reports  - Build dashboards for daily/weekly/monthly outreach data |
| FR-9 | Validation & Integrity | - Prevent data errors (e.g., quantity = 0) using Validation Rules  - Enforce Lookup relationships between Donation, Donor, and Volunteer objects |

**Non-Functional Requirements**

These define the system's expected performance, usability, and reliability standards.

|  |  |  |
| --- | --- | --- |
| **NFR No.** | **Non-Functional Requirement** | **Description** |
| NFR-1 | Usability | The system must be easy to use by volunteers, NGO workers, and donors. |
| NFR-2 | Security | Access control using Profiles & Permission Sets to secure donor and delivery data. |
| NFR-3 | Reliability | The CRM should function correctly for all donation and delivery operations. |
| NFR-4 | Performance | Fast record creation and email processing using optimized Flows and Triggers. |
| NFR-5 | Availability | The system should be available 24/7 with minimal downtime. |
| NFR-6 | Scalability | Able to support increasing numbers of donors, food pickups, and volunteers. |
| NFR-7 | Maintainability | Easy to modify logic and Flows to support new donation models or volunteer processes. |
| NFR-8 | Auditability | Enable field history tracking for critical objects like Food\_Donation\_\_c and Delivery\_\_c. |

**TECHNOLOGY STACK**

The CRM system to supply leftover food is built using Salesforce’s modern, secure, and scalable cloud platform. The stack integrates automation, data modeling, validation, and communication to streamline food redistribution.

**Technology Components Overview**

|  |  |  |
| --- | --- | --- |
| **Category** | **Tools / Technologies Used** | **Explanation** |
| Platform | Salesforce Lightning Experience | Offers a responsive and user-friendly interface with modern navigation for volunteers and NGOs. |
| Automation | Record-Triggered Flows, Workflow Rules | Automate actions like sending confirmation emails after successful food delivery. |
| Scripting | Apex Triggers | Automatically update fields (e.g., Delivered\_Quantity\_\_c, Status\_\_c) based on logic like pickup completion. |
| Data Modeling | Custom Objects: Donor\_\_c, Food\_Donation\_\_c, Pickup\_\_c, Delivery\_\_c, Volunteer\_\_c | Represents real-life entities with Lookup/Relationships (e.g., Donor ↔ Food ↔ Pickup ↔ Delivery). |
| Validation & Rules | Validation Rules, Formula Fields | Ensure correct data (e.g., donation quantity > 0) and auto-calculate values like total meals from weight. |
| Communication | Email Alerts, Email Templates, Flows | Use Flows to notify donors after deliveries using customized Email Templates. |
| Reporting & Insights | Reports and Dashboards | Visualize key metrics like food distributed per week, active donors, total pickups, and volunteer hours. |
| Access Control | Profiles, Permission Sets | Role-based security for Admins, Donors, and Volunteers; field-level access control based on responsibilities. |

**Sample Tools Used During Development**

|  |  |
| --- | --- |
| Tool | Purpose |
| Object Manager | To define custom objects and configure field relationships |
| Flow Builder | To automate record-triggered processes like sending confirmation emails |
| Developer Console | To write, test, and debug Apex Triggers |
| Email Template Builder | To create branded and dynamic billing confirmation messages |
| Report Builder | To create custom tabular and summary reports |
| Setup (Profiles & Permission Sets) | To implement user-specific access to fields, objects, and functionalities |