REQUIREMENT ANALYSIS PHASE

---

1. Introduction

The **Requirement Analysis Phase** defines the functional and non-functional requirements essential to developing the *To Supply Leftover Food to Poor* system. This phase identifies how the system should behave, what data it must store, and how users will interact with it.

Since this project aims to build a Salesforce-based platform that connects food donors, NGOs, and volunteers, the requirement analysis ensures that all stakeholder needs are clearly documented before design and development begin. This phase also helps prevent scope creep and establishes a solid foundation for successful implementation.

---

2. Purpose of the System

The purpose of this system is to create a centralized, cloud-based platform to facilitate efficient redistribution of surplus food. By automating the donation, pickup, and delivery processes, it ensures that food wastage is minimized and surplus food reaches those in need quickly and safely.

The system's purpose includes:

- Streamlining the food donation and acceptance process.
- Connecting donors, NGOs, and volunteers in real-time.
- Tracking donations from creation to delivery.
- Ensuring accountability and transparency through digital records.
- Providing reports and dashboards for data-driven insights.

---

3. System Users and Roles

The system involves multiple user groups, each with specific access levels and responsibilities:

User Role	Description	Key Responsibilities
Admin	Manages all user accounts, system settings, and data.	Approves NGOs and donors, monitors activity, generates reports.
Donor (Restaurant/Hotel/Event Organizer)	Registers and lists surplus food.	Creates donation entries with food details and pickup location.
NGO/Organization	Receives and distributes donated food.	Views available food, accepts donations, coordinates delivery.
Volunteer	Handles pickup and delivery tasks.	Updates delivery status, confirms successful handover.

---

4. Functional Requirements

Functional requirements describe what the system must do — its features and behaviors.

Core Functional Requirements

1. **User Registration and Authentication**
    - Donors, NGOs, and volunteers can register securely through the platform.

- Admin can approve or deactivate user accounts.

2. **Food Donation Management**
    - Donors can create new donation records specifying food type, quantity, pickup address, and expiry time.
    - NGOs can view a list or map of available donations and accept suitable ones.
3. **Volunteer Assignment and Coordination**
    - Volunteers can be auto-assigned or manually selected for donation pickup.
    - Volunteers can update pickup and delivery status in real-time.
4. **Donation Tracking and Confirmation**
    - The system tracks the entire donation lifecycle — from donor creation to NGO delivery confirmation.
    - Delivery confirmation includes timestamp, volunteer name, and feedback.
5. **Notifications and Alerts**
    - NGOs receive automatic alerts when a nearby donation is created.
    - Donors receive confirmation when their donation is accepted and delivered.
6. **Reporting and Analytics**
    - The admin can view reports on total donations, deliveries, food saved, and beneficiaries reached.
    - Graphical dashboards for weekly and monthly summaries.

---

5. Non-Functional Requirements

These define the quality attributes and performance criteria of the system.

Category	Requirement
Performance	The system should handle at least 100 concurrent users with response time <3 seconds.
Scalability	Must support increasing data volume as more donors and NGOs join.
Reliability	Ensure zero data loss and consistent availability with Salesforce uptime.
Security	Role-based access control, encrypted data, and secure authentication.
Usability	User-friendly interfaces with clear navigation for all user roles.
Maintainability	Code and configurations must be modular for easy updates.

---

6. Data Requirements

The system requires several Salesforce custom objects to manage and relate data efficiently.

Object Name	Description	Key Fields
Donor	Stores donor organization details.	Donor Name, Contact, Location.

Object Name	Description	Key Fields
Food Item	Contains data about the donated food.	Food Type, Quantity, Expiry Date.
NGO	Records NGO/organization details.	NGO Name, Address, Contact Info.
Volunteer	Details about volunteers handling deliveries.	Volunteer Name, Phone, Availability.
Delivery	Tracks pickup and delivery status.	Donation ID, Volunteer ID, Status, Delivery Time.

These objects are related using **lookup and master-detail relationships**, allowing real-time data synchronization across the Salesforce database.

---

7. System Requirements

Category	Details
Software	Salesforce Developer Org, Apex, Visualforce, Lightning Components
Hardware	Cloud-based – requires no local hardware beyond standard PCs and internet connection
Tools	Salesforce Flow Builder, Reports & Dashboards, Developer Console
Access	Browser-based; compatible with Chrome, Edge, Firefox

---

8. Assumptions and Dependencies

- Donors, NGOs, and volunteers must have internet access for real-time coordination.
- The system depends on the Salesforce cloud infrastructure for uptime and data storage.
- Notifications rely on Salesforce Flow and email alert configurations.
- Users are expected to provide accurate data during registration and donation updates.

---

9. Conclusion

The **Requirement Analysis Phase** provides a clear understanding of what the *To Supply Leftover Food to Poor* system must deliver. It outlines the functional behaviors, performance expectations, and data models that will guide the design and development phases.

By capturing every user requirement precisely, this phase ensures that the final Salesforce solution is reliable, scalable, and aligned with the project's mission to reduce food waste and support the underprivileged through efficient, transparent digital management.