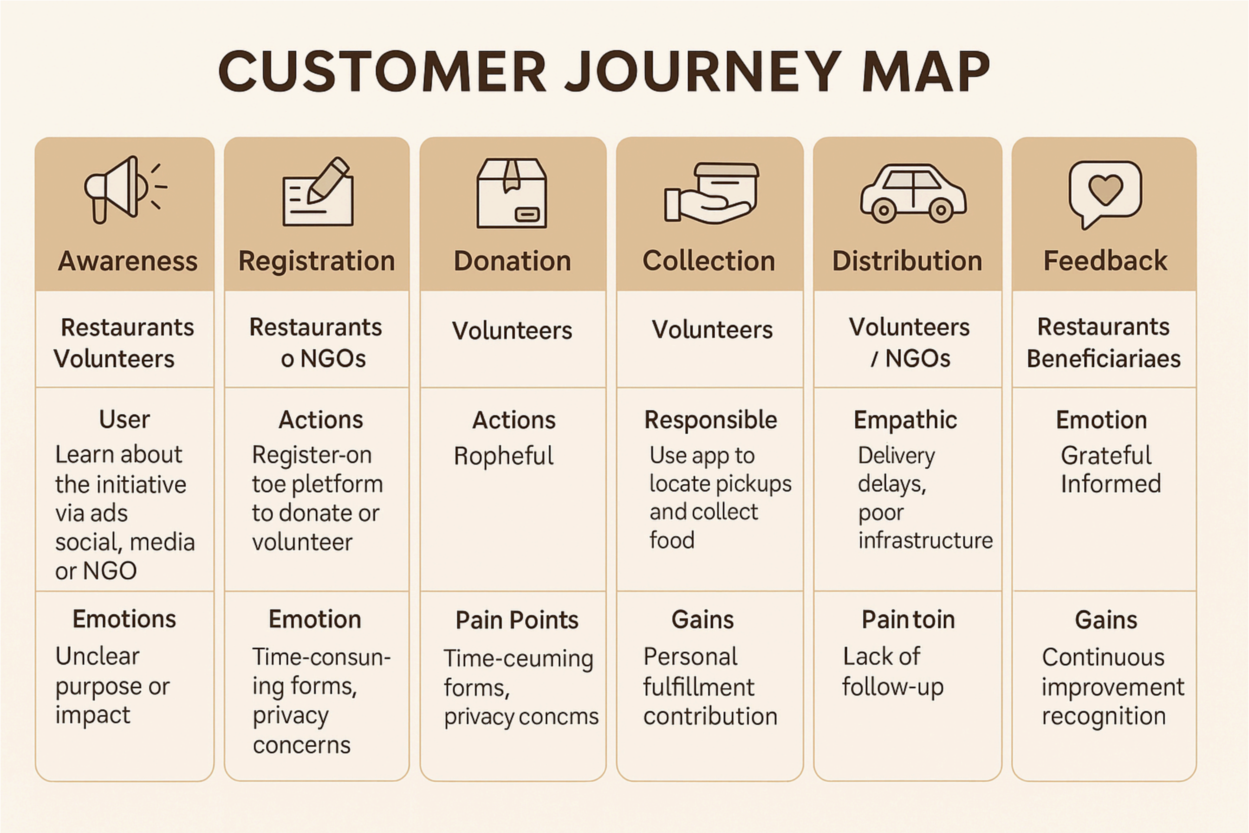
Requirement Analysis Phase

Date	
Team ID	**LTVIP2025TMID31460**
Project Name	**To Supply Leftover Food to Poor**
College Name	**Ideal Institute Of Technology**

Step-1: Customer Journey Map



CUSTOMER JOURNEY MAP

Awareness	Registration	Donation	Collection	Distribution	Feedback
Restaurants Volunteers	Restaurants o NGOs	Volunteers	Volunteers	Volunteers / NGOs	Restaurants Beneficiariaes
User Learn about the initiative via ads social, media or NGO	**Actions** Register-on toe pletform to donate or volunteer	**Actions** Ropheful	**Responsible** Use app to locate pickups and collect food	**Empathic** Delivery delays, poor infrastructure	**Emotion** Grateful Informed
Emotions Unclear purpose or impact	**Emotion** Time-consuning forms, privacy concerns	**Pain Points** Time-ceuming forms, privacy concms	**Gains** Personal fulfillment contribution	**Paintoin** Lack of follow-up	**Gains** Continuous improvement recognition

Step-2: Solution Requirement

🔹 1. Functional Requirements

These describe what the system should do.

ID	Requirement Description
FR1	Users (restaurants, NGOs, volunteers) should be able to register and log in to the system.
FR2	Restaurants should be able to submit food donation details (type of food, quantity, pickup time, location).
FR3	Volunteers should be able to view available food pickups and accept tasks.
FR4	The system should assign and notify volunteers automatically for pickups.
FR5	Volunteers should update collection status (e.g., "picked up", "in transit").
FR6	NGOs or volunteers should mark deliveries as complete and optionally include recipient feedback.
FR7	System should maintain a record/log of all transactions.
FR8	Users should be able to give and view feedback on the process.
FR9	Admin should be able to monitor, audit, and generate reports.

🔹 2. Non-Functional Requirements

These define how the system should behave.

Category	Requirement
Performance	System should support simultaneous requests from at least 100 users.
Reliability	System should have 99.5% uptime.
Usability	Interface must be mobile-friendly and easy to use for all age groups.
Security	User data must be protected via secure authentication & encryption.
Scalability	Platform should support scaling to new cities or states as needed.
Maintainability	System should allow easy updates and bug fixes without downtime.

🔹 3. Data Requirements

ID	Data Requirement
DR1	Store user data (name, role, contact, location).
DR2	Food donation data (type, expiry time, quantity, origin).

DR3	Volunteer task data (assigned, status updates, timestamps).
DR4	Feedback data from donors, volunteers, and recipients.

🔹 4. Technical Requirements

Area	Requirement
Platform	Web-based platform + Android mobile app
Database	Use of cloud-hosted relational database (e.g., PostgreSQL, Firebase)
Hosting	Cloud service provider like AWS, Azure, or GCP
Integration	SMS or push notification service (e.g., Twilio, Firebase)
Mapping	Integration with Google Maps API for navigation

🔹 5. Stakeholder Requirements

Stakeholder	Needs
Donors (restaurants)	Easy way to donate food with trust in the process

Volunteers	Efficient task coordination and safety
NGOs	Smooth distribution and reporting
Admin	Real-time system visibility and user management

Step-3: Data Flow Diagram

Level 0 – Context Level DFD (Overview)

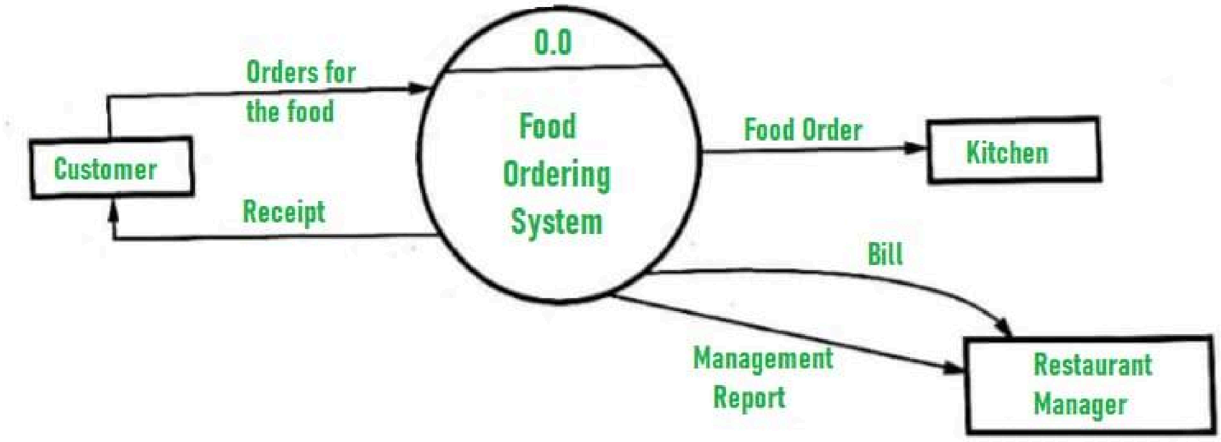
The system is designed and established across the world with input and output at this level.

Food Ordering System has the following input :

- Food order is input as the customer’s order for food.

Food Ordering System has the following output:

- Receipt of the order.
- For further processing the order, the food order is passed to the kitchen.
- The restaurant manager gets the report of Bill and Management.

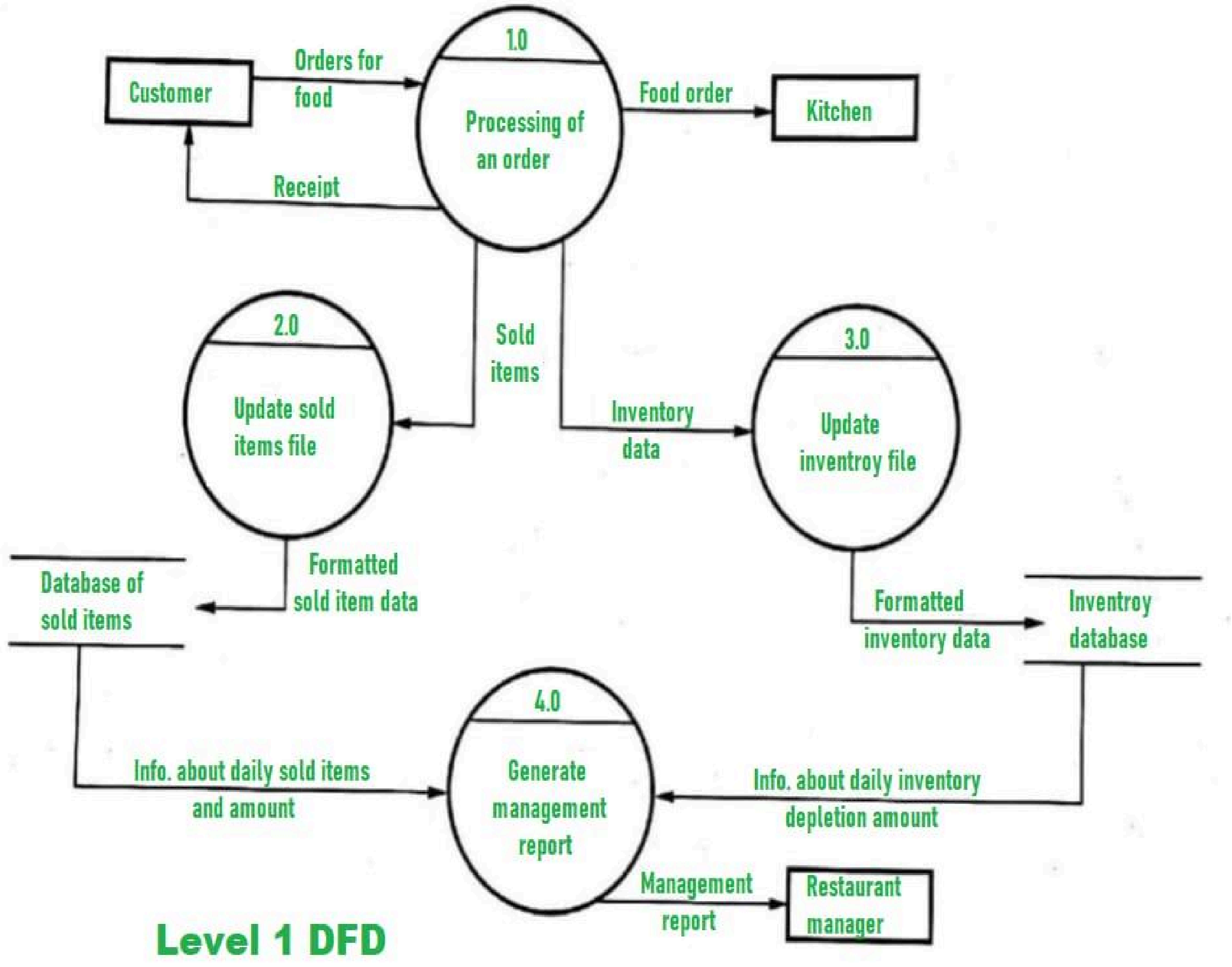


Level 0 DFD (Context Level

Level 1 – Detailed DFD

For processing the order, process 1.0 is responsible. For food, the housekeeping activities involved are represented by processes 2.0, 3.0, and 4.0. The detailed information about daily sold items should be available to create and report management and the list of items that are available 'in-stock' should be kept by maintaining the inventory data (describes the records of datasets such as their name, their content, source, many useful information, etc.) at the same time. Hence, two data stores are used in this level of DFD given below :

- Database of Sold items
- Inventory database



Step-4: Technology Stack Requirement Analysis

The system includes multiple components: frontend (user interface), backend (business logic), database (storage), APIs (communication), and infrastructure.

🔹 1. Frontend (User Interface)

Component	Technology	Purpose
Web App	React.js or Vue.js	For building the restaurant, volunteer, and NGO dashboards
Mobile App	Flutter or React Native	Cross-platform (Android/iOS) app for ease of use in field
UI Design	Figma or Adobe XD	Designing user-friendly wireframes and prototypes

---

🔹 2. Backend (Server-Side Logic)

Component	Technology	Purpose
Backend Framework	Node.js (Express.js) or Django (Python)	To manage API logic and data flow
Authentication	JWT (JSON Web Tokens) / OAuth 2.0	Secure login and role-based access control
Scheduler	Node-Cron / Celery	Automate tasks like reminders, pickup alerts

---

🔹 3. Database (Data Storage)

Type	Technology	Purpose
Relational DB	PostgreSQL or MySQL	Store structured data (users, donations, delivery logs)
Real-time DB (Optional)	Firebase Realtime Database	For quick updates and live tracking
Cloud Storage	Amazon S3 or Google Cloud Storage	Store images (e.g., food photos, proof of delivery)

🔹 4. APIs & Integrations

Type	Technology	Purpose
Maps API	Google Maps API	Location tracking, route optimization for volunteers
Notification API	Firebase Cloud Messaging / Twilio	Send real-time SMS or push alerts
Email Service	SendGrid / Mailgun	Email notifications for confirmations or reminders
Payment Gateway (Optional)	Razorpay / Stripe	If you accept donations or need logistics support funds

---

🔹 5. DevOps & Infrastructure

Tool	Purpose
Docker	Containerize application for consistent deployment
GitHub / GitLab	Version control and collaboration
CI/CD	GitHub Actions, Jenkins, or GitLab CI for continuous deployment
Cloud Provider	AWS / GCP / Azure to host backend, DB, and frontend
Monitoring	Prometheus, Grafana, or Google Stackdriver for system health tracking

---

🔹 6. Security Requirements

Security Layer	Technology
HTTPS	SSL Certificates
Data Encryption	AES or SHA-256 for sensitive info
Role-Based Access	Admin, Restaurant, Volunteer, NGO
Regular Backups	Cloud-based scheduled backups

🔹 7. Optional AI Features (Future Scope)

Use Case	Technology
Predicting food wastage patterns	TensorFlow / Scikit-learn
Matching pickup schedules	AI-based route optimization using Google OR-Tools
Sentiment Analysis on Feedback	Natural Language Processing APIs