

**E-Commerce Website for Local Goods**

# Software Requirements Specification Report

**Software name: Gathr**

**123cs0058** G V Sri Vardhan

**123cs0052** M Vighnesh Prasad

**123cs0011** Ankit Kumar

**123cs0004** N Pranav

Table of Contents

1. Description of Software……………………….1

2. Major Inputs to the System………………….2

3. Major Outputs of the System……………….3

4. Major Processing Functionality……………4

5. Modules of the Proposed System…………5

6. Business Constraints and Technical Constraints….6

7. User Characteristics…………………………..7

8. Data Model and Description…………………8

9. Masters Information………………………….11

10. Transaction Information…………………..12

11. Data Flow Diagrams………………………….13

12. Specific Requirements……………………..14

13. Performance Requirements………………15

1. Description of Software

The **Local Goods Ordering and Delivery System** is designed to connect customers with nearby shopkeepers for purchasing groceries, fruits, vegetables, and other daily-use items. Customers can select a vendor from a list of available shops in their locality, browse their product catalog, place an order, and choose a payment method (online or cash on delivery).

Shopkeepers can manage product listings, update stock availability, and accept or reject orders based on inventory. Delivery personnel are assigned to pick up accepted orders from shopkeepers and deliver them to customers’ specified addresses.

The system supports three user roles:

1. **Customer** – browses products, places orders, tracks deliveries.
2. **Shopkeeper** – manages product catalog, accepts orders, updates status.
3. **Delivery Person** – receives delivery assignments, updates delivery status.

2. Major Inputs to the System

# From Customers

* Login credentials (email/phone & password or OTP).
* Delivery address & contact number.
* Selected products and quantities.
* Payment details for online transactions.

# From Shopkeepers

* Product details (name, price, stock quantity, category).
* Inventory updates (stock in/out).
* Order acceptance or rejection.

# From Delivery Person

* Login credentials.
* Delivery confirmation status (picked up, on the way, delivered).

# From System or External Sources

* Payment gateway inputs for transaction verification.
* Location and GPS coordinates for delivery routing.

3. Major Outputs of the System

# From Customers

* Order confirmation message.
* Delivery status updates (order accepted, dispatched, out for delivery, delivered).

# From Shopkeepers

* New order alerts.
* Order details with customer name, items, and address.
* Payment confirmation from system.

# From Delivery Person

* Login credentials.
* Navigation directions to customer’s address.
* Delivery completion confirmation.

4. Major Processing Functionality

# User Authentication and Role Management

# Secure login or signup with separate dashboards for each role.

# Product Catalog Management

# Add, edit, delete, and update stock for vendor products.

# Order Processing

# Accept/reject orders, manage payment, update status.

# Delivery Management

# Assign delivery personnel based on availability and location.

# Payment Handling

# Integrate payment gateways for online transactions.

# Notifications System

# Send real-time order and delivery status updates to customers and vendors.

# Tracking and Mapping

# GPS integration for delivery tracking.

5. Modules of the Proposed System

# Authentication Module

# Handles login, signup, and role verification.

# Customer Module

# Product search, ordering, payment, order tracking.

# Vendor Module

# Product inventory management, order acceptance, and reporting.

# Delivery Module

# Delivery assignment, live location updates, completion reporting.

# Payment Module

# Secure payment handling and refund processing.

# Notification Module

# SMS, push, and email alerts.

6. Business Constraints and Technical Constraints

# Business Constraints

* The platform will only support vendors within a specified local radius.
* Orders will be delivered within a fixed time window (e.g., 9 AM – 9 PM).
* Delivery charges may vary based on distance and order size.

# Technical Constraints

* Currently only available on the web application.
* Payment gateway must support local currency and comply with RBI guidelines.
* Server must handle at least 1000 concurrent users.
* Real-time updates require stable internet and GPS access.

7. User Characteristics

# Customer

# Basic smartphone, browsing and internet skills. Needs simple navigation and fast checkout.

# Shopkeeper

Familiar with basic web app usage, able to manage product listings.

# Delivery Person

# Can operate a GPS-enabled smartphone, read and follow delivery instructions.

8. Data Model and Description

|  |  |  |
| --- | --- | --- |
| **Table Name** | **Description** | **Key Fields** |
| **Users** | Stores all platform users including customers, shopkeepers, and delivery personnel. | id (PK), created\_at, first\_name, last\_name, email, role, clerk\_id |
| **Shops** | Represents vendor shops managed by shopkeepers. | id (PK), created\_at, owner\_id (FK→Users), location (JSON), addr, shop\_name, contact, account\_no, rating |
| **Users** | Stores all platform users including customers, shopkeepers, and delivery personnel. | id (PK), created\_at, user\_id (FK→Users), location (JSON), mobile\_no, title, description |
| **Addresses** | Stores customer delivery addresses. | id (PK), created\_at, owner\_id (FK→Users), location (JSON), addr, shop\_name, contact, account\_no, rating |
| **Cart** | Holds items added by customers before checkout. | id (PK), created\_at, updated\_at, status, user\_id (FK→Users) |
| **Items** | Product listings in shops. | id (PK), created\_at, name, description, rating, quantity, price, images (JSON), sold\_qt, category, shop\_id (FK→Shops) |

|  |  |  |
| --- | --- | --- |
| **Table Name** | **Description** | **Key Fields** |
| **Orders** | Stores confirmed orders with assigned delivery personnel and payment info. | id (PK), created\_at, carrier\_id (FK→Users), customer\_id (FK→Users), address\_id (FK→Addresses), status, cart\_id (FK→Cart), shop\_id (FK→Shops), payment\_method, transaction\_id |
| **Payments** | Tracks payment transactions. | id (PK), created\_at, user\_id (FK→Users), order\_id (FK→Orders), amount, status |
| **Comments** | User reviews or feedback on items. Supports nested comments. | id (PK), created\_at, user\_id (FK→Users), comment, parent\_id (self-FK), likes, dislikes, item\_id (FK→Items) |



9. Masters Information

* User Roles Master

Defines the roles: Customer, Shopkeeper, Delivery Person, Admin.

* Shop Categories Master

Classification of shops (e.g., Grocery, Fruits & Vegetables, Bakery, Dairy, Meat, Household Items).

* Item Categories Master

Classification of products within shops.

* Payment Methods Master

Supported payment modes (e.g., Cash on Delivery, UPI, Card, Net Banking).

* Order Status Master

Possible statuses (Pending, Confirmed, Out for Delivery, Delivered, Cancelled).

10. Transaction Information

* User Registrations

Storing new customer, shopkeeper, and delivery personnel profiles.

* Shop Onboarding

Adding new shops with owner details and product listings.

* Cart Transactions

Adding, updating, and removing items in carts.

* Order Transactions

Placing orders, assigning delivery personnel, and tracking order status.

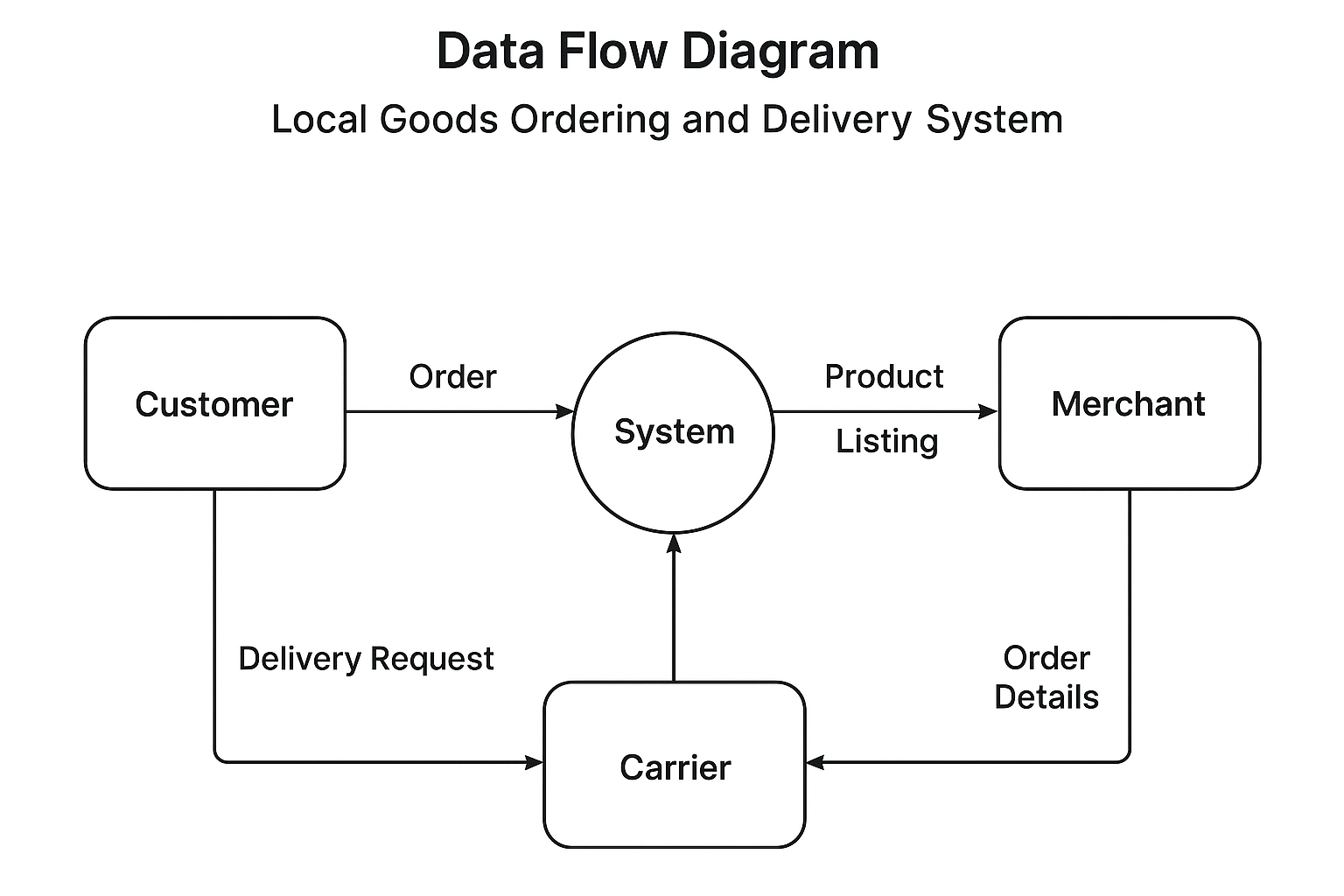
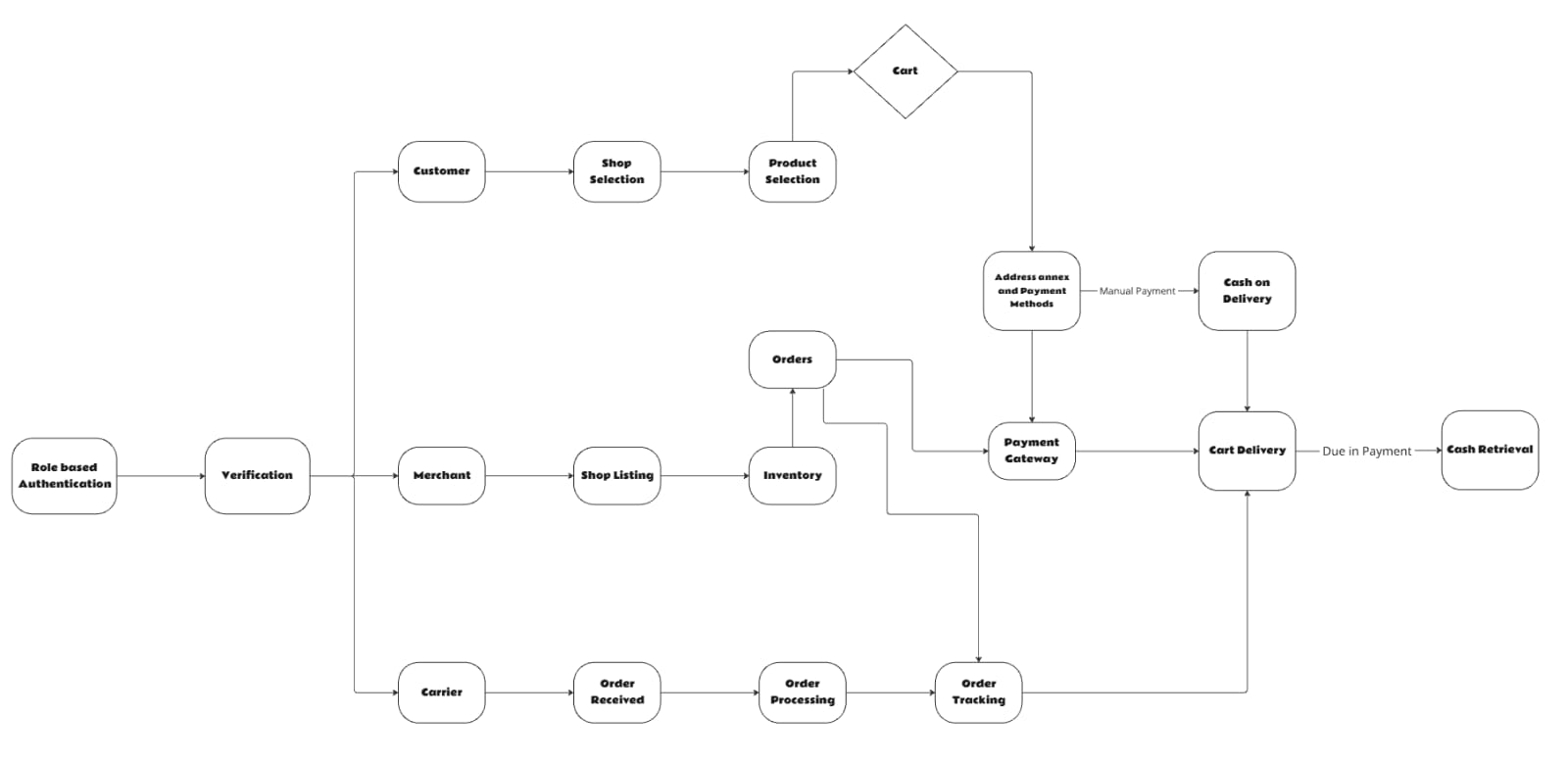
* Payment Transactions

Recording payment method, amount, and transaction status.

* Delivery Updates

Status updates from delivery personnel until the order is completed.

11. Data Flow Diagrams



12. Specific Requirements

# User Interface Requirements

# Customer UI

# Simple dashboard, product catalog, cart, order tracking, notifications.

# Shopkeeper UI

# Inventory management, incoming orders, sales overview.

# Delivery UI

# Assigned orders, navigation map, delivery status updates.

# Admin UI

# Manage users, shops, payments, and reports.

# General UI

# Responsive design, consistent color coding for order statuses, accessibility features.

# Hardware and System Software Requirements

# Server

# Quad-core CPU, 16 GB RAM, 1 TB SSD, Ubuntu Linux, Node.js, MongoDB or PostgreSQL.

# Client Devices

# Smartphones (≥2 GB RAM, Android/iOS) or PC with modern browsers.

# Network or Communication Interface

# All communication over secure HTTPS.

# REST APIs for standard data transfer.

# WebSockets for real-time updates.

# GPS APIs for delivery tracking.

# Network or Communication Interface

# External

# Payment gateway API, Google Maps API

# Internal

# Authentication service (Clerk), ORM/Database connector (supabase ORM).

13. Performance Requirements

# Support ≥1000 concurrent users.

# API response time ≤2 seconds.

# Database queries ≤300ms.

# Real-time updates delivered within 3 seconds.

# Availability: 99.5% uptime.