

Software Requirements Specification (SRS)

Dukaan

Course	Software Engineering (SEMTZC344)
Group	09
Project Title	Dukaan
Team	Sinchana Shetty - 2023LM70047
	Saranya T - 2023LM70049
	John Mathew - 2023LM70050
	Zishan Alam - 2023LM70051
	Helen Jenifer - 2023LM70048

Document Curated By:

Created By	John Mathew - 2023LM70050
Reviewed By	Sinchana Shetty - 2023LM70047

Index

Index	Section	Page No.
1	Introduction	3
1.1	Purpose	3
1.2	Scope	3
1.3	Definitions, Acronyms, and Abbreviations	3
2	Overall Description	3
2.1	Product Perspective	3
2.2	Product Functions	3
2.3	User Classes and Characteristics	3
2.4	Stakeholder Needs and Pain Points	4
3	System Features	5
3.1	Functional Requirements	5
3.2	Non-Functional Requirements	5
4	Data Model	6
4.1	Entity Descriptions	6
5	UML and System Diagrams	7

1. Introduction

1.1 Purpose

Dukaan is a hyperlocal e-commerce platform that connects users with nearby vendors, allowing them to browse, compare, and purchase items at local prices. It aims to decentralize the supply chain, promote local commerce, and support sustainable urban development.

1.2 Scope

- The application enables:
- Users to explore nearby shops.
- Price comparison across vendors.
- Real-time stock visibility.
- Local delivery by area-based partners.
- Vendor inclusion from local and super marts.

1.3 Definitions, Acronyms, and Abbreviations

- SDG 11: Sustainable Development Goal 11 – Sustainable Cities and Communities.
- ER Diagram: Entity-Relationship Diagram.
- CRUD: Create, Read, Update, Delete operations.

2. Overall Description

2.1 Product Perspective

Dukaan is similar to platforms like Blinkit and Zepto but also focuses on local vendor integration and price transparency.

2.2 Product Functions

- User registration and login
- Vendor registration and item listing
- Item search, filtering, and comparison
- Order placement and live tracking
- Delivery assignment and review system
- Admin analytics dashboard

2.3 User Classes and Characteristics

- Customer: Browses items, places orders, tracks deliveries.
- Vendor: Lists items, manages stock and prices.
- Delivery Partner: Fulfills deliveries and receives ratings.
- Admin: Manages users, vendors, deliveries, and analytics.

2.4 Stakeholder Needs and Pain Points Summary

- Customer: Difficulty locating nearby vendors, inconsistent pricing, and slow delivery.
- Vendor: Complex registration and manual stock management.
- Delivery Partner: Lack of real-time info and rating visibility.

Stakeholder	Pain Point / Feedback	System Requirement
Customer	Difficulty finding shops	Geolocation-based shop discovery
Customer	Price varies by store	Price comparison across vendors
Customer	Late/no delivery info	Live delivery tracking
Customer	Product unavailability	Real-time stock info
Vendor	Registration complexity	Simple onboarding
Vendor	Stock management issues	Inventory dashboard
Vendor	No sales analytics	Dashboard with insights
Delivery	No route guidance	Navigation & live order mapping
All	Communication friction	In-app notifications & chat

3. System Features

3.1 Functional Requirements

For Customers

- Shop Discovery: Locate nearby shops using device geolocation.
- Product Listing & Filtering: Browse and filter products by category, price, availability.
- Price Comparison: Compare product prices across different shops.
- Order Placement: Add items to cart, checkout, select delivery options.
- Order Tracking: Live map-based tracking after order is placed.
- Favorites: Mark items/shops for quick future orders.
- Multiple Payment Options: Cash, UPI, card.

For Vendors (Shopkeepers)

- Vendor Registration: Simple signup & onboarding.
- Product Management: Add/edit/delete items, manage inventory and pricing.
- Order Management: View/manage orders, update order status.
- Sales Insights: Dashboard for sales, popular items, customer behavior.
- Stock Alerts: Real-time notifications when inventory is low or orders arrive.

For Delivery Partners

- Order Assignment: Auto-assignment and manual order selection with map filtering.
- Navigation: Integrated map and address details for pickup and delivery.
- Live Status Updates: Update customer/vendor on progress.
- Earnings & History: Track daily earnings, order completion stats.

For Admin

- User & Vendor Management: Approve or suspend accounts.
- Analytics: View platform-wide metrics (orders, revenue, complaints).
- System Maintenance: Manage platform settings and resolve reports.

3.2 Non-Functional Requirements

- Performance: All regular operations (search, add to cart, order status) under 2 seconds.
- Scalability: Should handle city-wide simultaneous users.
- Reliability/Availability: 99.5% uptime.
- Security: All user payments and data encrypted; role-based access.
- Usability: Simple, minimalist interface for customers; detailed/advanced dashboard for vendors.

4. Data Model

4.1 Entity Descriptions

1. Customer

- **CustID** (*PK*)
- Name
- Address

2. Vendors

- **VendID** (*PK*)
- Vendname
- Address

3. Item

- **ItemID** (*PK*)
- Itemname
- Price
- **CategoryID** (*FK* → *Category.CategoryID*)

4. Orders

- **OrderID** (*PK*)
- **CustID** (*FK* → *Customer.CustID*)
- **VendID** (*FK* → *Vendors.VendID*)
- **ItemID** (*FK* → *Item.ItemID*)
- Quantity
- Price

5. Delivery

- **DeliveryID** (*PK*)
- **OrderID** (*FK* → *Orders.OrderID*)
- **DelServID** (*FK* → *DeliveryServ.DelServID*)

6. Category

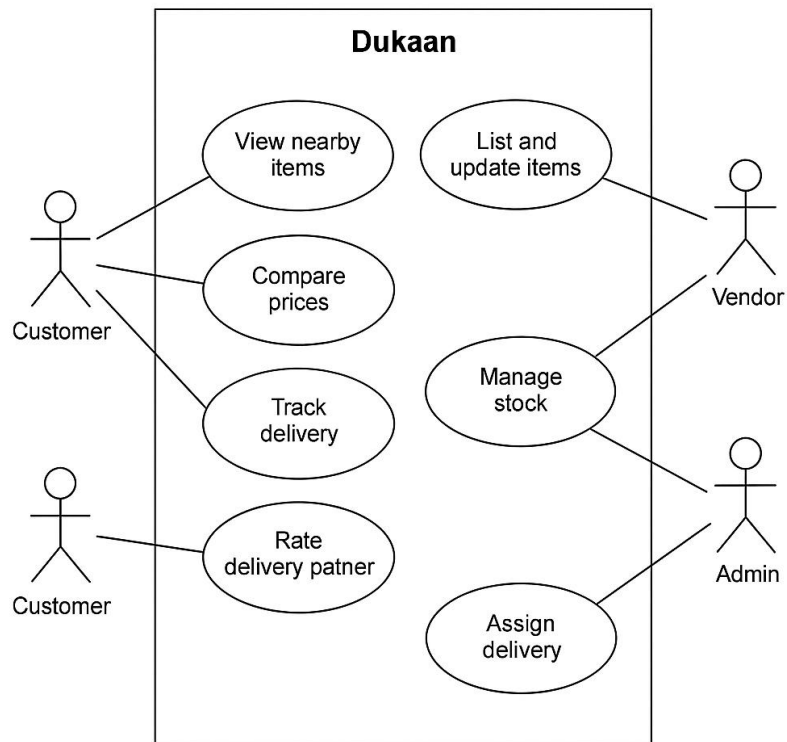
- **CategoryID** (*PK*)
- Categoryname

7. DeliveryServ

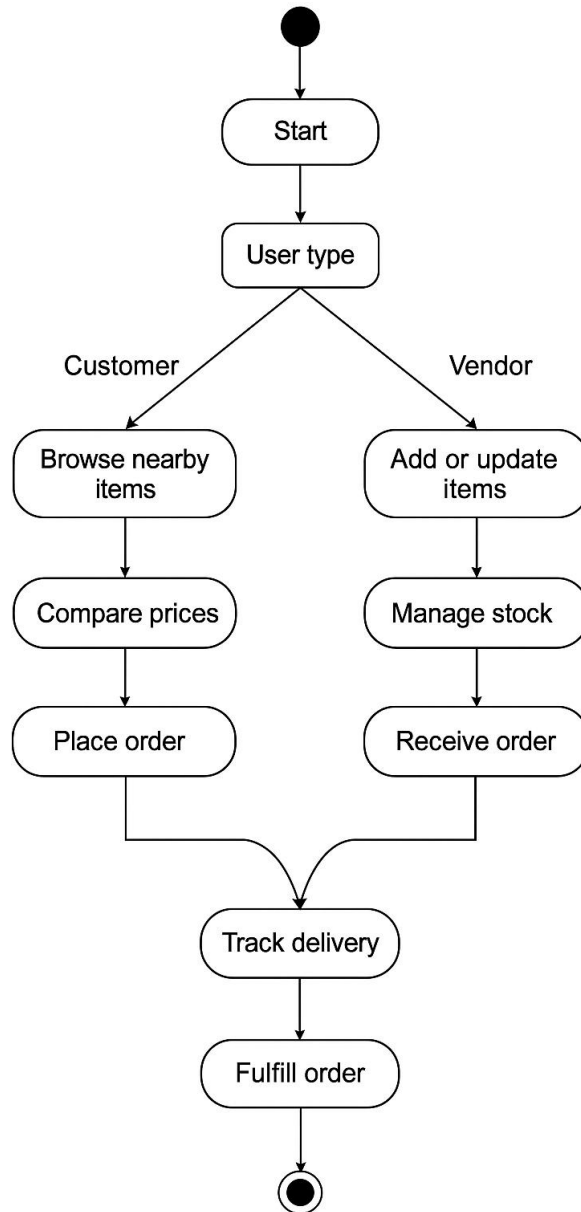
- **DelServID** (*PK*)
- Name

5. UML and System Diagrams

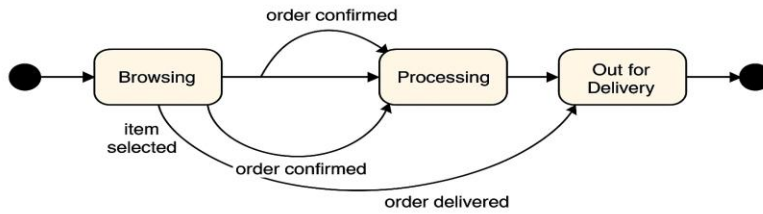
Use Case Diagram



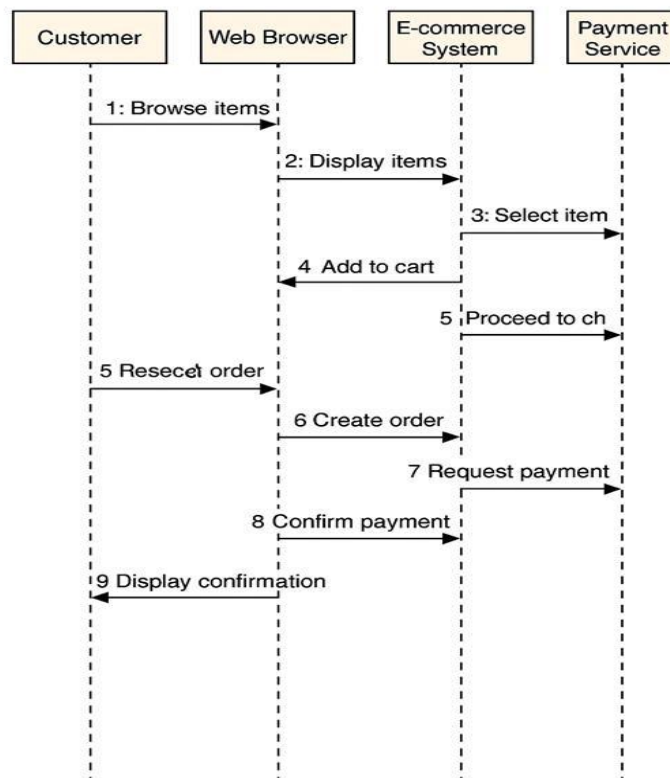
Activity Diagram



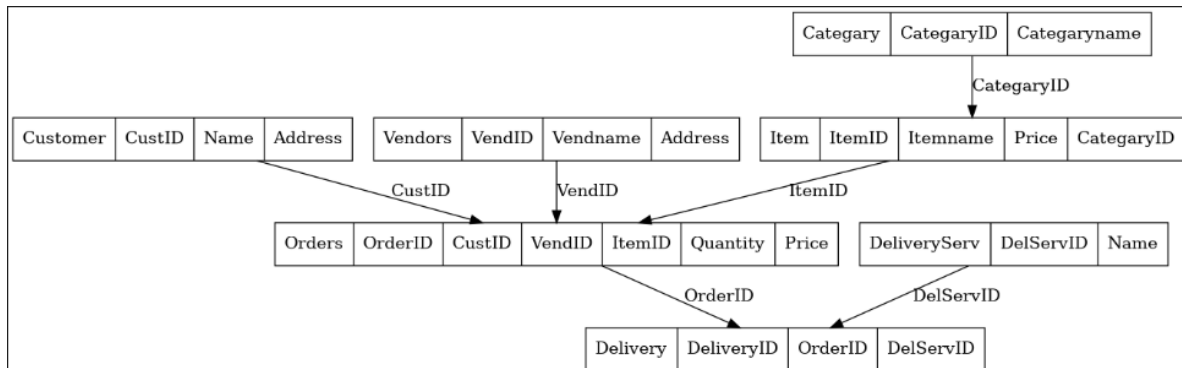
State Machine Diagram



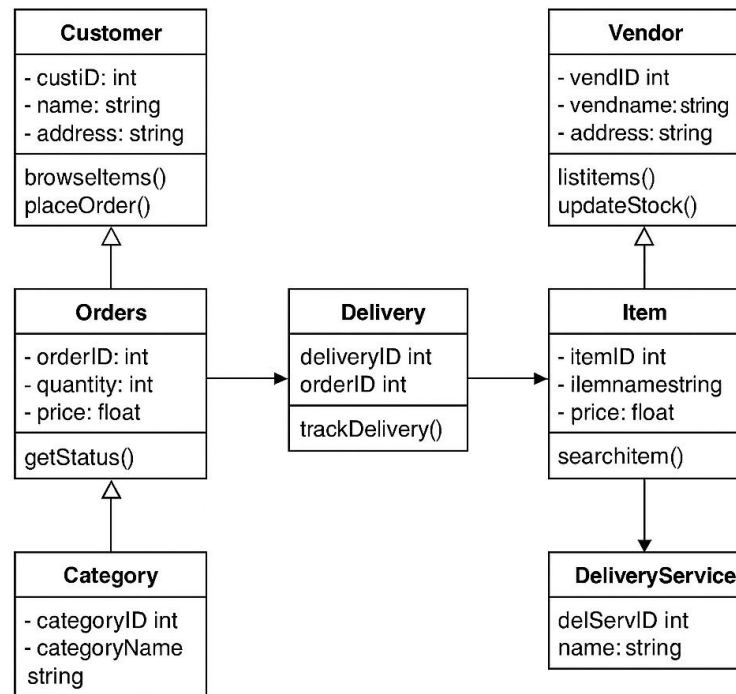
Sequence Diagram



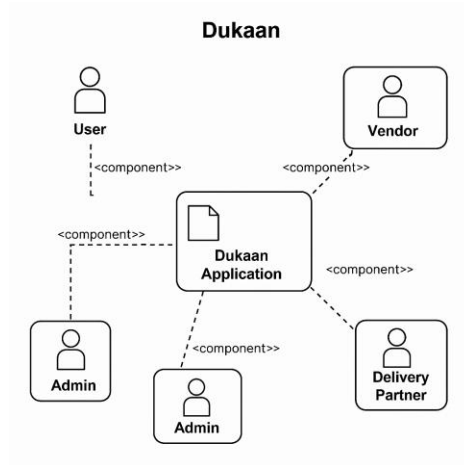
Data Modelling



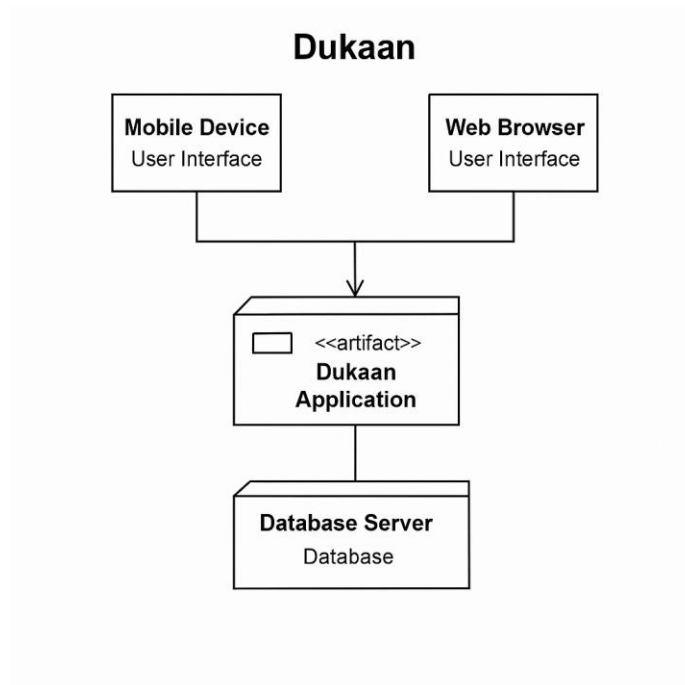
Class Diagram



Component Diagram

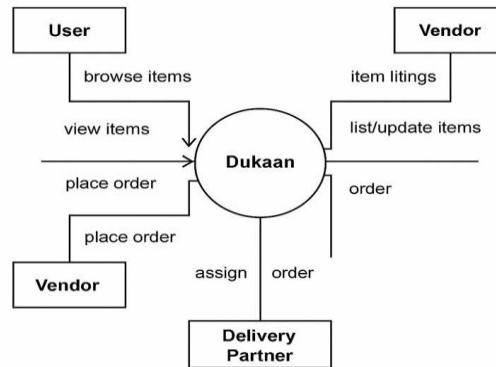


Deployment Diagram



Two levels of Data Flow Diagram

0 level Data Flow Diagram



1 level Data Flow Diagram

