Ahara – Food Redistribution & Waste Reduction Platform

System Design Document

1. Introduction

1.1 Project Overview
Ahara is a technology-driven food redistribution platform designed to reduce surplus food waste by securely connecting sellers, buyers, and volunteers. The platform ensures food safety, transparency, and traceability while optimizing pickup coordination and environmental impact tracking.

1.2 Problem Statement
Food waste contributes to environmental degradation and resource inefficiency. Surplus food often goes unused due to lack of structured redistribution systems.

Key Challenges:

- Lack of trust in redistribution systems

- Poor logistics coordination

- Limited traceability and compliance tracking

- No measurable impact transparency

1.3 System Objective
The Ahara platform aims to:

- Enable safe surplus food listing

- Enforce role-based access control

- Ensure hygiene and safety compliance

- Optimize pickup logistics

- Provide environmental and social impact visibility

- Maintain audit-ready compliance records

2. System Architecture

Figure 1: System Architecture Diagram

Ahara – Food Redistribution Platform Architecture

CLIENT LAYER (FLUTTER MOBILE APP)

Role-Based UI	State Management	API Communication	Multilingual & Accessibility
Buyer / Seller / Volunteer / Admin	Provider + GoRouter	Dio (HTTP + Interceptors)	Language switching + Low literacy mode

↓

SECURITY & IDENTITY LAYER

Firebase Authentication	Role-Based Access Control (RBAC)	Consent & Audit Logs
JWT Token Validation	Dynamic Permission Mapping	Timestamped + Hashed Records

↓

BACKEND API LAYER (NODE.JS + EXPRESS)

Listing Management	Trust & Safety Engine	Matching Engine	Pickup Coordination	Impact Engine
Metadata + Expiry + FSM States	Threshold Rules + Auto Blocking	Geo Radius + Expiry Ranking	Workload Tracking + Route Logic	CO_2 + Meal Units + Pricing Analytics

↓

DATA LAYER (MONGODB ATLAS)

User Collection	Food Listings	Transactions	Audit & Compliance Logs
Roles + Verification Levels	Metadata + Expiry + Batch IDs	Traceability ID + State Logs	Immutable Records

↓

REAL-TIME & NOTIFICATION SERVICES

Live Pickup Updates	Push Notifications	Failure Detection
Status Synchronization	Milestones & Alerts	Auto Rematching Workflow

↓

CLOUD & DEVOPS INFRASTRUCTURE

AWS EC2	PM2 Process Manager	GitHub Actions CI/CD	Environment Management
Ubuntu Server Hosting	Zero Downtime Restart	Automated Testing & Deployment	Secure Secret Variables

Ahara – Secure, Role-Aware, Trust-Centric Food Redistribution System

Figure 1

Presentation Layer: Flutter Mobile App, Web Interface

Application Layer: Node.js + Express backend, RBAC middleware

Data Layer: MongoDB, Firebase Authentication

External Services: Google Sign-In, Maps API, Notifications

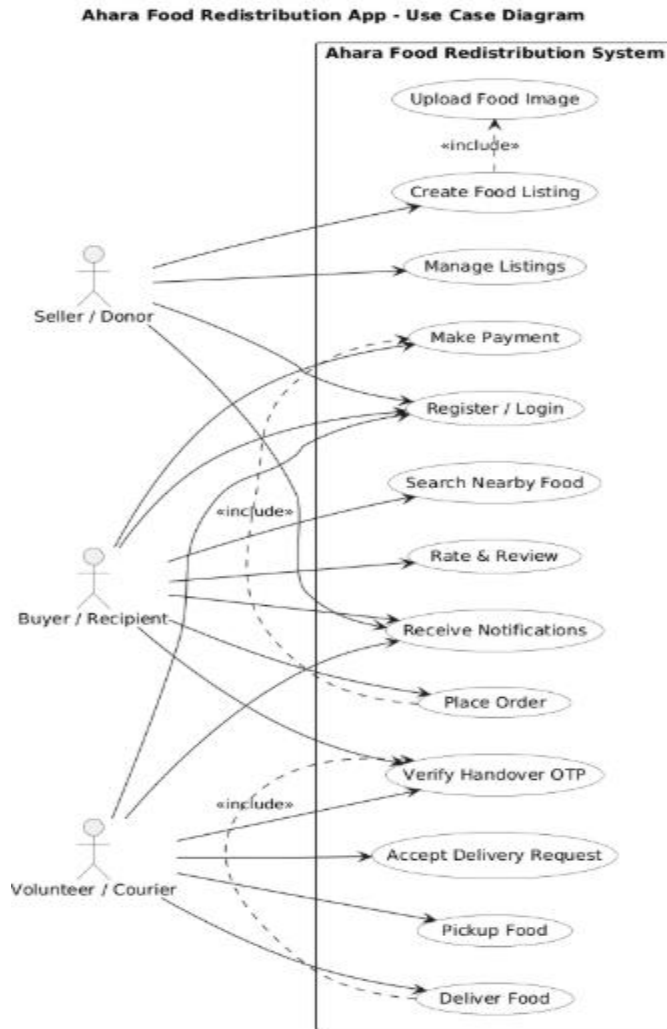# 3. Use Case Diagram

Figure 2: Use Case Diagram

Ahara Food Redistribution App - Use Case Diagram

Ahara Food Redistribution System

- Upload Food Image
- «include»
- Create Food Listing
- Manage Listings
- Make Payment
- Register / Login
- Search Nearby Food
- Rate & Review
- Receive Notifications
- Place Order
- Verify Handover OTP
- Accept Delivery Request
- Pickup Food
- Deliver Food

Actors: Seller / Donor, Buyer / Recipient, Volunteer / Courier

Figure 2

Actors:

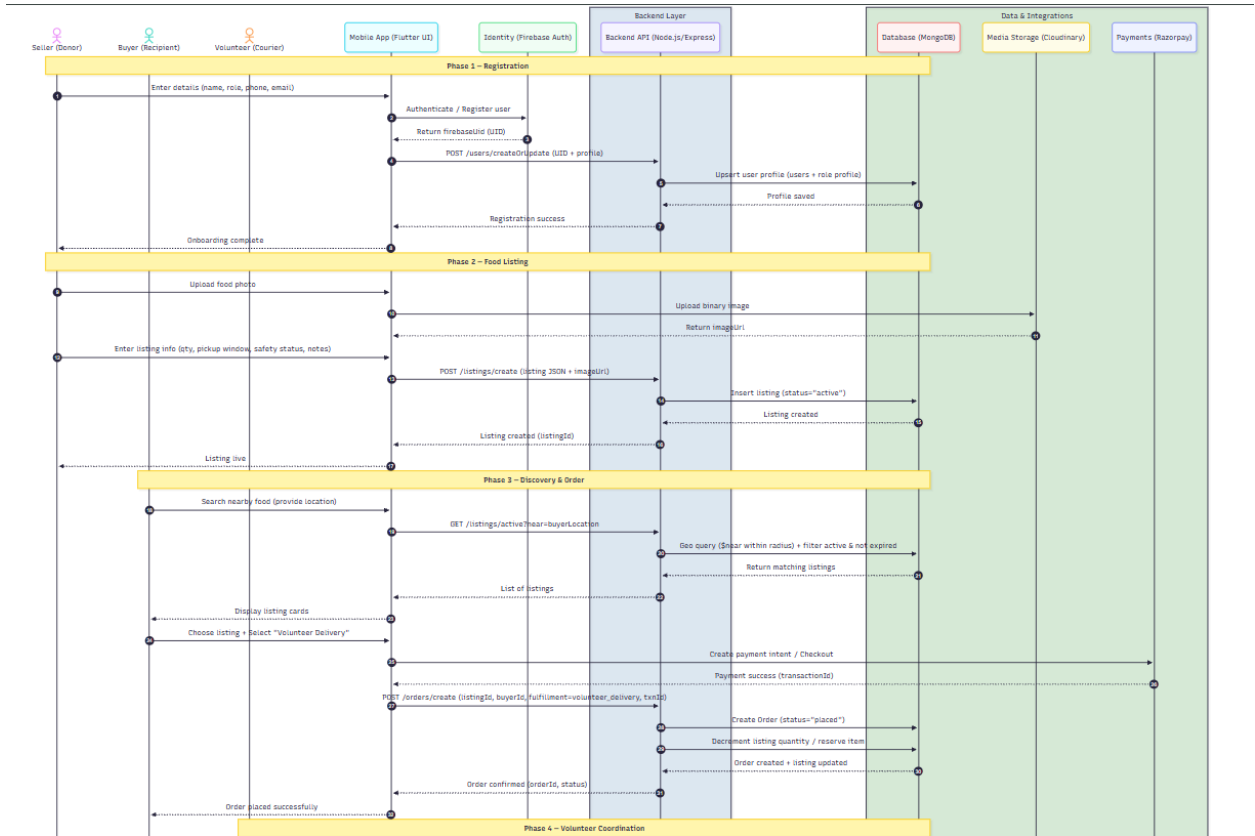Buyer – Browse listings, request pickup, view impact
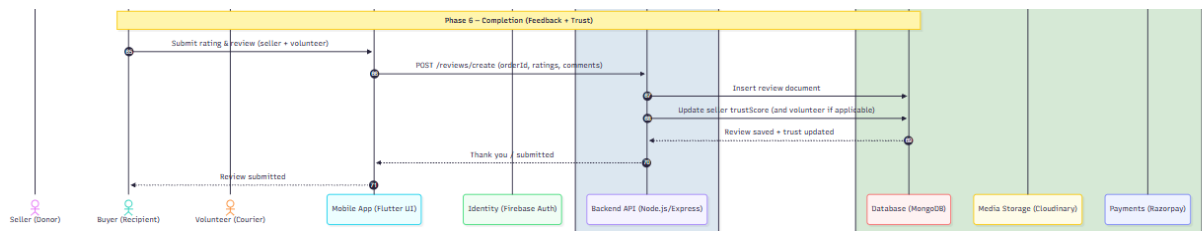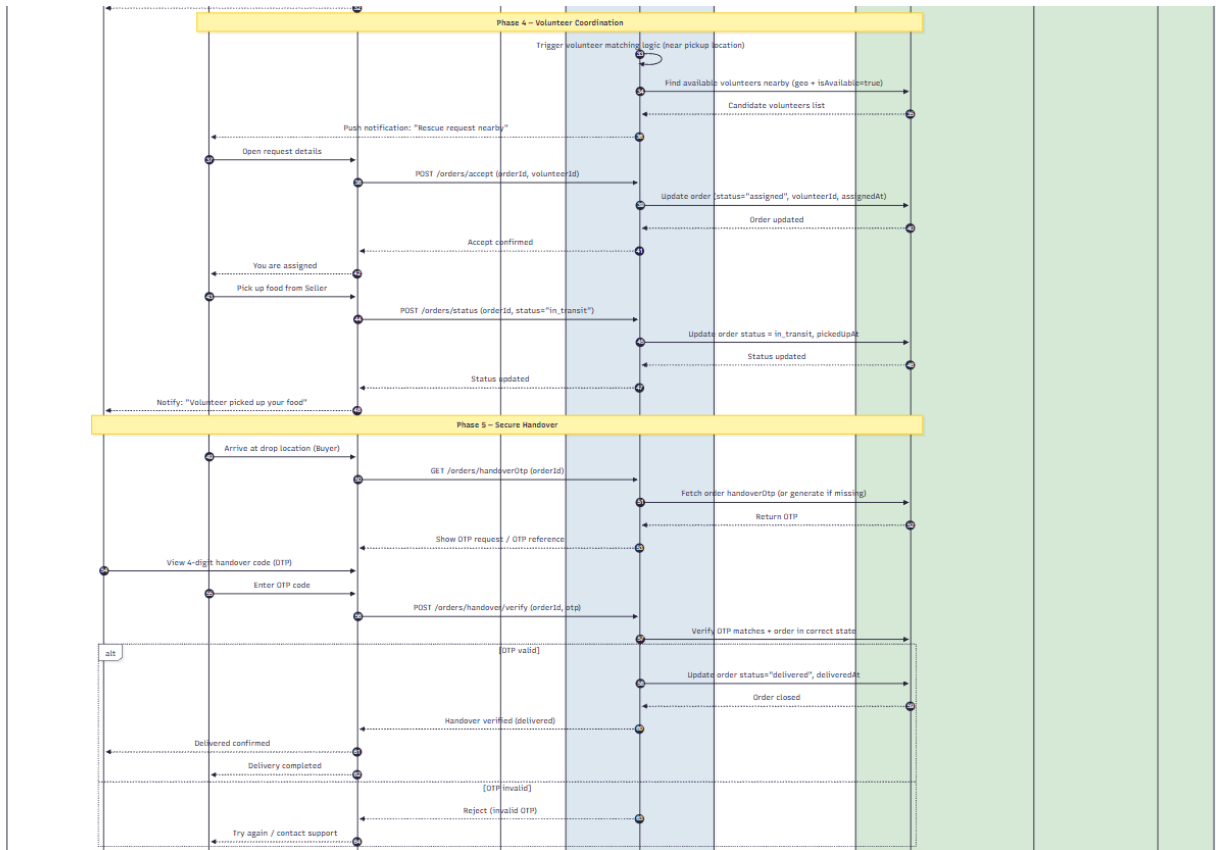
Seller – Create listings, submit hygiene declaration

Volunteer – Accept pickups, update status

Admin – Manage users, compliance, reports

4. Sequence Diagram

Figure 3: Sequence Diagram

Phase 4 – Volunteer Coordination

Trigger volunteer matching logic (near pickup location)

Find available volunteers nearby (geo + isAvailable=true)

Candidate volunteers list

Push notification: "Rescue request nearby"

Open request details

POST /orders/accept (orderId, volunteerId)

Update order (status="assigned", volunteerId, assignedAt)

Order updated

Accept confirmed

You are assigned

Pick up food from Seller

POST /orders/status (orderId, status="in_transit")

Update order status = in_transit, pickedUpAt

Status updated

Status updated

Notify: "Volunteer picked up your food"

Phase 5 – Secure Handover

Arrive at drop location (Buyer)

GET /orders/handoverOtp (orderId)

Fetch order handoverOtp (or generate if missing)

Return OTP

Show OTP request / OTP reference

View 4-digit handover code (OTP)

Enter OTP code

POST /orders/handover/verify (orderId, otp)

Verify OTP matches + order in correct state

alt   [OTP valid]

Update order status="delivered", deliveredAt

Order closed

Handover verified (delivered)

Delivered confirmed

Delivery completed

[OTP invalid]

Reject (invalid OTP)

Try again / contact support

Phase 6 – Completion (Feedback + Trust)

Submit rating & review (seller + volunteer)

POST /reviews/create (orderId, ratings, comments)

Insert review document

Update seller trustScore (and volunteer if applicable)

Review saved + trust updated

Thank you / submitted

Review submitted

Seller (Donor)   Buyer (Recipient)   Volunteer (Courier)   Mobile App (Flutter UI)   Identity (Firebase Auth)   Backend API (Node.js/Express)   Database (MongoDB)   Media Storage (Cloudinary)   Payments (Razorpay)

Flow: User submits registration → Firebase Auth → Backend JWT verification → MongoDB storage → Confirmation response.

Flow: Seller submits listing → Backend validation → Safety checks → MongoDB storage → Matching trigger.
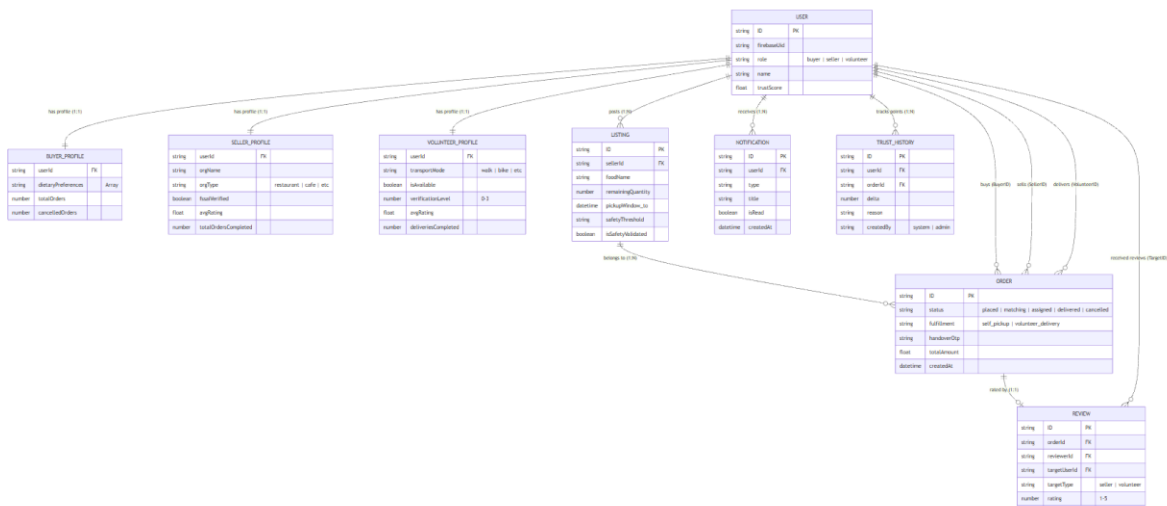
5. Schema Diagram

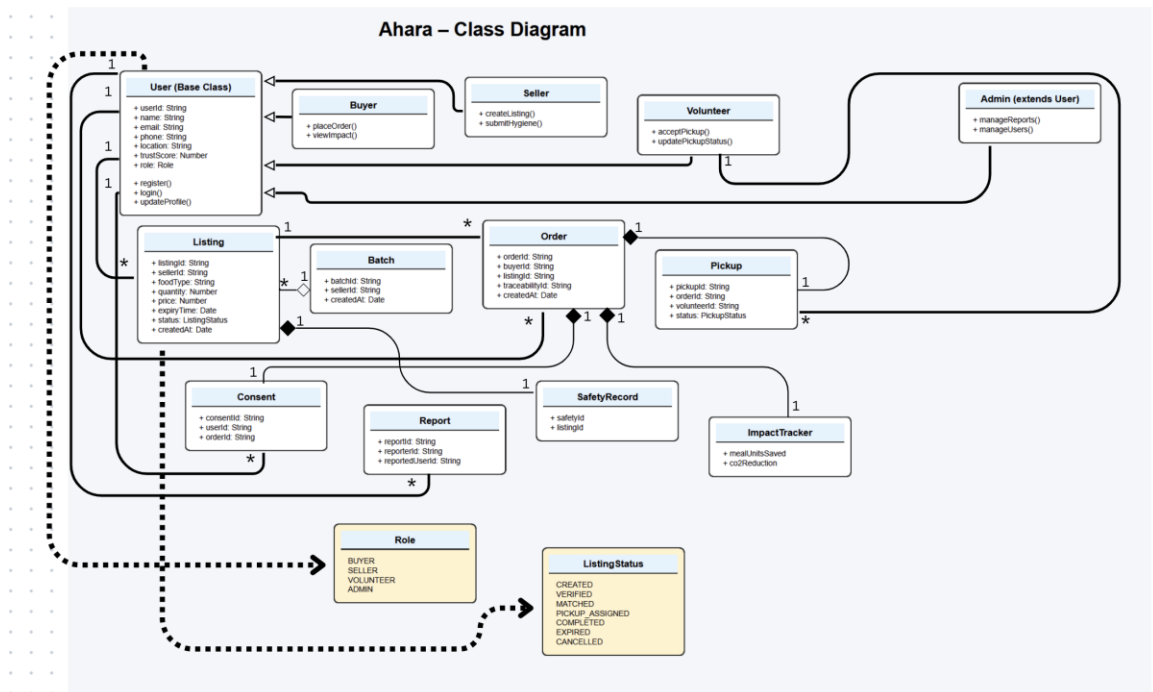Figure 4:

Figure 4

6. Class Diagram

Figure 5: Class Diagram

Figure 5

7. Conclusion & Future Enhancements

The Ahara platform integrates identity management, safety enforcement, logistics optimization, and impact tracking into a unified architecture.