

# Software Requirements Specification (SRS)

**Product:** Food Delivery Platform (Customer, Restaurant Partner, Rider, Admin)

**Version:** 1.0

**Date:** 29 Dec 2025

**Prepared by:** Abbas Hunaid (SQA Analyst – Customer Support)

---

## 1. Introduction

### 1.1 Purpose

This SRS defines functional and non-functional requirements for a multi-tenant food delivery platform similar to Foodpanda, consisting of:

- **Customer app/web**
- **Restaurant Partner portal/app**
- **Rider (courier) app**
- **Admin console**

### 1.2 Scope

The platform enables customers to discover restaurants, place delivery/pickup orders, track riders, and make secure payments. Restaurants manage menus, pricing, availability, and orders. Riders accept/fulfill deliveries. Admin oversees operations, catalog governance, promotions, and compliance.

### 1.3 Definitions, Acronyms

- **COD:** Cash on Delivery
- **ETA:** Estimated Time of Arrival
- **KYC:** Know Your Customer
- **NPS:** Net Promoter Score
- **OTP:** One-Time Password
- **SLA:** Service Level Agreement

### 1.4 References

- IEEE 29148:2018—Systems and software engineering—Life cycle processes—Requirements
  - PCI DSS v4.0 (payments), OWASP ASVS (security), WCAG 2.2 AA (accessibility) *(Standards cited for completeness; actual compliance plans to be defined in project governance.)*
- 

## 2. Overall Description

### 2.1 Product Perspective

Distributed microservices architecture; mobile clients (iOS/Android) and web SPA. Third-party integrations: payment gateways, maps, push notifications, SMS/email, analytics.

### 2.2 User Classes and Characteristics

- **Customer:** Places orders; expects convenience, trust, real-time tracking.
- **Restaurant Partner (Manager/Staff):** Manages catalog, receives/fulfills orders; needs reliability and simple ops.
- **Rider:** Accepts jobs, navigates routes, updates status; needs low battery usage and clear earnings.
- **Admin/Support:** Operations monitoring, issue resolution, promotions; requires auditability and role-based access.

### 2.3 Operating Environment

- **Mobile:** iOS 15+, Android 10+
- **Web:** Latest Chrome, Safari, Edge, Firefox
- **Backend:** Cloud (e.g., Azure/AWS), containerized services, managed DB.

### 2.4 Constraints

- Local payment options (e.g., COD, wallets), regional tax/VAT rules, peak-load events (rain/iftar/new year), data residency (if mandated), fair scheduling for riders.

### 2.5 Assumptions & Dependencies

- Map provider supports live routing and traffic.
  - Payment gateways support 3-D Secure and refunds.
  - SMS providers deliver OTPs within 10s P95.
-

## 3. Functional Requirements

IDs use the form **FR-X.Y** (X = module, Y = requirement). Acceptance Criteria (AC) with measurable outcomes included.

### 3.1 Customer App / Web

#### Discovery & Search

- **FR-C.01:** Browse nearby restaurants by cuisine, rating, ETA, delivery fee.
  - **AC:** Filter by  $\geq 5$  facets; results update within 500ms client-side.
- **FR-C.02:** Full-text search across restaurant names and menu items.
  - **AC:** Typing returns suggestions within 300ms; supports diacritics.

#### Restaurant & Menu

- **FR-C.10:** View restaurant page with hours, delivery area, fees, rating, popular items.
- **FR-C.11:** Menu with categories, item options (sizes), add-ons, dietary tags (veg, halal, gluten-free).
  - **AC:** Option selections update price immediately.

#### Cart & Checkout

- **FR-C.20:** Maintain cart (add/edit/remove) with multi-item and notes (e.g., “no onions”).
- **FR-C.21:** Apply promo codes, vouchers, wallet balance.
  - **AC:** Validation returns success/failure within 800ms.
- **FR-C.22:** Addresses (home, work) with geocoding and delivery instructions.
- **FR-C.23:** Payment methods: COD, cards, local wallets (e.g., Easypaisa/JazzCash), Apple Pay/Google Pay, saved tokens.
  - **AC:** Tokenized storage; no raw PAN retained (PCI DSS).

#### Order Lifecycle

- **FR-C.30:** Place order and receive status updates: *Placed* → *Accepted* → *Being prepared* → *Rider assigned* → *Picked up* → *En route* → *Delivered* → *Completed*.
- **FR-C.31:** Real-time map tracking of rider with ETA.
  - **AC:** Location updates every  $\leq 15s$ ; ETA error  $\leq \pm 3min$  P90.
- **FR-C.32:** Cancel order (policy- and state-dependent).
- **FR-C.33:** Refund workflow (partial/full) on cancellation or issues.

#### Engagement & Support

- **FR-C.40:** Push/SMS/email notifications for key events.
- **FR-C.41:** Rate order (1–5), leave review, photo upload (optional).
- **FR-C.42:** In-app chat with support, templated quick replies, ticket IDs.

- **FR-C.43:** Loyalty: points accrual, tiers, redeemable rewards.

### Account & Compliance

- **FR-C.50:** Registration/login via phone OTP, email/password, social login.
- **FR-C.51:** KYC checks for wallet usage where applicable.
- **FR-C.52:** Privacy controls: data download/delete request (GDPR-style).

## 3.2 Restaurant Partner Portal/App

### Onboarding & Ops

- **FR-R.01:** Partner signup, document upload, bank details, KYC approval.
- **FR-R.02:** Manage operating hours, delivery zones/fees, preparation time.

### Menu & Catalog

- **FR-R.10:** CRUD for categories/items/options/add-ons, images, availability toggles.
- **FR-R.11:** Bulk import/export (CSV/XLSX) with validation reports.

### Order Management

- **FR-R.20:** Receive orders with item list, notes, promised ready time; Accept/Reject with reason.
- **FR-R.21:** Print receipts/tickets (USB/Bluetooth printers).
- **FR-R.22:** Mark states: *Preparing* → *Ready for pickup*.

### Promotions & Insights

- **FR-R.30:** Create discounts, item bundles, happy hours; approval workflow.
- **FR-R.31:** Analytics: orders, AOV, cancellation rate, NPS, payout statements.
- **FR-R.32:** Payouts and ledgers; dispute flows.

## 3.3 Rider App

### Jobs & Routing

- **FR-D.01:** Receive job offers with pickup/drop, fee, distance, ETA; Accept/Decline.
- **FR-D.02:** In-app navigation (deep link to maps), traffic-aware routing.
- **FR-D.03:** Status updates: *On the way* → *Arrived pickup* → *Picked up* → *Arrived drop* → *Delivered*.

### Compliance & Safety

- **FR-D.10:** Rider KYC, vehicle info, documents (CNIC/license), background checks.
- **FR-D.11:** Incident reporting (spills, customer no-show), photo evidence.

## Earnings

- **FR-D.20:** Daily earnings, bonuses, heatmaps, withdrawal to wallet/bank.
- **FR-D.21:** Shift scheduling (optional) and incentives.

## 3.4 Admin Console

### Governance

- **FR-A.01:** Role-based access (RBAC) for Ops, Support, Finance, Catalog.
- **FR-A.02:** CRUD for restaurants, riders, users; ban/suspend/reactivate.
- **FR-A.03:** Promotion approvals and fraud checks.

### Ops & Support

- **FR-A.10:** Live order board, SLAs, escalations; reassign riders/orders.
- **FR-A.11:** Issue management: refunds, partial comps, voucher grants with audit logs.
- **FR-A.12:** Monitoring dashboards: demand/supply balance, pickup delays, cancellations.

### Data & Compliance

- **FR-A.20:** Reports: financial, tax/VAT, partner performance, CSAT/NPS.
  - **FR-A.21:** Data retention policies; PII minimization and masking in tools.
- 

# 4. External Interface Requirements

## 4.1 User Interfaces

- **Customer:** Modern, accessible UI (WCAG 2.2 AA), clear CTAs, skeleton loaders, dark mode.
- **Partner:** Operational clarity, large touch targets, printer settings.
- **Rider:** Glanceable cards, low-latency status toggles, battery-friendly map.
- **Admin:** Data-dense tables, exports, bulk actions, audit trails.

## 4.2 APIs

- REST/JSON over HTTPS, OAuth2/OIDC, rate limiting, idempotent POST for order placement.
- Webhooks for partner POS integration (order events).
- Real-time channels (WebSocket/SSE) for tracking.

## 4.3 Integrations

- **Payments:** Local wallets (Easypaisa/JazzCash), cards (via PCI-compliant gateway), COD reconciliation.
  - **Maps/Geocoding:** Google Maps (traffic, ETAs) or equivalent.
  - **Messaging:** Push (FCM/APNS), SMS providers, email (DKIM/SPF).
- 

## 5. Non-Functional Requirements (NFRs)

### Performance & Scalability

- **NFR-P.01:** P95 API latency  $\leq 500\text{ms}$  for read,  $\leq 800\text{ms}$  for write under normal load.
- **NFR-P.02:** Horizontal scaling to handle  $10\times$  peak order volume; auto-scaling policies.

### Availability & Reliability

- **NFR-A.01:** 99.9% monthly uptime for core ordering; graceful degradation (e.g., temporarily disable promos during incidents).
- **NFR-A.02:** Retry/backoff for transient failures; circuit breakers.

### Security & Privacy

- **NFR-S.01:** OWASP ASVS Level 2; secure storage (no sensitive card data); TLS 1.2+.
- **NFR-S.02:** Role-based authorization, least privilege, audit logs immutable.
- **NFR-S.03:** PII encryption at rest (AES-256), data minimization, retention policies; consent management.

### Compliance

- **NFR-C.01:** PCI DSS for payments (gateway + tokenization).
- **NFR-C.02:** Accessibility WCAG 2.2 AA; local tax compliance (VAT/GST).
- **NFR-C.03:** Support for customer data requests (download/delete).

### Localization & Accessibility

- **NFR-L.01:** Urdu/English support; RTL handling; local currency (PKR), date/time, address formats.
- **NFR-L.02:** Screen reader labels, color contrast, keyboard nav on web.

### Operability & Observability

- **NFR-O.01:** Centralized logging (correlation IDs), metrics (SLIs/SLOs), alerting.
- **NFR-O.02:** Feature flags for rollout and A/B tests.

### Maintainability & Testability

- **NFR-M.01:** 75%+ unit test coverage on critical services; contract tests for APIs.
  - **NFR-M.02:** Staging environment with production-like data masking; synthetic load tests.
- 

## 6. Data Requirements

### 6.1 Core Entities

- **User**(id, name, phone/email, auth, loyalty)
- **Address**(geo, label, instructions)
- **Restaurant**(id, profile, hours, zones, fees, rating)
- **Menu**(categories, items, options/add-ons, tags, price, availability)
- **Order**(id, user, restaurant, items, notes, pricing breakdown, status history, payment, delivery address, rider)
- **Payment**(method, token ref, transaction id, refund id)
- **Rider**(id, KYC, vehicle, status, location)
- **Promotion**(code, discount type, eligibility, validity)
- **Review**(stars, text, photos, moderation)

### 6.2 Data Retention

- Orders: 24 months (configurable); financial records per statutory requirements.
  - Location traces: coarse history for ETA improvements (e.g., 30 days), anonymized thereafter.
  - PII deletion upon user request (subject to legal holds).
- 

## 7. Use Cases (Selected)

### UC-C-01: Place Delivery Order

1. Customer searches and selects restaurant.
2. Adds items/options to cart.
3. Enters address; chooses payment; applies promo.
4. Confirms order; system authorizes payment.
5. Restaurant accepts; rider assigned; tracking begins.
6. Order delivered; customer rates.

### UC-R-02: Restaurant Fulfills Order

1. Receives order; verifies inventory.
2. Accepts; sets ready time.
3. Prepares; marks "Ready."

4. Hands off to rider; confirms pickup.

#### **UC-D-03: Rider Deliver Order**

1. Accepts job; navigates to pickup.
2. Confirms pickup; navigates to drop-off.
3. Marks delivered; completes proof (photo/OTP).

#### **UC-A-04: Admin Issues Refund**

1. Opens order in console; validates claim.
  2. Issues partial/full refund; logs reason and actor.
  3. Customer notified; ledger updated.
- 

## **8. Acceptance Criteria (End-to-End)**

- P95 order placement (from “Place Order” click to “Order Placed” response)  $\leq 2.5s$  under normal load.
  - Map ETA error  $\leq \pm 3$  minutes (P90), rider location updates  $\leq 15s$  interval.
  - Promo validation success/failure shown  $\leq 800ms$ .
  - App crash rate  $< 0.5\%$  per release; ANR  $< 0.7\%$  on Android.
  - Accessibility audits pass WCAG AA on top 20 screens.
- 

## **9. Risk Analysis**

- **Peak demand spikes** during weather/holidays. Mitigation: dynamic throttling, surge incentives for riders.
- **Payment failures/fraud**. Mitigation: 3-D Secure, risk scoring, velocity limits.
- **Catalog inaccuracies**. Mitigation: partner SLAs, availability sync, inventory checks.
- **Geo/routing issues**. Mitigation: fallback providers, manual rider instructions.