Software Requirements Specification (SRS)

Logistics and Delivery Management Application

1. Introduction

We are developing a **complete logistics and delivery management solution** that includes:

- A **mobile app** (for **Merchants** and **Riders**)
- A **web-based admin panel** (for **Administrators and Hub Managers**)
- A public Website

This system will allow **merchants** to easily book shipments, **riders** to manage and deliver parcels efficiently, and **admins/Hub** to oversee all operations in real time.

Key Features

- **Real-time Tracking:** Live parcel and rider tracking using Google Maps.
- **Online & Local Payments:** Integration with **Stripe, JazzCash, and EasyPaisa** for secure transactions.
- **Booking & Scheduling:** Merchants can book immediate or future deliveries.
- **QR Code System:** Auto-generated QR codes for shipment labeling and verification.
- **In-app Chat:** Real-time communication between merchants, riders, and admin via **chat support**.
- **Wallet System:** Manage COD payments and rider earnings through in-app wallets.
- **Multi-language Support:** Full **English and Urdu** (RTL) support across all platforms.
- **Admin Dashboard:** Oversee users, hubs, and delivery operations; manage invoices, rider cash submissions, analytics, and complete system configurations.
- **Notifications:** Instant push notifications and SMS/email updates for delivery statuses.

Deliverables

- **Merchant & Rider Mobile Apps** (Flutter – Android & iOS)
- **Admin Web Dashboard** (Next.js or Laravel)
- **Public Website (Laraval)**
- **Secure Backend API** with real-time updates and analytics

1.1 Purpose

The purpose of this document is to define the complete functional and non-functional requirements for the Logistics and Delivery Management Application. This SRS serves as a guide for developers, project managers, quality assurance teams, and stakeholders to understand the scope, objectives, and expected system behavior of the proposed platform. The application aims to provide a comprehensive delivery ecosystem — connecting merchants (senders), riders (delivery agents), and administrators through an integrated digital system, ensuring real-time tracking, automated invoicing, chat support, and payment management.

1.2 Scope

The system facilitates end-to-end logistics management across three core user roles:

- Merchants – Individuals or businesses who book shipments and monitor delivery status.
- Riders – Delivery agents who pick up, transport, and deliver parcels.
- Admins – Supervisors who manage operations, assign deliveries, monitor performance, and oversee financial activities.

The system will be accessible through:

- A mobile application (Flutter-based for Android and iOS) for Riders and Merchants.
- A web-based admin dashboard (Next.js or Laravel-based) for administrators.

The app will support real-time updates, cash-on-delivery (COD) tracking, wallet systems, Urdu/English language support, and local payment gateways (JazzCash, EasyPaisa).

1.3 Definitions, Acronyms, and Abbreviations

Term	Description
COD	Cash on Delivery
API	Application Programming Interface
KYC	Know Your Customer
OTP	One-Time Password
SRS	Software Requirements Specification

2. Overall Description

2.1 Product Perspective

The Logistics Delivery Application is a multi-platform delivery management solution integrating tracking, booking, payments, and analytics into a unified ecosystem. It will be cloud-hosted and will communicate via RESTful APIs between mobile and web interfaces. The architecture includes:

- Frontend (Mobile & Web) – Built with Flutter and Next.js.
- Backend – Built with Laravel or Node.js (Express).
- Database – MySQL or Firebase for data storage.
- Integration Services – Google Maps API, Firebase Messaging, and Payment Gateways.

2.2 Product Functions

- Booking and scheduling deliveries.
- Real-time GPS-based tracking of parcels and riders.
- QR code-based shipment verification.
- Secure payment processing via multiple gateways.
- In-app chat between users (Merchant ↔ Rider, Merchant ↔ Admin).
- Role-based access control and analytics for administrators.
- Notifications and invoice generation.

2.3 User Characteristics

User Type	Description	Access Platform
Merchant	Books deliveries, views shipment status, and manages payments.	Mobile App / Web
Rider	Accepts deliveries, completes pickups/drop-offs, manages earnings.	Mobile App
Admin	Manages users, operations, payments, and analytics.	Web Dashboard

3. System Features

3.1 Merchant Module

Core Features:

1. User Management
   - o Register/login using Email, Phone, Google, or Apple ID.
   - o OTP verification for phone numbers.
   - o Manage personal and business profiles.

2. Shipment Management
   - o Create new shipments (sender & receiver details).
   - o Add parcel dimensions, weight, and instructions.
   - o Schedule immediate or future deliveries.
   - o Generate and print QR shipment labels.

3. Real-Time Tracking
   - o Live map view of rider location.
   - o Delivery timeline and ETA updates.

4. Financial Management
   - o View invoices, payment history, and wallet balance.
   - o Fund wallet using Stripe, JazzCash, or EasyPaisa.
   - o Manage COD settlements.

5. Communication & Notifications
   - o In-app chat with riders or support.
   - o Real-time push notifications for delivery updates.
   - o Email & SMS confirmations.

6. Ratings and Feedback
   - o Rate riders and provide reviews after deliveries.

7. Multi-language Interface
   - o Support for both English and Urdu (RTL layout).

3.2 Rider Module

Core Features:

1. Registration & Verification
   - o Sign up using Email or Phone.
   - o Submit documents for KYC verification (CNIC, License).
   - o Admin approval workflow.

2. Task Management
   - o View available and assigned deliveries.
   - o Accept or reject tasks.
   - o Mark pickup and delivery completion.

3. Navigation & Tracking
   - o Real-time optimized route suggestions.
   - o Integrated Google Maps navigation.
   - o GPS tracking visible to admin and merchant.

4. Earnings & Payments
   - o COD tracking and daily collection summary.
   - o Wallet system for earnings and cash flow.

5. QR Code Scanning
   - o Scan shipment codes for pickup/drop-off confirmation.

6. Chat & Support
   - o Communicate with merchants or admin through in-app chat.

7. Multi-language Interface
   - o English and Urdu support.

3.3 Hub-Based Admin Module (Simplified)

The Admin Panel now includes a **Hub System**, allowing better control over parcel movement. Admins can manage everything, while each **Hub Manager** handles deliveries and riders in their own area.

Main Features

1. User & Hub Management

- Add or remove Merchants, Riders, and Hub Managers.
- Assign hubs to specific cities or areas.
- Approve rider documents (CNIC, license, etc.).
- Move parcels or deliveries between hubs if needed.

2. Hub Operations

- Riders collect parcels from merchants during the day.
- At the end of the day, riders drop all parcels at their assigned hub.
- The hub team sorts parcels by area and prepares them for delivery.
- Admin or Hub Manager assigns parcels to riders for next-day delivery.
- Manage return or failed deliveries.

3. Delivery & Tracking

- Track all riders and parcels in real-time.
- Manage unassigned, delayed, or failed deliveries.
- Control pricing and delivery zones.

4. Payments & Finance

- Track rider cash collections and COD submitted at the hub.
- Generate invoices and receipts for merchants.
- Manage wallet top-ups, refunds, and payouts.
- View daily or monthly financial reports.

5. Reports & Analytics

- View key stats: active riders, completed deliveries, hub performance, total revenue.
- Download reports by day, week, or month.

6. Support & Settings

- Handle customer complaints and rider issues.
- Send notifications to hubs, riders, or merchants.
- Manage app content, FAQs, and announcements.
- Control payment gateways (Stripe, JazzCash, EasyPaisa) and language (Urdu + English).

4. System-Wide Features

- Real-Time Tracking: Rider and shipment tracking via Google Maps API.
- Payment Integrations: Stripe (global), JazzCash, EasyPaisa (local).
- Chat System: Powered by Socket.io or Firebase Realtime Database.
- Push Notifications: Firebase Cloud Messaging integration.
- Multi-language Support: English and Urdu with right-to-left UI.
- QR Code System: Shipment generation and verification.
- Analytics Dashboard: Graphical insights for admin and merchants.

- Security: OTP-based login, role-based permissions, encrypted API requests.

5. Non-Functional Requirements

Requirement	Description
Performance	Must handle 10,000+ concurrent users with minimal latency.
Scalability	Auto-scaling architecture with horizontal scaling support.
Security	HTTPS communication, JWT authentication, encrypted data storage.

Availability	Minimum 99.5% uptime with redundant failover mechanisms.
Usability	Intuitive UX for all user types with consistent design language.
Maintainability	Modular structure and RESTful architecture for easy upgrades.
Localization	Support RTL layouts and Urdu fonts across mobile and web.

6. Technology Stack

Layer	Technology
Frontend (Mobile)	Flutter
Frontend (Web)	Next.js
Backend	Laravel / Node.js (Express)
Database	MySQL / Firebase Realtime DB
Real-Time Chat	Socket.io / Firebase
Maps & Tracking	Google Maps API
Notifications	Firebase Cloud Messaging
Payment Gateways	Stripe, JazzCash, EasyPaisa
Hosting	AWS / Firebase / DigitalOcean

Payment Terms, Timeline & Budget by Phase

Payment Terms

Total Project cost : 4 lac

- 30% Advance: **1,20,000 PKR** (Due upon contract signing)
- The remaining cost after 30% would be **2 lac 80k**.
- The cost of each phase after 30% is tabulated below.

Timeline & Budget by Phase

S.No	Phase	Timeline	Payment (PKR)
2	Project Designing	Weeks 1-2	30,000
3	Development	Weeks 3- 12	2 lac
4	Testing	Weeks 13-14	20,000
5	Deployment	Weeks 15-16	30,000
6	API	-	200 to 400$
Total **Note :-** We expect to deliver the project before the originally agreed-upon deadline.		**4 Months**	**2 lac 80k PKR**

- Bi-weekly updates on progress will be submitted throughout every phase.
- Payments are required to be remitted phase-wise; upon completion and acceptance of each phase, the corresponding payment will be processed before we proceed to the subsequent phase.

Logistics and Delivery Management Application - Support & Maintenance

Support & Maintenance

- Three months of free support after implementation (including bug fixes and remote assistance).
- Optional extended support: $150/month (24/7 priority response).

Looking Forward

- Sign-off on this proposal.
- Contract finalization and advance payment.
- Kickoff meeting to align on priorities.

Logistics and Delivery Management Application - MUTUAL NON-DISCLOSURE AGREEMENT (NDA)

ZIMLITECH (SMC-PRIVATE) LIMITED (referred to as "Party A") and

(referred to as "Party B"). Each Party may act as a "Disclosing Party" or a "Receiving Party" depending on the context of information exchange.

Purpose of Disclosure:

The Parties agree to exchange confidential information solely for the purpose of exploring, evaluating, and performing potential or existing services related to projects, including the " Logistics and Delivery Management Application " and any associated long-term services agreements.

Obligations of Each Party (as Receiving Party):

When receiving Confidential Information from the other Party, each Party agrees to:

Hold all such information in strict confidence.

Use the information solely for the agreed-upon project purpose.

Not disclose it to any unauthorized third parties.

Limit access only to personnel who have a "need to know" and are bound by similar confidentiality obligations.

What is NOT Confidential (Exclusions):

Information is not confidential if it is already public without fault of the Receiving Party, was known prior to disclosure, developed independently, or legally received from a third party without restrictions. Information legally required to be disclosed by law is also excluded, provided prior notice is given to the Disclosing Party.

Term of Confidentiality:

These confidentiality obligations typically remain in effect for a specified period (3 years) from the Effective Date, or indefinitely for trade secrets, even if the primary services agreement terminates.

Return or Destruction:

Upon request or termination of the agreement, the Receiving Party must promptly return or destroy all received Confidential Information.

Logistics and Delivery Management Application Development-Contract Sign-off

IN WITNESS WHEREOF, the Parties, through their duly authorized representatives, have executed this Agreement as of the Effective Date.

zimlitech

ZIMLITECH (SMC-PRIVATE) LIMITED ____________________

M Asif

Signature____________ **Signature**____________

Muhammad Asif
CEO
Zimli Technologies

info@zimlitech.com