Namal University, Mianwali
Department of Computer Science Software

Requirements Specification

for

FareShare

A Ride-Sharing and Fare-Sharing System

Version 2.0 - Complete Edition

Team FareShare

Muhammad Naveed	NUM-BSCS-2024-54	Group Lead
Munawar Ali	NUM-BSCS-2024-60	Group Member
Areeba Tahir	NUM-BSCS-2024-15	Group Member

Instructor: Asiya Batool
Requirement Provider: Rana Muhammad Adeel

Submission Date: January 11, 2026

Contents

List of Figures

List of Tables

1 Introduction

1.1 Purpose

This Software Requirements Specification (SRS) document provides a comprehensive description of the FareShare ride-sharing and fare-sharing system. This document serves as the definitive source of requirements for all stakeholders including developers, testers, project managers, and the requirement provider.

Intended Audience:

- **Development Team:** For system design, implementation, and testing
- **Quality Assurance Team:** For developing test plans and cases
- **Project Managers:** For project planning and tracking
- **Requirement Provider:** For validation and verification
- **Academic Evaluation:** For course assessment and grading
- **Future Maintenance Team:** For system enhancements and support

This document adheres to **IEEE Std 830-1984** guidelines for Software Requirements Specifications.

1.2 Scope

FareShare is an innovative transportation platform designed for the Pakistani market, featuring unique fare-sharing capabilities alongside traditional ride-hailing services.

1.2.1 System Boundaries

Table 1: System Inclusions and Exclusions

Aspect	Included (IN-SCOPE)	Excluded (OUT-OF-SCOPE)
Payment Methods	Cash-only transactions	Digital payments, credit cards, mobile wallets
User Types	Riders, Drivers, Administrators	Corporate accounts, fleet managers
Features	Ride booking, fare-sharing, SOS, ratings, tracking	Food delivery, package delivery, ride scheduling
Geography	Initial: Mianwali region, Expandable to other Pakistani cities	International operations
Technology	Mobile apps (Android/iOS), Web admin panel, Cloud backend	IoT integration, blockchain, cryptocurrency
Verification	Manual document verification by administrators	Automated background checks, biometric verification

1.2.2 Business Objectives

1. Reduce transportation costs by 30-50% through fare-sharing
2. Provide fair income to drivers with 15% commission (vs. 25-30% competitors)
3. Achieve 95% rider satisfaction through transparent pricing
4. Ensure 100% driver verification before ride acceptance
5. Maintain 99.5% system availability during operational hours
6. Process 10,000 rides monthly within first year of operation
7. Achieve 4.0+ average rating on app stores

1.3 Definitions, Acronyms, and Abbreviations

Table 2: Complete Terminology Reference

Term	Definition
SRS	Software Requirements Specification - Formal document defining system requirements
Rider	End-user requesting transportation services (passenger)
Driver	Verified vehicle operator providing transportation services
Admin	System administrator managing operations and verifications
Primary Rider	Initial rider who books and initiates a ride
Secondary Rider	Additional rider joining through fare-sharing feature
Fare-Sharing	Feature allowing multiple riders to share ride and split cost
SOS	Emergency alert system for rider/driver safety
CNIC	Computerized National Identity Card (Pakistani national ID)
OTP	One-Time Password for authentication
GPS	Global Positioning System for location tracking
ETA	Estimated Time of Arrival
FCM	Firebase Cloud Messaging for push notifications
JWT	JSON Web Token for secure authentication
API	Application Programming Interface
REST	Representational State Transfer architecture
NoSQL	Non-relational database (MongoDB)
UI/UX	User Interface/User Experience
TLS	Transport Layer Security encryption

1.4 Document Overview

- **Section 2:** General system description and context
- **Section 3:** Comprehensive functional requirements by module
- **Section 4:** Complete non-functional requirements
- **Section 5:** External interface specifications
- **Section 6:** System models and diagrams
- **Section 7:** Appendices and supplementary information

2 General Description

2.0.1 System Components

Table 3: System Component Description

Component	Technology	Responsibility
Rider Mobile App	React Native (Android/iOS)	Booking, tracking, payments, safety features
Driver Mobile App	React Native (Android/iOS)	Ride acceptance, navigation, earnings management
Admin Web Panel	React.js + Bootstrap	User management, verification, analytics
Backend Server	Node.js/Python Django	Business logic, APIs, database operations
Database	MongoDB + Redis Cache	Data persistence, session management
Real-time Service	Socket.io WebSockets	Live tracking, notifications
External APIs	Google Maps, SMS Gateway	Maps, routes, OTP, emergency alerts
Cloud Infrastructure	AWS/DigitalOcean	Hosting, scalability, storage

2.1 Product Functions

The system provides comprehensive functionality across all user roles:

2.1.1 Complete Functional Matrix

Table 4: Functionality by User Role

Module	Rider Functions	Driver Functions	Admin Functions
Authentication	Register, Login, Profile Setup	Register, Document Upload, Verification	Login, Role Management
Ride Management	Book, Cancel, Track, Rate	Accept, Navigate, Complete, Rate	Monitor, Intervene, Report
Fare-Sharing	Enable, Search, Join, Split Fare	Approve, Manage Multiple Pickups	Monitor, Adjust Rules
Safety	SOS, Share Ride, Emergency Contacts	SOS, Verify Riders	Emergency Response, Alert Management
Payments	Cash Payment, Receipt View	Cash Collection, Earnings View	Revenue Reports, Commission Tracking
Administration	Profile Management	Vehicle Management, Availability	User Management, System Configuration
Analytics	Ride History, Spending Reports	Earnings Reports, Performance Stats	System Analytics, Business Intelligence

2.2 User Characteristics

2.2.1 Rider Profile

Table 5: Rider User Characteristics

Characteristic	Description
Age Range	16-65 years (16-18 require parental consent)
Education	High school to postgraduate
Technical Proficiency	Basic to advanced smartphone usage
Frequency of Use	Daily commuters to occasional users
Primary Needs	Affordability, safety, reliability, convenience
Payment Preference	Cash-based (aligned with local economy)
Language	Urdu primary, English secondary
Device Ownership	95% smartphone penetration in target demographic

2.2.2 Driver Profile

Table 6: Driver User Characteristics

Characteristic	Description
Age Range	21-60 years (legal driving age 18+)
Education	Matriculation to intermediate typically
Technical Proficiency	Basic smartphone usage, navigation apps
Vehicle Ownership	Personal or authorized commercial vehicles
Income Motivation	Primary or supplementary income source
Working Hours	Flexible, part-time to full-time
Safety Concerns	Personal safety, vehicle security, fair payment
Training Required	App usage, navigation, customer service

2.2.3 Administrator Profile

Table 7: Administrator User Characteristics

Characteristic	Description
Education	Bachelor's in Computer Science or related field
Technical Skills	Database management, system monitoring, analytics
Responsibilities	User verification, dispute resolution, system maintenance
Work Schedule	Rotational shifts for 24/7 coverage
Decision Authority	Document approval, user suspension, fare adjustments
Tools Proficiency	Web interfaces, analytics dashboards, communication tools

2.3 General Constraints

2.3.1 Technical Constraints

Table 8: Technical System Constraints

Constraint Category	Constraint Description
Platform Compatibility	Android 8.0+, iOS 14.0+, Web browsers (Chrome 90+, Firefox 88+)
Device Requirements	GPS, Camera, 4G/LTE, Minimum 2GB RAM, 100MB storage
Network Requirements	Minimum 3G, Recommended 4G/LTE, GPS signal availability
Performance Limits	Max 500 concurrent rides, 10,000 active users
Data Storage	User data: 5 years retention, Ride data: 90 days active, 5 years archive
Third-party Dependencies	Google Maps API (critical), SMS Gateway, FCM
Security Compliance	TLS 1.3+, AES-256 encryption, Pakistani data protection laws

2.3.2 Business Constraints

Table 9: Business and Operational Constraints

Constraint Category	Constraint Description
Geographic Limit	Initial: Mianwali region (50km radius)
Service Hours	24/7 operation with reduced support 12AM-5AM
Payment Model	Cash-only, no digital payments or wallets
Commission Structure	Fixed 15% commission from driver earnings
Driver Requirements	Valid license, vehicle registration, insurance, CNIC
Rider Requirements	Minimum age 16, phone verification, emergency contacts
Legal Compliance	Pakistan transport regulations, data protection laws
Support Capacity	100 support tickets/day maximum

2.4 Assumptions and Dependencies

2.4.1 Critical Assumptions

Table 10: System Assumptions

Assumption Category	Assumption Details
User Behavior	Users will grant location permissions, maintain app updates
Market Conditions	Sufficient driver/rider adoption in target areas
Infrastructure	Reliable internet and GPS coverage in service areas
Legal Environment	Current transport regulations remain favorable
Technical Environment	External APIs remain available and affordable
Economic Factors	Cash economy remains dominant in target market
Safety	Users will provide accurate emergency contact information

2.4.2 Critical Dependencies

Table 11: System Dependencies

Dependency	Impact if Unavailable	Mitigation Strategy
Google Maps API	Core functionality fails	Cache maps, offline fallback routes
SMS Gateway	User verification impossible	Email fallback, manual verification
Internet Connectivity	Real-time features disabled	Offline mode with sync on reconnect
GPS Signal	Location tracking fails	Network-based location estimation
App Store Approval	Cannot distribute updates	Progressive Web App alternative
Payment Cash Flow	Drivers cannot earn	Daily cash settlements, emergency fund

3 Specific Requirements - Functional

3.1 Authentication & User Management Module

3.1.1 Rider Authentication Requirements

Table 12: Rider Authentication Requirements

Req ID	**FR-AUTH-01**	**Requirement Type**	**Mandatory**
Feature Name	Rider Registration via Phone Number	Priority	Critical
Description	System shall allow new riders to register using Pakistani mobile number		
Preconditions	User has valid Pakistani mobile number (+92 format)		
Inputs	Mobile number, User role selection (Rider)		
Processing	1. Validate number format 2. Check existing registration 3. Generate 6-digit OTP 4. Send OTP via SMS 5. Create unverified user record		
Outputs	OTP sent confirmation, Pending verification status		
Postconditions	User can proceed to OTP verification		
Error Handling	Duplicate number: "Number already registered" Invalid format: "Please enter valid Pakistani number" SMS failure: "Could not send SMS, try again"		

Table 13: Rider OTP Verification Requirements

Req ID	**FR-AUTH-02**	**Requirement Type**	**Mandatory**
Feature Name	OTP Verification	Priority	Critical
Description	System shall verify rider's mobile number via OTP		
Preconditions	OTP sent to valid mobile number		
Inputs	6-digit OTP entered by user		
Processing	1. Retrieve stored OTP for session 2. Compare with entered OTP 3. Check OTP expiration (2 minutes) 4. Generate JWT access token 5. Update user status to verified		
Outputs	Access token, Rider dashboard, Verification success		
Postconditions	User can access all rider features		
Error Handling	Invalid OTP: "Incorrect code, try again" (3 attempts max) Expired OTP: "Code expired, request new OTP"		

3.1.2 Driver Authentication Requirements

Table 14: Driver Registration Requirements

Req ID	**FR-AUTH-03**	**Requirement Type**	**Mandatory**
Feature Name	Driver Registration with Documents	Priority	Critical
Description	System shall register drivers with mandatory document submission		
Preconditions	User selects Driver role, has required documents		
Inputs	Mobile number, CNIC images (front/back), Driving license, Vehicle registration		
Processing	1. Perform rider registration steps 2. Upload document images to secure storage 3. Validate image formats and sizes 4. Create verification ticket for admin 5. Set status: "Pending Verification"		
Outputs	Registration success, Documents under review message		
Postconditions	Driver can login but cannot accept rides until verified		
Error Handling	Invalid document: "Please upload clear images" File too large: "Maximum 5MB per image"		

3.1.3 Administrator Authentication Requirements

Table 15: Admin Authentication Requirements

Req ID	**FR-AUTH-04**	**Requirement Type**	**Mandatory**
Feature Name	Administrator Login	Priority	Critical
Description	System shall provide secure admin authentication		
Preconditions	Admin account created by super admin		
Inputs	Username/Email, Password, 2FA code (if enabled)		
Processing	1. Validate credentials against admin database 2. Check account status (active/suspended) 3. Verify 2FA if enabled 4. Generate admin JWT with elevated privileges 5. Log login attempt with IP address		
Outputs	Admin dashboard, Session token, Login timestamp		
Postconditions	Admin can access management features based on role		
Error Handling	Invalid credentials: "Access denied" Account locked: "Contact super administrator" 2FA failure: "Invalid authentication code"		

3.1.4 Profile Management Requirements

Table 16: User Profile Management Requirements

Req ID	**FR-AUTH-05**	**Requirement Type**	**Mandatory**
Feature Name	Profile Management	Priority	High
Description	Users shall manage their personal information		
Preconditions	User is logged in and verified		
Inputs	Name, Email, Profile photo, Emergency contacts		
Processing	1. Validate input data format 2. Upload profile photo to cloud storage 3. Update user record in database 4. Verify emergency contact numbers 5. Send confirmation to updated email (if changed)		
Outputs	Profile updated confirmation, Updated profile view		
Postconditions	Changes reflected across system		
Error Handling	Invalid email: "Please enter valid email" Duplicate emergency contact: "Contact already exists" Photo upload failed: "Could not upload image"		

3.2 Ride Booking & Management Module

3.2.1 Ride Booking Requirements

Table 17: Ride Booking Requirements

Req ID	**FR-BOOK-01**	**Requirement Type**	**Mandatory**
Feature Name	Location Selection	Priority	Critical
Description	Rider shall select pickup and drop-off locations		
Preconditions	Rider is logged in, GPS enabled, Internet available		
Inputs	Pickup location (auto-detected or manual), Drop-off location		
Processing	1. Get current location via GPS 2. Provide address search with autocomplete 3. Validate locations within service area 4. Calculate route and display on map 5. Show pickup pin, drop-off pin, route line		
Outputs	Map with locations marked, Route distance and time		
Postconditions	Ready for vehicle selection and fare estimation		
Error Handling	Outside service area: "Service not available here" Invalid address: "Could not find location"		

Table 18: Fare Estimation Requirements

Req ID	**FR-BOOK-02**	**Requirement Type**	**Mandatory**
Feature Name	Fare Estimation	Priority	Critical
Description	System shall provide fare estimate before booking		
Preconditions	Valid pickup and drop-off locations selected		
Inputs	Pickup coordinates, Drop-off coordinates, Vehicle type		
Processing	1. Calculate route distance and time via Google Maps 2. Retrieve pricing for vehicle type 3. Apply formula: Base + (Distance × Rate) + (Time × Rate) 4. Add 15% commission 5. Round to nearest PKR 10		
Outputs	Estimated fare range (e.g., PKR 200-250)		
Postconditions	Rider can decide to proceed with booking		
Error Handling	Route calculation failed: "Could not calculate fare" Pricing data missing: "Service temporarily unavailable"		

3.2.2 Driver Matching Requirements

Table 19: Driver Matching Requirements

Req ID	**FR-BOOK-03**	**Requirement Type**	**Mandatory**
Feature Name	Driver Search and Matching	Priority	Critical
Description	System shall find and match nearest available drivers		
Preconditions	Ride request created with valid parameters		
Inputs	Rider location, Vehicle type preference, Rider rating		
Processing	1. Query geospatial index for online drivers within 5km 2. Filter by vehicle type availability 3. Sort by: Proximity (60%), Rating (25%), Acceptance rate (15%) 4. Select top 3 drivers 5. Send push notification to selected drivers		
Outputs	Driver match results, Notifications sent		
Postconditions	Drivers have 30 seconds to accept		
Error Handling	No drivers available: "No drivers nearby, try again" All drivers decline: "Could not find driver, please retry"		

3.2.3 Ride Acceptance Requirements

Table 20: Ride Acceptance Requirements

Req ID	**FR-BOOK-04**	**Requirement Type**	**Mandatory**
Feature Name	Driver Ride Acceptance	Priority	Critical
Description	Driver shall accept ride requests within time limit		
Preconditions	Driver is online and available, Notification received		
Inputs	Accept/Decline button press		
Processing	1. Check if ride still available (not taken) 2. Assign driver to ride record 3. Update ride status to "Accepted" 4. Notify rider with driver details 5. Start navigation to pickup		
Outputs	Ride acceptance confirmation, Navigation started		
Postconditions	Ride moves to active state, Driver status: Busy		
Error Handling	Ride already taken: "Ride no longer available" Timeout: "Request expired"		

3.2.4 Ride Cancellation Requirements

Table 21: Ride Cancellation Requirements

Req ID	**FR-BOOK-05**	**Requirement Type**	**Mandatory**
Feature Name	Ride Cancellation	Priority	High
Description	Users shall cancel rides with appropriate handling		
Preconditions	Ride exists and is cancellable (not completed)		
Inputs	Cancellation request with reason		
Processing	**Rider Cancellation:** 1. Check cancellation window (before driver arrives) 2. Apply cancellation fee if applicable 3. Notify driver 4. Update ride status to "Cancelled" **Driver Cancellation:** 1. Check if pickup not reached 2. Apply penalty to driver rating 3. Notify rider and find alternative driver		
Outputs	Cancellation confirmation, Fee/Penalty applied		
Postconditions	Ride closed, Users returned to available state		
Error Handling	Invalid cancellation: "Cannot cancel at this stage" Multiple cancellations: "Excessive cancellations may limit service"		

3.3 Fare-Sharing Module

3.3.1 Fare-Sharing Activation Requirements

Table 22: Fare-Sharing Activation Requirements

Req ID	**FR-SHARE-01**	**Requirement Type**	**Enhanced**
Feature Name	Enable Fare-Sharing	Priority	High
Description	Primary rider shall enable fare-sharing for their ride		
Preconditions	Ride accepted by driver, Not yet started		
Inputs	”Enable Sharing” toggle, Maximum passengers		
Processing	1. Check vehicle capacity for additional riders 2. Calculate potential fare reduction 3. Get driver approval 4. Make ride visible to nearby riders 5. Update ride as ”Sharable”		
Outputs	Sharing enabled confirmation, Updated fare estimate		
Postconditions	Ride appears in shared rides search		
Error Handling	Driver rejected: ”Driver declined sharing request” Vehicle full: ”Maximum capacity reached”		

3.3.2 Shared Ride Search Requirements

Table 23: Shared Ride Search Requirements

Req ID	**FR-SHARE-02**	**Requirement Type**	**Enhanced**
Feature Name	Search Shared Rides	Priority	High
Description	Riders shall search for available shared rides		
Preconditions	Rider logged in, Location services enabled		
Inputs	Current location, Destination, Time preference		
Processing	1. Query sharable rides within 2km radius 2. Filter by direction compatibility (15% route deviation) 3. Sort by: Proximity (40%), Fare saving (40%), ETA (20%) 4. Calculate individual fare for each option 5. Display with pickup ETA and savings		
Outputs	List of available shared rides with details		
Postconditions	Rider can select and request to join		
Error Handling	No shared rides: "No shared rides available, book individual"		

3.3.3 Join Shared Ride Requirements

Table 24: Join Shared Ride Requirements

Req ID	**FR-SHARE-03**	**Requirement Type**	**Enhanced**
Feature Name	Join Shared Ride	Priority	High
Description	Secondary rider shall request to join shared ride		
Preconditions	Shared ride available, Seats remaining		
Inputs	Join request with pickup location		
Processing	1. Validate pickup is on route (within 500m deviation) 2. Send join request to primary rider and driver 3. Wait for approvals (60 seconds timeout) 4. If approved: Add rider, Recalculate fares 5. Update all riders with new details		
Outputs	Join request sent, Approval pending		
Postconditions	Ride continues with additional passenger		
Error Handling	Request rejected: "Join request declined" Timeout: "No response, request cancelled"		

3.3.4 Fare Splitting Requirements

Table 25: Fare Splitting Requirements

Req ID	**FR-SHARE-04**	**Requirement Type**	**Enhanced**
Feature Name	Dynamic Fare Splitting	Priority	High
Description	System shall split fare fairly among shared riders		
Preconditions	Shared ride with multiple riders completed		
Inputs	Individual journey segments, Vehicle pricing rates		
Processing	1. Calculate total route distance and time 2. Determine each rider's segment distance/time 3. Apply proportional formula: Fare = Base/n + (Dist$_i$/$TotalDistDistFare$)+ ($Time_i$/$TotalTimeTimeFare$) 4. Ensure minimum 25% savings vs individual ride 5. Round to nearest PKR 10		
Outputs	Individual fare amounts, Total collected amount		
Postconditions	Each rider sees their specific fare		
Error Handling	Calculation error: Use fallback equal splitting		

3.4 Safety & Emergency Module

3.4.1 SOS Emergency System Requirements

Table 26: SOS Emergency Requirements

Req ID	**FR-SAFE-01**	**Requirement Type**	**Critical**
Feature Name	SOS Emergency Alert	Priority	Critical
Description	Users shall trigger emergency alerts during rides		
Preconditions	Active ride in progress, Network connectivity		
Inputs	SOS button press (3-second hold)		
Processing	1. Capture current GPS location 2. Retrieve ride details and user information 3. Send SMS to emergency contacts: "EMERGENCY: [User] needs help. Location: [Link]. Ride: [Details]" 4. Notify FareShare emergency response team 5. Start continuous location tracking (every 30 seconds) 6. Record emergency in audit log		
Outputs	SOS activated confirmation, Alerts sent, Admin notified		
Postconditions	Emergency response initiated, Ride flagged		
Error Handling	SMS failed: Retry 3 times, then store locally No network: Store alert, send when connectivity resumes		

3.4.2 Emergency Contact Management

Table 27: Emergency Contact Requirements

Req ID	**FR-SAFE-02**	**Requirement Type**	**Mandatory**
Feature Name	Emergency Contact Management	Priority	High
Description	Users shall manage emergency contacts		
Preconditions	User logged in and verified		
Inputs	Contact name, Phone number, Relationship		
Processing	1. Validate phone number format 2. Check duplicate entries 3. Send verification SMS to contact 4. Store verified contact 5. Limit: 3 emergency contacts maximum		
Outputs	Contact saved, Verification sent		
Postconditions	Contact available for emergency alerts		
Error Handling	Invalid number: "Please enter valid number" Verification failed: "Could not verify contact"		

3.4.3 Ride Sharing Safety Requirements

Table 28: Ride Sharing Safety Requirements

Req ID	**FR-SAFE-03**	**Requirement Type**	**Mandatory**
Feature Name	Ride Sharing with Trusted Contacts	Priority	High
Description	Users shall share ride details with trusted contacts		
Preconditions	Active or upcoming ride		
Inputs	Share button, Contact selection		
Processing	1. Generate ride sharing link with live tracking 2. Send via SMS/WhatsApp: "I'm taking a FareShare ride. Track me: [Link]" 3. Include: Driver details, Vehicle info, Route, ETA 4. Enable contact to view real-time location		
Outputs	Sharing link generated, Message sent		
Postconditions	Contact can monitor ride progress		
Error Handling	Sharing failed: "Could not share ride details"		

3.4.4 Driver Safety Verification

Table 29: Driver Safety Verification Requirements

Req ID	**FR-SAFE-04**	**Requirement Type**	**Critical**
Feature Name	Driver Background Verification	Priority	Critical
Description	System shall ensure driver safety through verification		
Preconditions	Driver registration submitted		
Inputs	CNIC, License, Registration documents		
Processing	1. Admin manually verifies document authenticity 2. Check for any previous violations or bans 3. Verify vehicle fitness and insurance 4. Set verification status: Approved/Rejected 5. Record verification details and admin ID		
Outputs	Verification result, Driver status updated		
Postconditions	Driver can go online (if approved)		
Error Handling	Documents unclear: Request better quality images Verification failed: Notify driver with reason		

3.5 Administrative Module

3.5.1 User Management Requirements

Table 30: User Management Requirements

Req ID	**FR-ADMIN-01**	**Requirement Type**	**Mandatory**
Feature Name	Comprehensive User Management	Priority	High
Description	Admin shall manage all system users		
Preconditions	Admin logged in with user management permissions		
Inputs	Search filters, User actions (suspend, verify, delete)		
Processing	1. Search users by: Phone, Name, Status, Date 2. View user details: Profile, Rides, Ratings, Documents 3. Perform actions: Verify, Suspend, Delete, Warn 4. Record admin action with reason 5. Notify user of significant changes		
Outputs	User list, Action confirmation, Audit log entry		
Postconditions	User status updated, Affected rides handled		
Error Handling	Invalid action: "Cannot perform this action" User not found: "User does not exist"		

3.5.2 Driver Verification Requirements

Table 31: Driver Verification Requirements

Req ID	**FR-ADMIN-02**	**Requirement Type**	**Critical**
Feature Name	Driver Document Verification	Priority	Critical
Description	Admin shall verify driver documents manually		
Preconditions	Driver submitted documents, Pending verification		
Inputs	Document images, Verification decision, Comments		
Processing	1. Display documents in verification interface 2. Check: CNIC validity, License authenticity, Registration 3. Make decision: Approve, Reject, Request more info 4. If reject: Specify reason, Block ride acceptance 5. If approve: Enable driver to go online 6. Send notification to driver		
Outputs	Verification status updated, Driver notified		
Postconditions	Driver can (or cannot) accept rides		
Error Handling	Documents missing: Request complete set Verification conflict: Escalate to senior admin		

3.5.3 Emergency Response Requirements

Table 32: Emergency Response Requirements

Req ID	**FR-ADMIN-03**	**Requirement Type**	**Critical**
Feature Name	Emergency Alert Management	Priority	Critical
Description	Admin shall monitor and respond to emergency alerts		
Preconditions	SOS alert triggered by user		
Inputs	Emergency alert details, Response actions		
Processing	1. Display real-time emergency dashboard 2. Show: User details, Location, Ride info, Contact info 3. Enable: Call user, Call driver, Contact emergency services 4. Track response actions and timeline 5. Mark as resolved when situation handled		
Outputs	Emergency interface, Response log, Resolution status		
Postconditions	Emergency handled, Follow-up actions recorded		
Error Handling	Cannot reach user: Escalate to authorities False alarm: Record and update user status		

3.5.4 Analytics & Reporting Requirements

Table 33: Analytics Requirements

Req ID	**FR-ADMIN-04**	**Requirement Type**	**Enhanced**
Feature Name	System Analytics and Reporting	Priority	Medium
Description	Admin shall view system analytics and generate reports		
Preconditions	Admin logged in with analytics permissions		
Inputs	Date range, Report type, Filters		
Processing	1. Calculate key metrics: Rides, Revenue, Users, Ratings 2. Generate visualizations: Charts, Graphs, Trends 3. Create reports: Daily, Weekly, Monthly, Custom 4. Export reports: PDF, Excel, CSV 5. Monitor system health: Uptime, Errors, Performance		
Outputs	Analytics dashboard, Reports, Export files		
Postconditions	Data available for business decisions		
Error Handling	Data unavailable: Show cached or estimated data Export failed: Provide alternative format		

3.5.5 System Configuration Requirements

Table 34: System Configuration Requirements

Req ID	**FR-ADMIN-05**	**Requirement Type**	**Mandatory**
Feature Name	System Configuration Management	Priority	High
Description	Admin shall configure system parameters		
Preconditions	Admin logged in with configuration permissions		
Inputs	Configuration values, Service areas, Pricing, Rules		
Processing	1. Update pricing: Base fares, Distance rates, Time rates 2. Manage service areas: Add/remove regions 3. Configure rules: Cancellation fees, Commission rates 4. Set system parameters: Timeouts, Limits, Thresholds 5. Validate changes before applying		
Outputs	Configuration updated, Changes applied		
Postconditions	New settings active across system		
Error Handling	Invalid values: "Configuration validation failed" Update conflict: "Another admin is modifying settings"		

3.6 Rating & Feedback Module

3.6.1 Rating System Requirements

Table 35: Rating System Requirements

Req ID	**FR-RATE-01**	**Requirement Type**	**Mandatory**
Feature Name	Two-Way Rating System	Priority	High
Description	Riders and drivers shall rate each other after rides		
Preconditions	Ride completed successfully		
Inputs	Star rating (1-5), Optional comments, Tags		
Processing	**Rider rates Driver:** 1. Prompt rider within 24 hours of ride completion 2. Collect rating and feedback 3. Update driver's average rating 4. Store feedback for driver improvement **Driver rates Rider:** 1. Optional rating of rider behavior 2. Flag problematic riders for admin review		
Outputs	Rating submitted, Average updated		
Postconditions	Ratings visible in user profiles		
Error Handling	Already rated: "You have already rated this ride" Rating window expired: "Rating period has ended"		

3.6.2 Feedback Analysis Requirements

Table 36: Feedback Analysis Requirements

Req ID	**FR-RATE-02**	**Requirement Type**	**Enhanced**
Feature Name	Feedback Analysis and Reporting	Priority	Medium
Description	System shall analyze ratings and feedback		
Preconditions	Ratings and feedback collected		
Inputs	Rating data, Comments, Tags		
Processing	1. Calculate average ratings per user 2. Identify trends: Common complaints, Praises 3. Flag users with consistently low ratings 4. Generate improvement suggestions 5. Create rating reports for admin review		
Outputs	Rating analytics, User performance reports		
Postconditions	Data available for quality improvement		
Error Handling	Insufficient data: "More ratings needed for analysis"		

3.7 Payment & Earnings Module

3.7.1 Cash Payment Requirements

Table 37: Cash Payment Requirements

Req ID	**FR-PAY-01**	**Requirement Type**	**Critical**
Feature Name	Cash Payment Processing	Priority	Critical
Description	System shall handle cash payment confirmation		
Preconditions	Ride completed, Fare calculated		
Inputs	Driver confirmation of cash received		
Processing	1. Display final fare to rider and driver 2. Driver confirms cash receipt via app 3. System records payment as completed 4. Calculate commission (15% of fare) 5. Update driver's earnings record 6. Generate receipt for rider		
Outputs	Payment confirmed, Earnings updated, Receipt generated		
Postconditions	Ride fully completed, Payment recorded		
Error Handling	Payment dispute: Initiate dispute resolution process Confirmation missing: Reminder after 1 hour		

3.7.2 Earnings Management Requirements

Table 38: Earnings Management Requirements

Req ID	**FR-PAY-02**	**Requirement Type**	**Mandatory**
Feature Name	Driver Earnings Management	Priority	High
Description	Drivers shall track and manage their earnings		
Preconditions	Driver has completed rides with payments		
Inputs	Date range, Earnings filters		
Processing	1. Calculate total earnings: Gross - Commission 2. Show breakdown: By day, week, month, ride 3. Display statistics: Average per ride, Best days, Trends 4. Provide earnings summary for tax purposes 5. Enable earnings withdrawal process (external)		
Outputs	Earnings dashboard, Detailed reports, Summary		
Postconditions	Driver informed about earnings performance		
Error Handling	No earnings: "Complete rides to see earnings" Calculation error: "Earnings temporarily unavailable"		

4 Specific Requirements - Non-Functional

4.1 Performance Requirements

4.1.1 Response Time Requirements

Table 39: Response Time Requirements

Requirement ID	Description	Target	Max
NFR-PERF-01	App launch to main screen	2 seconds	3 seconds
NFR-PERF-02	Location search autocomplete	500ms	1 second
NFR-PERF-03	Fare calculation	1 second	2 seconds
NFR-PERF-04	Driver matching	3 seconds	5 seconds
NFR-PERF-05	Map loading and rendering	2 seconds	3 seconds
NFR-PERF-06	Real-time location update	3 seconds	5 seconds
NFR-PERF-07	API response time (95th percentile)	1 second	2 seconds
NFR-PERF-08	Database query response	500ms	1 second
NFR-PERF-09	Push notification delivery	2 seconds	5 seconds
NFR-PERF-10	SOS alert processing	1 second	2 seconds

4.1.2 Throughput & Capacity Requirements

Table 40: Throughput and Capacity Requirements

Requirement ID	Description	Requirement
NFR-PERF-11	Concurrent active rides	500 minimum
NFR-PERF-12	Simultaneous users	10,000 minimum
NFR-PERF-13	Ride requests per minute (peak)	100
NFR-PERF-14	API requests per second	200
NFR-PERF-15	Database transactions per second	500
NFR-PERF-16	WebSocket connections	5,000 concurrent
NFR-PERF-17	SMS messages per minute	50
NFR-PERF-18	Push notifications per minute	200
NFR-PERF-19	Image uploads per minute	20
NFR-PERF-20	Geolocation queries per second	100

4.1.3 Resource Utilization Requirements

Table 41: Resource Utilization Requirements

Requirement ID	Description	Limit
NFR-PERF-21	Mobile app memory usage	150MB
NFR-PERF-22	Mobile app storage usage	100MB
NFR-PERF-23	Mobile app battery consumption	5% per hour
NFR-PERF-24	Mobile app data usage	10MB per ride
NFR-PERF-25	Backend server CPU usage	70% average
NFR-PERF-26	Backend server memory usage	80%
NFR-PERF-27	Database storage growth	1GB per month
NFR-PERF-28	Network bandwidth	10Mbps average

4.2 Reliability Requirements

4.2.1 Availability Requirements

Table 42: Availability Requirements

Requirement ID	Description	Requirement
NFR-REL-01	System uptime (24/7)	99.5%
NFR-REL-02	System uptime (operational hours 6AM-11PM)	99.9%
NFR-REL-03	API availability	99.9%
NFR-REL-04	Database availability	99.95%
NFR-REL-05	External service dependency availability	99%
NFR-REL-06	Maximum continuous downtime	15 minutes
NFR-REL-07	Maximum monthly downtime	4 hours
NFR-REL-08	Maintenance window availability	95%

4.2.2 Fault Tolerance Requirements

Table 43: Fault Tolerance Requirements

Requirement ID	Description	Requirement
NFR-REL-09	Automatic failover time	2 minutes
NFR-REL-10	Data loss after failure	Zero
NFR-REL-11	Graceful degradation	Essential features remain
NFR-REL-12	Service recovery time	5 minutes
NFR-REL-13	Database recovery point objective	5 minutes
NFR-REL-14	Database recovery time objective	15 minutes
NFR-REL-15	Disaster recovery time	4 hours
NFR-REL-16	Backup completion time	1 hour

4.2.3 Data Integrity Requirements

Table 44: Data Integrity Requirements

Requirement ID	Description	Requirement
NFR-REL-17	Ride data accuracy	100%
NFR-REL-18	Payment data accuracy	100%
NFR-REL-19	User data consistency	100%
NFR-REL-20	Transaction atomicity	Guaranteed
NFR-REL-21	Data synchronization	Real-time
NFR-REL-22	Conflict resolution	Automatic
NFR-REL-23	Data validation	Pre and post
NFR-REL-24	Audit trail completeness	100%

4.3 Security Requirements

4.3.1 Authentication & Authorization

Table 45: Authentication and Authorization Requirements

Requirement ID	Description
NFR-SEC-01	All communications shall use TLS 1.3 encryption
NFR-SEC-02	User authentication via OTP with JWT tokens
NFR-SEC-03	JWT token expiration: 30 days access, 90 days refresh
NFR-SEC-04	Admin authentication with 2-factor option
NFR-SEC-05	Role-based access control (RBAC) with minimum privileges
NFR-SEC-06	Session timeout: 30 minutes inactivity
NFR-SEC-07	Password policy: Minimum 8 chars, complexity requirements
NFR-SEC-08	Account lockout after 5 failed attempts
NFR-SEC-09	Secure password hashing using bcrypt (cost factor 12)
NFR-SEC-10	API key rotation every 90 days

4.3.2 Data Protection Requirements

Table 46: Data Protection Requirements

Requirement ID	Description
NFR-SEC-11	Sensitive data encryption at rest (AES-256)
NFR-SEC-12	Personal data masking in logs and displays
NFR-SEC-13	Secure storage of driver documents
NFR-SEC-14	GDPR-like compliance for Pakistani data protection
NFR-SEC-15	Data retention: Ride data 90 days, User data 5 years
NFR-SEC-16	Right to data deletion upon account closure
NFR-SEC-17	Data backup encryption
NFR-SEC-18	Secure key management using HSM/equivalent
NFR-SEC-19	Regular security audits and vulnerability scans
NFR-SEC-20	Penetration testing quarterly

4.3.3 Application Security Requirements

Table 47: Application Security Requirements

Requirement ID	Description
NFR-SEC-21	Input validation and sanitization for all inputs
NFR-SEC-22	SQL injection prevention using parameterized queries
NFR-SEC-23	Cross-site scripting (XSS) protection
NFR-SEC-24	Cross-site request forgery (CSRF) protection
NFR-SEC-25	Rate limiting: 100 requests/minute per user
NFR-SEC-26	API endpoint authorization checks
NFR-SEC-27	Secure error handling (no sensitive data in errors)
NFR-SEC-28	Mobile app root/jailbreak detection
NFR-SEC-29	Certificate pinning for API calls
NFR-SEC-30	Code obfuscation for mobile apps

4.3.4 Emergency System Security

Table 48: Emergency System Security Requirements

Requirement ID	Description
NFR-SEC-31	SOS system shall work even if app is force-closed
NFR-SEC-32	Emergency data transmission priority over other traffic
NFR-SEC-33	Emergency location tracking even with GPS off
NFR-SEC-34	Secure emergency contact verification
NFR-SEC-35	Emergency alert non-repudiation
NFR-SEC-36	Emergency system audit logging
NFR-SEC-37	Emergency data retention: 7 years minimum
NFR-SEC-38	Law enforcement access protocol
NFR-SEC-39	Emergency system redundancy
NFR-SEC-40	False alarm prevention mechanisms

4.4 Safety Requirements

4.4.1 Emergency System Safety

Table 49: Emergency System Safety Requirements

Requirement ID	Description
NFR-SAFE-01	SOS button accessible from lock screen (configurable)
NFR-SAFE-02	Emergency alerts shall have 99.9% delivery reliability
NFR-SAFE-03	Emergency location accuracy within 10 meters
NFR-SAFE-04	Emergency response time: Admin notification within 10 seconds
NFR-SAFE-05	Continuous location tracking during emergency
NFR-SAFE-06	Emergency contact notification within 30 seconds
NFR-SAFE-07	Emergency system shall function in low-network conditions
NFR-SAFE-08	Emergency false alarm prevention (3-second hold)
NFR-SAFE-09	Emergency data backup every 10 seconds during incident
NFR-SAFE-10	Emergency system testing monthly

4.4.2 Driver Safety Requirements

Table 50: Driver Safety Requirements

Requirement ID	Description
NFR-SAFE-11	100% driver document verification before first ride
NFR-SAFE-12	Driver re-verification annually
NFR-SAFE-13	Driver rating threshold: Minimum 3.0 to remain active
NFR-SAFE-14	Driver working hours limit: 12 hours per day
NFR-SAFE-15	Driver rest period: 8 hours between shifts
NFR-SAFE-16	Driver location sharing with admin during rides
NFR-SAFE-17	Driver emergency training materials
NFR-SAFE-18	Driver incident reporting system
NFR-SAFE-19	Driver insurance verification
NFR-SAFE-20	Driver vehicle safety checks

4.4.3 Rider Safety Requirements

Table 51: Rider Safety Requirements

Requirement ID	Description
NFR-SAFE-21	Rider emergency contact requirement
NFR-SAFE-22	Ride sharing with contacts feature
NFR-SAFE-23	Rider rating visibility before ride acceptance
NFR-SAFE-24	Anonymous rider-driver communication
NFR-SAFE-25	Ride route deviation alerts
NFR-SAFE-26	Estimated time of arrival accuracy ±2 minutes
NFR-SAFE-27	Rider cancellation safety (no penalty in unsafe situations)
NFR-SAFE-28	Rider incident reporting
NFR-SAFE-29	Rider safety guidelines and tips
NFR-SAFE-30	Night ride safety enhancements

4.4.4 Vehicle Safety Requirements

Table 52: Vehicle Safety Requirements

Requirement ID	Description
NFR-SAFE-31	Vehicle registration verification
NFR-SAFE-32	Vehicle insurance verification
NFR-SAFE-33	Vehicle fitness certification
NFR-SAFE-34	Vehicle capacity limits enforcement
NFR-SAFE-35	Vehicle tracking during rides
NFR-SAFE-36	Vehicle speed monitoring
NFR-SAFE-37	Vehicle maintenance reminders
NFR-SAFE-38	Vehicle safety equipment verification
NFR-SAFE-39	Vehicle inspection records
NFR-SAFE-40	Vehicle replacement policy

4.5 Usability Requirements

4.5.1 User Interface Requirements

Table 53: User Interface Requirements

Requirement ID	Description
NFR-USE-01	First-time user booking within 3 minutes
NFR-USE-02	Mobile app learnability: 90% feature discovery in 1 week
NFR-USE-03	Error recovery: 95% success rate
NFR-USE-04	Task completion: 95% success rate for primary tasks
NFR-USE-05	User satisfaction: 4.0+ average app store rating
NFR-USE-06	Accessibility: WCAG 2.1 AA compliance
NFR-USE-07	Multilingual support: Urdu and English
NFR-USE-08	Touch target size: Minimum 44×44 pixels
NFR-USE-09	Font size: Minimum 16 points for critical information
NFR-USE-10	Color contrast: 4.5:1 minimum ratio

4.5.2 Driver Interface Safety

Table 54: Driver Interface Safety Requirements

Requirement ID	Description
NFR-USE-11	Hands-free interaction: Voice commands for critical actions
NFR-USE-12	Minimal interaction while driving
NFR-USE-13	Large touch targets: 48×48 dp minimum
NFR-USE-14	High contrast mode for daylight visibility
NFR-USE-15	Voice-guided navigation
NFR-USE-16	Automatic screen brightness adjustment
NFR-USE-17	Driving mode detection
NFR-USE-18	Distraction minimization
NFR-USE-19	Emergency access without unlocking
NFR-USE-20	Offline capability for critical functions

4.5.3 Error Handling & Help

Table 55: Error Handling and Help Requirements

Requirement ID	Description
NFR-USE-21	Error messages: Clear, actionable, non-technical
NFR-USE-22	Help availability: Contextual help within 2 taps
NFR-USE-23	Tutorials: Interactive for first-time users
NFR-USE-24	Support access: Within app, maximum 3 steps
NFR-USE-25	FAQ coverage: 95% of common questions
NFR-USE-26	Error recovery suggestions
NFR-USE-27	Confirmation for destructive actions
NFR-USE-28	Undo functionality where applicable
NFR-USE-29	Progress indication for long operations
NFR-USE-30	Feedback collection after critical actions

4.6 Maintainability Requirements

4.6.1 Code Quality Requirements

Table 56: Code Quality Requirements

Requirement ID	Description
NFR-MAIN-01	Code modularity: Separation of concerns
NFR-MAIN-02	Test coverage: Minimum 80% unit test coverage
NFR-MAIN-03	Code documentation: 30% comment density
NFR-MAIN-04	Coding standards: ESLint, PEP8 compliance
NFR-MAIN-05	Code review: 100% of changes reviewed
NFR-MAIN-06	Dependency management: Regular updates
NFR-MAIN-07	Configuration externalization
NFR-MAIN-08	Logging: Structured, searchable logs
NFR-MAIN-09	Monitoring: Comprehensive system metrics
NFR-MAIN-10	Alerting: Proactive issue detection

4.6.2 Deployment & Operations

Table 57: Deployment and Operations Requirements

Requirement ID	Description
NFR-MAIN-11	Zero-downtime deployments
NFR-MAIN-12	Rollback capability: 30-minute rollback
NFR-MAIN-13	Feature flags for gradual rollouts
NFR-MAIN-14	A/B testing framework
NFR-MAIN-15	Configuration hot reload
NFR-MAIN-16	Database migration automation
NFR-MAIN-17	Backup automation and testing
NFR-MAIN-18	Disaster recovery procedures
NFR-MAIN-19	Performance monitoring
NFR-MAIN-20	Capacity planning and scaling

4.7 Portability Requirements

4.7.1 Platform Compatibility

Table 58: Platform Compatibility Requirements

Requirement ID	Description
NFR-PORT-01	Android compatibility: 8.0+ (API 26+)
NFR-PORT-02	iOS compatibility: 14.0+
NFR-PORT-03	Web browser compatibility: Chrome 90+, Firefox 88+, Safari 14+
NFR-PORT-04	Screen size support: 4.7” to 7” phones
NFR-PORT-05	Orientation support: Portrait and landscape
NFR-PORT-06	Density support: MDPI to XXXHDPI
NFR-PORT-07	Network compatibility: 3G minimum, 4G/LTE recommended
NFR-PORT-08	Language support: Urdu (RTL), English (LTR)
NFR-PORT-09	Localization: Date, time, currency formats
NFR-PORT-10	Accessibility: Screen reader support

4.8 Scalability Requirements

4.8.1 System Scalability

Table 59: Scalability Requirements

Requirement ID	Description
NFR-SCAL-01	User growth: Support 10× growth without redesign
NFR-SCAL-02	Geographic expansion: Multi-city support
NFR-SCAL-03	Horizontal scaling: Stateless services
NFR-SCAL-04	Database scaling: Read replicas, sharding
NFR-SCAL-05	Cache scaling: Distributed caching
NFR-SCAL-06	Load balancing: Automatic traffic distribution
NFR-SCAL-07	Queue processing: Asynchronous job processing
NFR-SCAL-08	CDN integration: Static asset delivery
NFR-SCAL-09	Monitoring at scale
NFR-SCAL-10	Cost optimization at scale

4.9 Legal & Compliance Requirements

4.9.1 Regulatory Compliance

Table 60: Regulatory Compliance Requirements

Requirement ID	Description
NFR-LEGAL-01	Pakistan transport regulations compliance
NFR-LEGAL-02	Data protection laws compliance
NFR-LEGAL-03	Tax reporting requirements
NFR-LEGAL-04	Age verification: 16+ riders, 21+ drivers
NFR-LEGAL-05	Terms of Service and Privacy Policy
NFR-LEGAL-06	Record keeping: 5 years minimum
NFR-LEGAL-07	Law enforcement cooperation protocol
NFR-LEGAL-08	Insurance requirements verification
NFR-LEGAL-09	Accessibility compliance
NFR-LEGAL-10	Environmental regulations

5 External Interface Requirements

5.1 User Interfaces

5.1.1 Rider Mobile Application Interfaces

Table 61: Rider App Screen Specifications

Screen	Components	Priority
Splash Screen	Logo, Loading indicator	Critical
Onboarding	Feature introduction, Permissions request	High
Login/Register	Phone input, OTP verification	Critical
Home/Map	Map view, Current location, Search bar	Critical
Location Selection	Pickup/Dropoff selection, Address search	Critical
Vehicle Selection	Vehicle types, Fare estimates, Selection	Critical
Ride Request	Driver search, Waiting screen	Critical
Ride Tracking	Live map, Driver info, ETA, SOS button	Critical
Shared Rides Browser	Available rides, Savings, Join options	High
Payment Screen	Fare amount, Cash payment confirmation	Critical
Rating Screen	Star rating, Comments, Tags	High
Ride History	Past rides, Receipts, Filters	Medium
Profile Management	Personal info, Emergency contacts, Photo	High
Settings	Preferences, Notifications, Language	Medium
Emergency Contacts	Contact list, Add/remove, Verification	High
Help & Support	FAQ, Contact support, Tutorials	Medium

5.1.2 Driver Mobile Application Interfaces

Table 62: Driver App Screen Specifications

Screen	Components	Priority
Registration	Document upload, Verification status	Critical
Home Screen	Online/Offline toggle, Earnings summary	Critical
Ride Request	Request modal, Pickup details, Accept/Decline	Critical
Navigation	Turn-by-turn navigation, Route overview	Critical
Ride Management	Passenger list, Start/End ride, Multiple pickups	Critical
Payment Collection	Fare amount, Cash received confirmation	Critical
Earnings Dashboard	Daily/weekly earnings, Charts, Withdrawal	High
Vehicle Management	Vehicle details, Documents, Maintenance	High
Rating Management	Rider ratings, Feedback, Performance	Medium
Profile Settings	Personal info, Availability schedule	Medium
Support	Help, Issue reporting, Contact	Medium

5.1.3 Administrator Web Panel Interfaces

Table 63: Admin Panel Interface Specifications

Section	Components	Priority
Dashboard	System metrics, Alerts, Quick actions	Critical
User Management	User search, Details view, Actions	Critical
Driver Verification	Document review, Approval/Rejection	Critical
Ride Monitoring	Active rides map, Ride details, Intervention	Critical
Emergency Alerts	Real-time alerts, Response tools, History	Critical
Analytics	Reports, Charts, Export, Trends	High
Configuration	System settings, Pricing, Rules, Areas	High
Support Tickets	Ticket management, Resolution tracking	Medium
Audit Logs	System activity, User actions, Security events	Medium
System Health	Performance metrics, Error logs, Uptime	High

5.2 Hardware Interfaces

5.2.1 Mobile Device Requirements

Table 64: Mobile Device Hardware Requirements

Component	Requirement	Criticality
GPS Receiver	Accurate to 10 meters, Supports background updates	Critical
Camera	Minimum 5MP, Auto-focus, Flash support	High
Network	4G/LTE recommended, 3G minimum	Critical
Processor	Quad-core minimum, 1.5GHz+	High
Memory	2GB RAM minimum	High
Storage	100MB free space minimum	Medium
Battery	2500mAh minimum, Power management	Medium
Sensors	Accelerometer, Gyroscope, Compass	Medium
Audio	Microphone, Speaker, Headphone jack	Medium
Display	4.7” minimum, Touch screen	Critical

5.3 Software Interfaces

5.3.1 External API Dependencies

Table 65: External API Interface Requirements

API Service	Interface Requirements	Criticality
Google Maps Platform	Maps SDK, Places API, Directions API, Geocoding API	Critical
SMS Gateway (Twilio/MSG91)	SMS sending API, Delivery reports, Error handling	Critical
Firebase Cloud Messaging	Push notification API, Device registration, Topics	Critical
Cloud Storage (AWS S3)	File upload/download API, Security, CDN	High
Payment Gateway (Future)	REST API for payments, Webhooks, Security	Low
Analytics (Google Analytics)	Event tracking API, User analytics, Custom events	Medium
Error Tracking (Sentry)	Error reporting API, Stack traces, Environment data	Medium

5.3.2 Internal API Specifications

Table 66: Internal API Requirements

API Category	Specifications	Protocol
Authentication API	JWT-based, OTP verification, Session management	REST/HTTPS
Ride Management API	Ride creation, matching, tracking, completion	REST/HTTPS
Real-time API	WebSocket connections, Live location updates	WebSocket
Payment API	Fare calculation, Payment recording, Earnings	REST/HTTPS
User Management API	Profile management, Ratings, Preferences	REST/HTTPS
Admin API	User management, Verification, Analytics	REST/HTTPS
Emergency API	SOS alerts, Emergency handling, Notifications	REST/HTTPS

5.4 Communication Interfaces

5.4.1 Network Communication Requirements

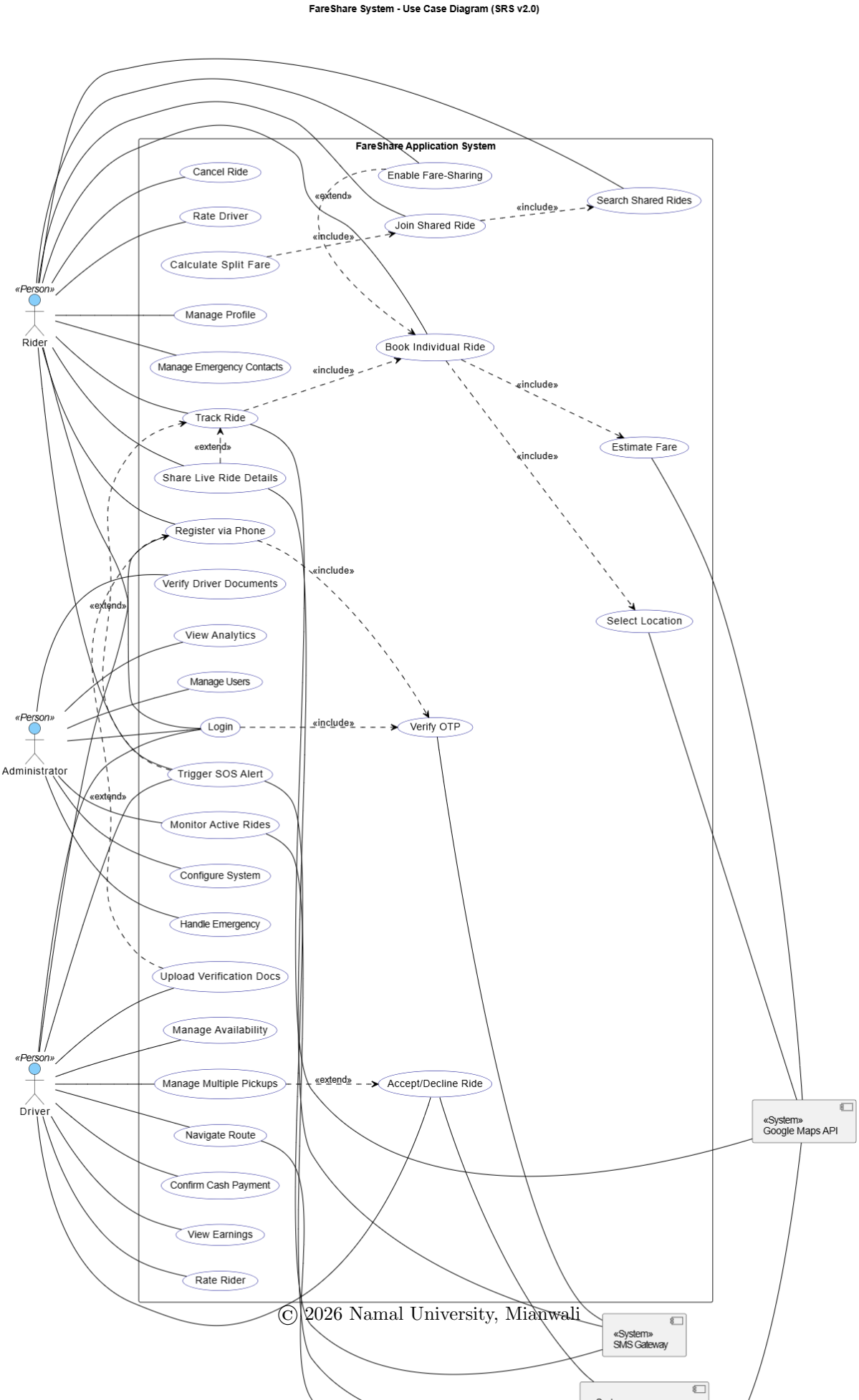
Table 67: Communication Interface Requirements

Interface	Requirements
HTTP/HTTPS	TLS 1.3 encryption, HTTP/2 support, Compression
WebSocket	Secure WebSocket (wss://), Heartbeat mechanism, Reconnection
SMS	Reliable delivery, Delivery reports, Error handling
Push Notifications	Platform-specific (APNS for iOS, FCM for Android)
Email	Transactional emails, Templates, Delivery tracking
File Transfer	Secure upload/download, Progress tracking, Resume capability
Real-time Updates	Low latency (¡5s), Efficient data format (JSON/Protocol Buffers)

6 System Models and Analysis

6.1 Use Case Model

6.1.1 Use Case Diagram



6.1.2 Primary Use Cases

Table 68: Primary Use Case Specifications

Use Case ID	**UC-01: Book Individual Ride**
Primary Actor	Rider
Description	Rider books a non-shared ride from pickup to destination
Preconditions	Rider logged in, Location services enabled, Payment method set
Main Flow	1. Rider opens app and views map 2. Rider selects pickup location (auto or manual) 3. Rider selects destination 4. System shows available vehicle types with fares 5. Rider selects vehicle type 6. System finds nearby drivers 7. Driver accepts ride 8. System notifies rider with driver details 9. Ride proceeds to completion
Alternative Flows	A1: No drivers available - Suggest retry or wait A2: Rider cancels before driver acceptance A3: Driver cancels after acceptance
Postconditions	Ride completed or cancelled with appropriate handling

Table 69: SOS Emergency Use Case

Use Case ID	**UC-02: Trigger SOS Emergency Alert**
Primary Actor	Rider or Driver
Description	User triggers emergency alert during active ride
Preconditions	Active ride in progress, Network connectivity
Main Flow	1. User presses and holds SOS button (3 seconds) 2. System captures current GPS location 3. System retrieves ride and user details 4. System sends SMS to emergency contacts 5. System notifies FareShare emergency team 6. System starts continuous location tracking 7. Emergency team responds and handles situation 8. Emergency marked as resolved
Alternative Flows	A1: Network unavailable - Store alert and send when available A2: False alarm - User cancels within 10 seconds
Postconditions	Emergency handled, Incident recorded, Follow-up actions

Table 70: Admin Use Case: Driver Verification

Use Case ID	UC-03: Verify Driver Documents
Primary Actor	Administrator
Description	Admin verifies driver submitted documents manually
Preconditions	Driver submitted documents, Admin logged in
Main Flow	1. Admin views driver verification queue 2. Admin selects driver for verification 3. System displays driver documents (CNIC, License, Registration) 4. Admin examines document authenticity and clarity 5. Admin makes decision: Approve, Reject, or Request more 6. System updates driver status based on decision 7. System notifies driver of verification result 8. Admin records verification details and comments
Alternative Flows	A1: Documents unclear - Request better quality images A2: Verification conflict - Escalate to senior admin
Postconditions	Driver verified or rejected, Appropriate notifications sent

6.2 Data Model

6.2.1 Context Diagram

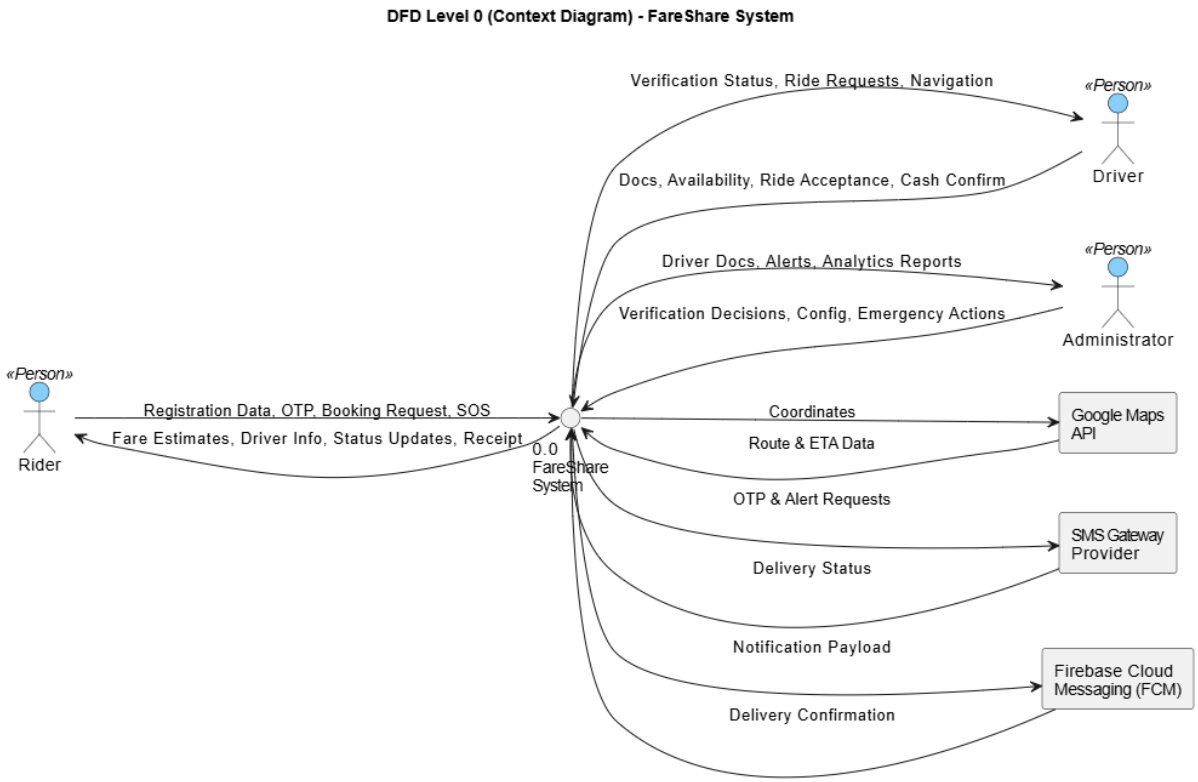


Figure 2: FareShare Context Diagram

6.2.2 Core Database Tables

Table 71: Core Database Schema

Table Name	**Users**
Primary Key	user_id (UUID)
Fields	phone_number (String, Unique), role (Enum: rider/driver/admin), full_name (String), email (String, Optional), profile_photo_url (String), verification_status (Enum), rating_average (Float), total_ratings (Integer), created_at (Timestamp), last_login (Timestamp), status (Enum)
Relationships	One-to-Many: Rides (as rider), Ratings (given/received), One-to-One: DriverDetails (if role=driver), EmergencyContacts

Table 72: Rides Table Schema

Table Name	**Rides**
Primary Key	ride_id (UUID)
Fields	rider_ids (Array of UUIDs), driver_id (UUID, Foreign Key), status (Enum), is_shared (Boolean), vehicle_type (Enum), pickup_locations (Array of GeoPoints), dropoff_locations (Array), estimated_fare (Float), final_fare (Float), individual_fares (Map), distance (Float), duration (Integer), route_polyline (String), created_at (Timestamp), accepted_at (Timestamp), started_at (Timestamp), completed_at (Timestamp), cancelled_at (Timestamp), cancellation_reason (String)
Indexes	Geospatial index on pickup_locations, driver_id + status, created_at (for time-based queries), rider_ids (array contains)

7 Appendices

7.1 Appendix A: References

1. **IEEE Std 830-1984** - IEEE Guide to Software Requirements Specifications
2. **IEEE Std 29148-2018** - Systems and software engineering - Life cycle processes - Requirements engineering
3. **FareShare Project Proposal** - November 9, 2025
4. **Meeting Minutes** - November 3, 2025 to January 10, 2026
5. **Project Milestone 2 Description** - CSC-225 Course Document
6. **Google Maps Platform Documentation**
7. **Firebase Documentation**
8. **Pakistan Transport Regulations** - Ministry of Communications

7.2 Appendix B: Complete Use Case List

7.2.1 Rider Use Cases (15)

(a) Register Account with Phone Number
(b) Verify OTP and Login
(c) Set/Update Profile Information
(d) Add/Manage Emergency Contacts
(e) Book Individual Ride
(f) Enable Fare-Sharing for Ride
(g) Search and Join Shared Rides
(h) Track Ride in Real-time
(i) Cancel Ride (with reasons)
(j) Rate Driver after Ride
(k) View Ride History and Receipts
(l) Share Ride Details with Contacts
(m) Trigger SOS Emergency Alert
(n) Report Issue or Incident
(o) Manage Payment and Preferences

7.2.2 Driver Use Cases (12)

(a) Register with Document Upload
(b) Complete Profile and Vehicle Information
(c) Go Online/Offline (Availability)
(d) Accept/Decline Ride Requests
(e) Navigate to Pickup Location
(f) Start Ride (Pickup Confirmation)
(g) Manage Multiple Pickups (Shared Rides)
(h) Navigate to Destination(s)
(i) End Ride and Calculate Fare
(j) Confirm Cash Payment Received
(k) Rate Rider after Ride
(l) View Earnings and Performance

7.2.3 Administrator Use Cases (10)

(a) Login with Admin Credentials
(b) Verify Driver Documents
(c) Manage User Accounts (Suspend/Activate)
(d) Monitor Active Rides
(e) Respond to Emergency Alerts
(f) Handle Disputes and Reports
(g) Generate Analytics Reports
(h) Configure System Settings
(i) Manage Service Areas and Pricing
(j) Audit System Activity and Logs

7.3 Appendix C: Data Dictionary

7.3.1 User-Related Data Elements

Table 73: User Data Dictionary

Field	Type	Description
user_id	UUID v4	Unique user identifier
phone_number	String (+92XXXXXXXXXX)	Pakistani mobile number
role	Enum	'rider', 'driver', 'admin'
verification_status	Enum	'unverified', 'pending', 'verified', 'rejected'
rating_average	Decimal(3,2)	Average rating 0.00-5.00
total_rides	Integer	Completed ride count
emergency_contacts	Array	List of emergency contact objects
preferences	JSON	User preferences and settings

7.3.2 Ride-Related Data Elements

Table 74: Ride Data Dictionary

Field	Type	Description
ride_id	UUID v4	Unique ride identifier
status	Enum	'created', 'searching', 'accepted', 'in_progress', 'completed', 'cancelled'
is_shared	Boolean	True if fare-sharing enabled
vehicle_type	Enum	'bike', 'rickshaw', 'mini', 'sedan'
pickup_locations	GeoJSON Array	Array of pickup coordinates
dropoff_locations	GeoJSON Array	Array of dropoff coordinates
estimated_fare	Decimal(10,2)	PKR estimated fare
final_fare	Decimal(10,2)	PKR actual fare
distance_km	Decimal(6,2)	Total distance in kilometers
duration_min	Integer	Total duration in minutes
route_polyline	String	Encoded route polyline

7.3.3 Safety-Related Data Elements

Table 75: Safety Data Dictionary

Field	Type	Description
emergency_alert_id	UUID v4	Unique emergency identifier
alert_type	Enum	'sos', 'route_deviation', 'panic'
alert_status	Enum	'active', 'responding', 'resolved', 'false_alarm'
location_history	GeoJSON Array	Array of location points during emergency
response_actions	JSON Array	List of response actions taken
resolution_notes	Text	Details of how emergency was resolved
contact_notifications	JSON Array	Record of contacts notified
admin_response_time	Integer	Seconds to first admin response

7.4 Appendix D: Pricing Algorithm Details

7.4.1 Fare Calculation Formula

The fare calculation uses the following formula:

$$\text{Fare} = \text{Base Fare} + (\text{Distance} \times \text{Distance Rate}) + (\text{Time} \times \text{Time Rate})$$

7.4.2 Vehicle Type Pricing

Table 76: Vehicle Pricing Matrix (PKR)

Vehicle Type	Base Fare	Per Km Rate	Per Min Rate	
Bike	30	8	2	
Rickshaw	50	12	3	
Mini Car	80	18	4	
Sedan	120	25	5	

7.4.3 Fare-Sharing Calculation

For shared rides with n passengers:

$$\text{Individual Fare}_i = \frac{\text{Base Fare}}{n} + \left(\frac{\text{Distance}_i}{\text{Total Distance}} \times \text{Distance Fare}\right) + \left(\frac{\text{Time}_i}{\text{Total Time}} \times \text{Time Fare}\right)$$

Minimum Savings Guarantee: Each passenger saves at least 25% compared to individual ride.