

PROJECT REPORT

ON

ISE TASK-2

**Comprehensive System Design and Agile Implementation for a Ride-Hailing Application**

Submitted to

**NMAM Institute of Technology, Nitte**

(Off-Campus Centre, Nitte Deemed to be University, Nitte – 576110, Karnataka, India)

On partial fulfilment of the requirements for the award of the

Degree of Bachelor of Technology

In

INFORMATION SCIENCE AND ENGINEERING

By

**Ayush B Shetty**

**USN: NNM23IS029**

Under the guidance of

**Dr. JASON ELROY MARTIS , Associate Professor**

Department of Information Science and Technology,

NMAM Institute of Technology, Nitte, Karnataka,

India

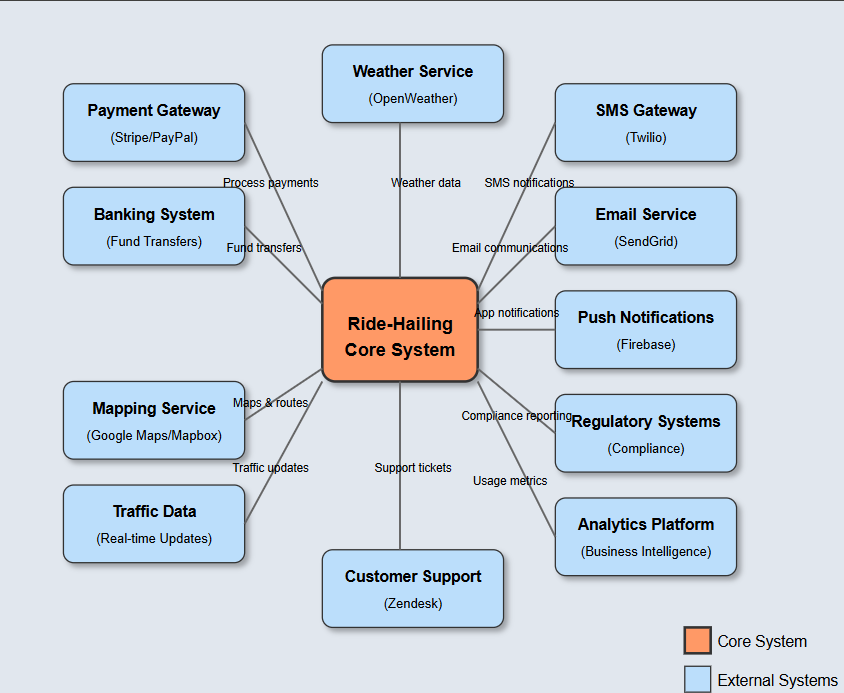
# **Agile Development Plan for Ride-Hailing App**

Using Scrum Methodology

## **Executive Summary**

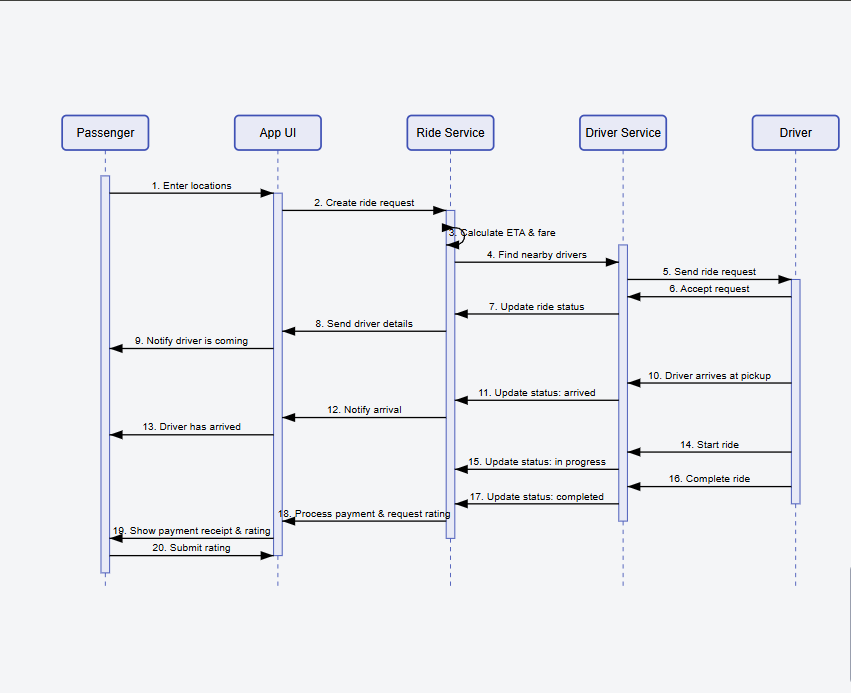
This report outlines a comprehensive 12-week Agile development plan for building a ride-hailing application using Scrum methodology. The plan includes sprint structures, prioritized user stories, and quality assurance standards to ensure iterative delivery of a high-quality product.

## **Sprint Timeline Overview**



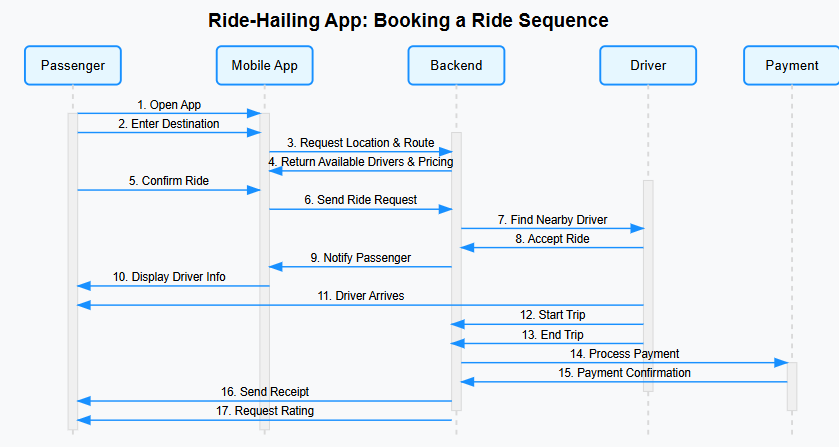
### **6 Sprints (12 Weeks Total)**

1. Sprint 1 - Foundation (2 weeks)
   1. Core authentication systems
   2. Location service integration
   3. Database architecture
2. Sprint 2 - Core Ride Functions (2 weeks)
   1. Ride request workflow
   2. Driver matching algorithm
   3. Basic navigation implementation
3. Sprint 3 - Payments & Ratings (2 weeks)
   1. Payment gateway integration
   2. Rating system architecture
   3. Dynamic fare calculation
4. Sprint 4 - Enhanced Features (2 weeks)
   1. Scheduled rides functionality
   2. In-app messaging system
   3. Ride history tracking
5. Sprint 5 - Optimization (2 weeks)
   1. Performance enhancements
   2. Surge pricing model
   3. Analytics dashboard
6. Sprint 6 - Final Polishing (2 weeks)
   1. UI/UX refinements
   2. Comprehensive testing
   3. Launch preparation

**Key User Stories**

### **Passenger-Critical Features**

1. Ride Request System
   1. GPS-based pickup selection
   2. Destination input with fare estimation
   3. Vehicle type selection
   4. Confirmation workflow
2. Real-Time Tracking
   1. Live driver location updates (5-second refresh)
   2. Dynamic ETA calculations
   3. Arrival notifications
3. Payment Management
   1. Multiple payment method support
   2. PCI-compliant storage
   3. Automated receipt generation



### **Driver-Critical Features**

1. Ride Acceptance Flow
   1. Geofenced request notifications
   2. 15-second response window
   3. Automatic navigation initiation
2. Navigation System
   1. Turn-by-turn directions
   2. Traffic-aware routing
   3. Background operation support
3. Earnings Management
   1. Real-time earnings dashboard
   2. Historical data visualization
   3. Exportable financial records

## **Quality Assurance Standards**

### **Universal Definition of Done**

1. Code Requirements
   1. Peer-reviewed by at least one developer
   2. 100% passing automated test suite
   3. Adherence to style guidelines
2. Performance Benchmarks
   1. Sub-2-second response times
   2. Memory/battery optimization
   3. Support for 2x expected load
3. Security Protocols
   1. Vulnerability scanning completed
   2. Data encryption implemented
   3. Authentication safeguards
4. User Experience
   1. Accessibility compliance
   2. Responsive design validation
   3. Cross-device functionality

## **Implementation Methodology**

### **Scrum Process Flow**

1. Sprint Planning
   1. Backlog prioritization
   2. Story point estimation
   3. Deliverable commitment
2. Daily Execution
   1. 15-minute standups
   2. Scrum board updates
   3. Continuous integration
3. Review Cycles
   1. Biweekly demos
   2. Stakeholder feedback incorporation
   3. Backlog refinement
4. Continuous Improvement
   1. Retrospective meetings
   2. Process optimization
   3. Knowledge sharing

## **Technical Recommendations**

1. State Management
   1. Persistent state machine with audit trails
2. Real-Time Systems
   1. WebSocket implementation for live updates
3. Geospatial Processing
   1. Geohash-based driver matching
4. Payment Security
   1. Idempotent transaction processing
5. Observability
   1. Comprehensive transition logging

## **Conclusion**

This Agile plan provides a structured yet flexible framework for delivering a full-featured ride-hailing application. The sprint-based approach ensures regular delivery of functional components while maintaining adaptability to changing requirements. By adhering to the rigorous Definition of Done criteria, the development team can maintain high quality standards throughout the project lifecycle.

The prioritized user stories focus first on core ride functionality before expanding to enhanced features, ensuring early delivery of business value. Regular review cycles and continuous integration practices will maintain alignment with stakeholder needs while technical best practices address scalability and reliability concerns.

This methodology balances rapid iteration with production readiness, positioning the project for successful deployment within the 12-week timeline while establishing a foundation for future enhancements.