Software Requirements Specification

For Car On Roads Advertisement System (CORADS)

Version 1.0
Prepared by Amir
May 1, 2021

Table of Contents

1. Introduction

1.1 Purpose

This Software Requirements Specification (SRS) document provides a detailed description of the Car On Roads Advertisement System (CORADS). It outlines the functional and non-functional requirements for the development of this location-based advertising platform that connects advertisers with drivers through mobile applications. This document will serve as a guide for the development team and as a contract between the stakeholders and the development team.

1.2 Document Conventions

This document uses the following conventions:

- Requirements are organized hierarchically with unique identifiers.
- Functional requirements are labeled with "FR-" followed by a number (e.g., FR-1).
- Non-functional requirements are labeled with "NFR-" followed by a number (e.g., NFR-1).

- Priority levels for requirements: High, Medium, Low.

1.3 Intended Audience and Reading Suggestions

This document is intended for:

- Development Team: To understand what to build and how to build it.
- Project Managers: To plan and track project progress.
- Testers: To develop test cases and validation procedures.
- Business Stakeholders: To confirm their needs are correctly captured.

For developers, it is recommended to read the entire document with a focus on sections 3 and 4. For project managers, sections 1, 2, and 6 are most relevant. Testers should focus on sections 3, 4, and 5.

1.4 Product Scope

CORADS is a location-based advertising platform that connects advertisers with drivers through mobile applications to display advertisements on moving cars. The platform aims to provide a cost-effective advertising solution by leveraging cars as mobile advertisement displays.

The system will:

- Allow advertisers to create and manage advertising campaigns
- Enable drivers to sign up and accept advertisement assignments
- Provide real-time tracking of advertisement displays

- Offer performance analytics for advertising campaigns
- Facilitate payments between advertisers and drivers

Support communication between advertisers, drivers, and administrators

The goal is to provide small businesses with an affordable and effective advertising alternative to expensive digital platforms while giving car drivers an opportunity to earn additional income.

1.5 References

IEEE Standard 830-1998, IEEE Recommended Practice for Software Requirements Specifications.

1. CORADS Thesis Document, 2021.
2. Android Development Guidelines, Google.
3. AWS Service Documentation.
4. Firebase Documentation.

2. Overall Description

2.1 Product Perspective

CORADS is a new, self-contained product designed to bridge the gap between traditional static advertisements and expensive digital advertising platforms. It consists of three main components:

1. **Advertiser Mobile Application**: Used by businesses to create campaigns, track advertisements, and analyze performance.
2. **Driver Mobile Application**: Used by drivers to receive and accept advertisement assignments, track their routes, and manage payments.
3. **Admin Web Portal**: Used by administrators to manage users, campaigns, payments, and overall system functionality.

The system will be hosted on AWS cloud infrastructure and will utilize Firebase for real-time communications, Google Maps for location services, and JAZZ CASH for payment processing.

2.2 Product Functions

The major functions of CORADS include:

1. **Campaign Management**: Creation, tracking, and analysis of advertising campaigns.

2. **Driver Trail Management**: Verification and monitoring of drivers' participation.

3. **Payment Processing**: Handling transactions between advertisers, drivers, and the platform.

4. **User Management**: Registration, authentication, and profile management.

5. **Real-time Tracking**: Monitoring the movement and display of advertisements.

6. **Reporting and Analytics**: Generating insights about campaign performance.

7. **Communication**: Facilitating interaction between advertisers, drivers, and administrators.

2.3 User Classes and Characteristics

The system will serve three primary user classes:

1. **Advertisers**:

- Small to medium-sized businesses looking for cost-effective advertising
- May have limited technical expertise
- Need easy-to-use interfaces for creating and monitoring campaigns
- Require transparent reporting on advertisement performance

2. **Drivers**:

- Car owners looking to earn additional income
- Varying levels of technical proficiency
- Need simple interfaces for accepting jobs and tracking earnings
- May require training on advertisement installation and verification

3. **Administrators**:

- Technical staff managing the platform
- Responsible for system maintenance and user support
- Need comprehensive control interfaces
- Require access to detailed reports and system configurations

2.4 Operating Environment

The CORADS system will operate within the following environment:

1. **Mobile Applications**:

- Android Operating System (minimum version 6.0)
- Requires internet connectivity
- Requires Google Play Services

- Utilizes device GPS and camera

2. **Admin Portal**:

- Web-based interface compatible with modern browsers (Chrome, Firefox, Safari, Edge)
- Responsive design supporting desktop and tablet screens

3. **Server Infrastructure**:

- Hosted on AWS EC2 instances
- PHP backend for API services
- Firebase for real-time notifications
- MySQL database for data storage

2.5 Design and Implementation Constraints

The development of CORADS is subject to the following constraints:

1. **Device Compatibility**: The mobile applications must function on a wide range of Android devices with varying screen sizes and capabilities.

2. **Connectivity Requirements**: The system requires internet connectivity for most operations, though limited offline functionality may be available.

3. **Third-Party Dependencies**: The system relies on several third-party services (AWS, Firebase, Google Maps, JAZZ CASH) which may impose their own limitations.

4. **Regulatory Compliance**: The system must comply with local advertising regulations and data protection laws.

5. **Budget Constraints**: Development must be completed within the allocated budget, requiring careful prioritization of features.

2.6 User Documentation

The following user documentation will be provided:

1. **Installation Guides**: Step-by-step instructions for installing the mobile applications.
2. **User Manuals**: Comprehensive guides for advertisers, drivers, and administrators.
3. **Video Tutorials**: Short instructional videos demonstrating key functions.
4. **FAQ Section**: Answers to commonly asked questions about the system.
5. **Help Center**: In-app support resources and contact information.

2.7 Assumptions and Dependencies

The development of CORADS is based on the following assumptions and dependencies:

1. **Assumptions**:

- Users will have access to smartphones with GPS capabilities
- Cars will be suitable for advertisement installation
- Advertisers will provide appropriate advertisement content
- Drivers will follow designated routes when displaying advertisements

2. **Dependencies**:

- Availability of AWS cloud services
- Reliability of Firebase SDK
- Functionality of Google Maps API
- Integration capabilities with JAZZ CASH payment gateway
- Stability of third-party libraries and services

3. System Features

3.1 Advertiser Module

3.1.1 Registration and Authentication

- **FR-1**: Advertisers shall be able to register with the application.
- **FR-2**: Registration shall require verification through OTP (One-Time Password).
- **FR-3**: Advertisers shall be able to log in to the application using their credentials.
- **FR-4**: Advertisers shall be able to reset their password through a forgotten password feature.

3.1.2 Profile Management

- **FR-5**: Advertisers shall be able to view their profile information.
- **FR-6**: Advertisers shall be able to update their profile information.

3.1.3 Wallet Management

- **FR-7**: Advertisers shall be able to view their wallet balance.
- **FR-8**: Advertisers shall be able to view their transaction history.
- **FR-9**: Advertisers shall be able to recharge their wallet using JAZZ CASH.
- **FR-10**: Advertisers shall be able to recharge their wallet using vouchers.

3.1.4 Campaign Management

- **FR-11**: Advertisers shall be able to create advertising campaigns.

- **FR-12**: Advertisers shall be able to specify campaign parameters including:
  - Target location
  - Campaign duration
  - Number of cars
  - Advertisement content (images)
- **FR-13**: Advertisers shall be able to view active campaigns.
- **FR-14**: Advertisers shall be able to view pending campaigns.
- **FR-15**: Advertisers shall be able to view completed campaigns.
- **FR-16**: Advertisers shall be able to track the current location of drivers displaying their advertisements.

3.1.5 Support and Communication

- **FR-17**: Advertisers shall be able to contact customer support.
- **FR-18**: Advertisers shall be able to inquire about driver status.

3.2 Driver Module

3.2.1 Registration and Authentication

- **FR-19**: Drivers shall be able to register with the application.
- **FR-20**: Drivers shall be able to log in to the application using their credentials.
- **FR-21**: Drivers shall be able to sign out from the application.

3.2.2 Profile Management

- **FR-22**: Drivers shall be able to view their profile information.
- **FR-23**: Drivers shall be able to update their profile information.
- **FR-24**: Drivers shall be able to upload images of their vehicles.

3.2.3 Campaign Interaction

- **FR-25**: Drivers shall be able to receive campaign requests.
- **FR-26**: Drivers shall be able to accept or reject campaign requests.
- **FR-27**: Drivers shall be able to view navigation to installation centers.
- **FR-28**: Drivers shall be able to start campaigns after installation.
- **FR-29**: Drivers shall be able to record their campaign activities.
- **FR-30**: Drivers shall be able to upload verification images with location data.

3.2.4 Wallet Management

- **FR-31**: Drivers shall be able to view their wallet balance.
- **FR-32**: Drivers shall be able to view their earnings history.

3.2.5 Support and Communication

- **FR-33**: Drivers shall be able to access help center resources.
- **FR-34**: Drivers shall be able to contact customer support.

3.3 Admin Module

3.3.1 Authentication and Access Control

- **FR-35**: Administrators shall be able to log in to the admin portal.
- **FR-36**: Administrators shall be able to reset their password.
- **FR-37**: The system shall support different admin roles with varying access levels.

3.3.2 User Management

- **FR-38**: Administrators shall be able to add, delete, and update admin accounts.
- **FR-39**: Administrators shall be able to add drivers to the system.
- **FR-40**: Administrators shall be able to create driver profiles.
- **FR-41**: Administrators shall be able to update user information.
- **FR-42**: Administrators shall be able to block and unblock driver accounts.
- **FR-43**: Administrators shall be able to send notifications to drivers and advertisers.
- **FR-44**: Administrators shall be able to send emails to drivers and advertisers.

3.3.3 Campaign Management

- **FR-45**: Administrators shall be able to search for campaigns and users.
- **FR-46**: Administrators shall be able to activate campaigns.
- **FR-47**: Administrators shall be able to start driver campaigns from installation centers.
- **FR-48**: Administrators shall be able to complete driver trials.

3.3.4 System Configuration

- **FR-49**: Administrators shall be able to add, delete, and update coupons.
- **FR-50**: Administrators shall be able to add, delete, and update vouchers.
- **FR-51**: Administrators shall be able to add, delete, and update car types.
- **FR-52**: Administrators shall be able to manage installation centers.

3.4 Campaign Management

3.4.1 Campaign Creation

- **FR-53**: The system shall allow the creation of advertising campaigns with specified parameters.
- **FR-54**: The system shall support multiple images in campaign creation.
- **FR-55**: The system shall validate campaign parameters before creation.

3.4.2 Campaign Assignment

- **FR-56**: The system shall match campaigns to appropriate drivers based on location and availability.
- **FR-57**: The system shall send campaign requests to selected drivers.
- **FR-58**: The system shall track driver responses to campaign requests.

3.4.3 Campaign Execution

- **FR-59**: The system shall guide drivers to installation centers.
- **FR-60**: The system shall track the installation process.
- **FR-61**: The system shall monitor campaign progress in real-time.
- **FR-62**: The system shall enforce campaign time and distance constraints.

3.4.4 Campaign Reporting

- **FR-63**: The system shall generate campaign performance reports.
- **FR-64**: The system shall provide heat maps showing advertisement display concentration.
- **FR-65**: The system shall track the duration and location of advertisement displays.

3.5 Payment System

3.5.1 Payment Processing

- **FR-66**: The system shall integrate with JAZZ CASH payment gateway.
- **FR-67**: The system shall process wallet recharge transactions.
- **FR-68**: The system shall calculate driver earnings based on campaign participation.
- **FR-69**: The system shall maintain transaction records.

3.5.2 Discount Management

- **FR-70**: The system shall support discount coupons for advertisers.
- **FR-71**: The system shall support voucher-based wallet recharges.
- **FR-72**: The system shall validate coupons and vouchers before application.

3.6 Tracking System

3.6.1 Real-time Tracking

- **FR-73**: The system shall track the real-time location of drivers.
- **FR-74**: The system shall display active campaigns on a map.
- **FR-75**: The system shall allow advertisers to track specific drivers.

3.6.2 Route Monitoring

- **FR-76**: The system shall record driver routes during campaigns.
- **FR-77**: The system shall analyze driver movement patterns.
- **FR-78**: The system shall detect deviations from expected routes.

3.6.3 Verification System

- **FR-79**: The system shall request verification images from drivers.
- **FR-80**: The system shall verify image location data.
- **FR-81**: The system shall maintain a record of verification activities.

3.7 Support System

3.7.1 Customer Support

- **FR-82**: The system shall provide support channels for advertisers and drivers.
- **FR-83**: The system shall allow users to submit support tickets.
- **FR-84**: The system shall track support request status.

3.7.2 Communication

- **FR-85**: The system shall support FCM notifications.
- **FR-86**: The system shall support email communications.
- **FR-87**: The system shall maintain communication logs.

4. External Interface Requirements

4.1 User Interfaces

The CORADS system will include the following user interfaces:

4.1.1 Advertiser Mobile Application

- **UI-1**: Login screen with username and password fields.
- **UI-2**: Registration screen with required advertiser information.
- **UI-3**: Dashboard displaying active campaigns and wallet balance.
- **UI-4**: Campaign creation form with fields for campaign parameters.
- **UI-5**: Campaign list screens (active, pending, completed).
- **UI-6**: Wallet screen showing balance and transaction history.
- **UI-7**: Map interface showing campaign drivers.
- **UI-8**: Profile management screen.
- **UI-9**: Support contact screen.

4.1.2 Driver Mobile Application

- **UI-10**: Login screen with username and password fields.
- **UI-11**: Dashboard displaying active campaigns and earnings.
- **UI-12**: Campaign request notification and response interface.
- **UI-13**: Navigation map to installation centers.
- **UI-14**: Camera interface for verification photos.
- **UI-15**: Wallet screen showing earnings and history.
- **UI-16**: Profile management screen.
- **UI-17**: Support contact screen.

4.1.3 Admin Web Portal

- **UI-18**: Login screen with username and password fields.
- **UI-19**: Dashboard displaying system statistics.
- **UI-20**: User management interface.
- **UI-21**: Campaign management interface.
- **UI-22**: Configuration screens (coupons, vouchers, car types).
- **UI-23**: Installation center management screen.
- **UI-24**: Support ticket management interface.
- **UI-25**: Reporting and analytics interface.

4.2 Hardware Interfaces

The CORADS system will interface with the following hardware components:

4.2.1 Mobile Devices

- **HW-1**: Android smartphones (minimum API level 23/Android 6.0).
- **HW-2**: Device GPS for location tracking.
- **HW-3**: Device camera for verification photos.
- **HW-4**: Device storage for application data.

4.2.2 Car Displays

- **HW-5**: Advertisement displays mounted on vehicles.
- **HW-6**: Installation equipment at installation centers.

4.3 Software Interfaces

The CORADS system will interface with the following software systems:

4.3.1 External Services

- **SW-1**: AWS cloud services for hosting.

    - Protocol: HTTPS

- Data Format: JSON
- Authentication: API Keys/AWS IAM

- **SW-2**: Firebase for real-time communication.

  - Protocol: WebSocket/HTTPS
  - Data Format: JSON
  - Authentication: Firebase Auth

- **SW-3**: Google Maps API for location services.

  - Protocol: HTTPS
  - Data Format: JSON
  - Authentication: API Keys

- **SW-4**: JAZZ CASH payment gateway.

  - Protocol: HTTPS
  - Data Format: JSON
  - Authentication: Merchant ID and Key

4.3.2 Internal Components

- **SW-5**: Android Application to Backend API.

  - Protocol: HTTPS
  - Data Format: JSON
  - Authentication: JWT

- **SW-6**: Admin Portal to Backend API.

  - Protocol: HTTPS
  - Data Format: JSON
  - Authentication: JWT

- **SW-7**: Database Interface.

  - Protocol: SQL
  - Authentication: Database credentials

4.4 Communications Interfaces

The CORADS system will utilize the following communication protocols:

- **COM-1**: HTTPS for secure API communications.
- **COM-2**: WebSockets for real-time updates.

- **COM-3**: Firebase Cloud Messaging for push notifications.
- **COM-4**: SMTP for email communications.

5. Non-Functional Requirements

5.1 Performance Requirements

- **NFR-1**: The mobile applications shall load within 3 seconds on devices meeting the minimum specifications.
- **NFR-2**: The system shall support a minimum of 1000 concurrent users.
- **NFR-3**: Location tracking updates shall occur at least every 30 seconds.
- **NFR-4**: API responses shall be delivered within 2 seconds under normal load.
- **NFR-5**: The system shall handle up to 100 campaign creations per minute.
- **NFR-6**: Push notifications shall be delivered within 5 seconds of triggering events.

5.2 Safety Requirements

- **NFR-7**: The driver application shall not require interaction while the vehicle is in motion.
- **NFR-8**: The system shall comply with local road safety regulations regarding vehicle modifications.
- **NFR-9**: The system shall verify that advertisements do not obstruct driver visibility.
- **NFR-10**: The system shall monitor and flag potentially unsafe driving behaviors.

5.3 Security Requirements

- **NFR-11**: User authentication shall employ secure password hashing.
- **NFR-12**: All API communications shall be encrypted using TLS.
- **NFR-13**: Payment information shall be handled in compliance with PCI DSS standards.
- **NFR-14**: User sessions shall timeout after 30 minutes of inactivity.
- **NFR-15**: The system shall implement rate limiting to prevent brute force attacks.
- **NFR-16**: The system shall maintain audit logs of all administrative actions.

5.4 Software Quality Attributes

- **NFR-17**: Usability: The interface shall be intuitive enough that a new user can complete basic tasks without assistance.
- **NFR-18**: Reliability: The system shall maintain 99.5% uptime during operating hours.
- **NFR-19**: Availability: The system shall be available 24/7 with scheduled maintenance windows.
- **NFR-20**: Maintainability: The codebase shall follow consistent coding standards and include documentation.

- **NFR-21**: Portability: The mobile applications shall function on all Android devices meeting the minimum requirements.
- **NFR-22**: Scalability: The system shall be designed to scale horizontally to accommodate growth.

5.5 Business Rules

- **BR-1**: Drivers must complete a verification process before accepting campaigns.
- **BR-2**: Advertisers must maintain a minimum wallet balance to create campaigns.
- **BR-3**: Campaign pricing shall be based on duration, number of cars, and location.
- **BR-4**: Drivers shall receive payment only for verified campaign participation.
- **BR-5**: Advertisements must comply with local advertising regulations.
- **BR-6**: System shall retain campaign data for a minimum of 1 year.

6. Other Requirements

6.1 Database Requirements

- **DR-1**: The system shall use a relational database for structured data storage.
- **DR-2**: The database shall maintain referential integrity between related entities.
- **DR-3**: The database shall be backed up daily with a retention period of 30 days.
- **DR-4**: Database performance shall be monitored, with query optimization applied as needed.

6.2 Legal Requirements

- **LR-1**: The system shall comply with local data protection regulations.
- **LR-2**: User terms and conditions shall be clearly presented during registration.
- **LR-3**: The system shall implement appropriate data retention and deletion policies.
- **LR-4**: The system shall maintain records of user consent for data processing.

7. System Testing

7.1 Testing Strategy

The CORADS system will undergo various testing procedures to ensure it meets all specified requirements:

7.1.1 Unit Testing

- **UT-1**: All individual modules shall be tested independently.

- **UT-2**: Test cases shall cover normal, boundary, and error cases.
- **UT-3**: Unit tests shall achieve at least 80% code coverage.

7.1.2 Integration Testing

- **IT-1**: Integration tests shall verify correct interactions between system components.
- **IT-2**: All API endpoints shall be tested for correct request and response handling.
- **IT-3**: Integration tests shall verify data consistency across system components.

7.1.3 System Testing

- **ST-1**: End-to-end system testing shall verify complete workflows.
- **ST-2**: System testing shall verify all functional requirements.
- **ST-3**: Performance testing shall verify the system meets performance requirements under expected load.
- **ST-4**: Security testing shall identify potential vulnerabilities.

7.1.4 User Acceptance Testing

- **UAT-1**: Representatives from each user class shall participate in acceptance testing.
- **UAT-2**: Acceptance testing shall verify the system meets business requirements.
- **UAT-3**: Feedback from acceptance testing shall be incorporated into the final system.

7.2 Testing Methods

7.2.1 Functional Testing

- Test each function of the application against the functional requirements.
- Verify input validation, processing, and output.
- Test all user workflows and business processes.

7.2.2 Usability Testing

- Test the system's user interface for ease of use.
- Verify the system meets accessibility standards.
- Gather feedback on user experience.

7.2.3 Performance Testing

- Measure response times under various load conditions.
- Test the system's capacity and scalability.
- Identify performance bottlenecks.

7.2.4 Security Testing

- Test for vulnerabilities in authentication and authorization.

- Verify data encryption and protection.
- Test for resistance to common attack vectors.

7.2.5 Compatibility Testing

- Test on various Android device models and OS versions.
- Test on different web browsers for the admin portal.
- Verify operation under different network conditions.

7.3 Test Documentation

The following test documentation will be maintained:

- **Test Plan**: Outlines the overall testing approach and schedule.
- **Test Cases**: Detailed scenarios to be tested with expected results.
- **Test Reports**: Results of executed tests, including pass/fail status and issues found.
- **Defect Reports**: Documentation of identified issues, their severity, and resolution.

8. Conclusion

The CORADS (Car On Roads Advertisement) system represents an innovative solution to the challenges faced by small businesses in advertising their products and services. By leveraging moving cars as mobile advertisement platforms, CORADS provides a cost-effective alternative to expensive digital advertising channels while offering several advantages over traditional static advertisements.

The system will connect advertisers with drivers through user-friendly mobile applications, supported by a comprehensive administrative backend. Real-time tracking, performance analytics, and transparent payment mechanisms will ensure all stakeholders derive value from the platform.

This Software Requirements Specification document has outlined the functional and non-functional requirements that will guide the development of CORADS. Adherence to these requirements will ensure that the final system meets the needs of advertisers seeking affordable advertising options, drivers looking for additional income opportunities, and administrators managing the platform.

The development team should use this document as a foundation for system design, implementation, and testing. Regular review and updates to this specification will be necessary as the project progresses and additional requirements are identified or refined.

9. Supporting Information

9.1 Glossary

- **CORADS**: Car On Roads Advertisement System
- **OTP**: One-Time Password, used for user verification
- **Campaign**: An advertising initiative created by an advertiser
- **JWT**: JSON Web Token, used for secure authentication
- **FCM**: Firebase Cloud Messaging, used for push notifications
- **API**: Application Programming Interface
- **SDK**: Software Development Kit
- **AWS**: Amazon Web Services, cloud hosting platform
- **Heat Map**: Visual representation of data showing concentration of advertisement display
- **Installation Center**: Facility where advertisements are installed on vehicles

9.2 Analysis Models

9.2.1 Data Flow Diagram

The data flow within the CORADS system follows these primary paths:

- Advertiser creates campaign → System processes request → Campaign details stored in database
- System identifies suitable drivers → Notification sent to drivers → Drivers accept/reject
- Driver arrives at installation center → Admin verifies → Campaign starts
- Driver moves around city → System tracks location → Data collected for analytics
- Campaign completes → System generates reports → Advertiser views performance

9.2.2 Entity-Relationship Diagram

Key entities and their relationships:

- Advertiser (1) - creates - (N) Campaigns
- Campaign (1) - assigned to - (N) Drivers
- Driver (1) - has - (1) Car
- Admin (1) - manages - (N) Installation Centers
- Campaign (1) - has - (N) Verification Records
- Advertiser (1) - has - (1) Wallet
- Driver (1) - has - (1) Wallet

9.3 References

1. IEEE Standard 830-1998, IEEE Recommended Practice for Software Requirements Specifications.

2. CORADS Thesis Document, "Advertisement on Roads", 2021.

3. Android Developer Guidelines, Google, https://developer.android.com/docs

4. AWS Documentation, Amazon Web Services, https://docs.aws.amazon.com

5. Firebase Documentation, Google, https://firebase.google.com/docs

6. Google Maps Platform Documentation, https://developers.google.com/maps/documentation

7. Payment Card Industry Data Security Standard (PCI DSS), https://www.pcisecuritystandards.org

8. Software Engineering: A Practitioner's Approach, Roger S. Pressman, 8th Edition.

9.4 Appendices

Appendix A: Sample User Interfaces

[Placeholder for wireframes and UI mockups of key screens]

Appendix B: Data Dictionary

1. **User**

   - UserID (Primary Key)
   - Name
   - Email
   - Phone
   - Password (Hashed)
   - UserType (Advertiser/Driver/Admin)
   - CreationDate
   - LastLoginDate
   - Status

2. **Campaign**

   - CampaignID (Primary Key)
   - AdvertiserID (Foreign Key)
   - Title
   - Description
   - StartDate

- EndDate
- TargetArea
- Budget
- Status
- CreationDate
- Images[]

3. **Driver**

- DriverID (Foreign Key to User)
- CarType
- LicenseNumber
- RegistrationNumber
- Rating
- AvailabilityStatus
- CurrentLocation
- LastActive

4. **CampaignAssignment**

- AssignmentID (Primary Key)
- CampaignID (Foreign Key)
- DriverID (Foreign Key)
- AssignmentDate
- CompletionDate
- Status
- PaymentAmount
- VerificationRecords[]

Appendix C: Timeline and Milestones

1. **Requirements Phase**: 8 weeks

- Requirements gathering
- SRS document creation
- Stakeholder approval

2. **Design Phase**: 16 weeks

- System architecture design
- Database design
- UI/UX design
- Design document creation

3. **Implementation Phase**: 16 weeks

   - Mobile application development
   - Admin portal development
   - Backend API development
   - Integration with third-party services

4. **Testing Phase**: 4 weeks

   - Unit testing
   - Integration testing
   - System testing
   - User acceptance testing

5. **Deployment Phase**: 4 weeks

   - System deployment
   - User training
   - Go-live support