

1.Introduction

1.1 Purpose

This Software Requirements Specification (SRS) document outlines the requirements for the development of Automated Logistics Transportation System. The primary purpose of this SRS is to provide a comprehensive and detailed understanding of the functional and non-functional requirements that will guide the design and implementation of the Automated Logistics Transportation System.

1.1.1 Purpose of the SRS

The purpose of this SRS is to serve as a blueprint for the development team, guiding them in the creation of a robust, user-friendly, and efficient Automated Logistics Transportation System. It outlines the specific features and functionalities that system must incorporate to meet user needs and industry standards.

1.1.2 Intended Audience

The intended audience for this SRS includes:

- **Development Team:** This document is crucial for the development team, including software engineers, architects, and project managers, to understand the system's requirements and implement them effectively.
- **Quality Assurance Team:** The QA team will use this document as a basis for creating test cases and ensuring that the developed system meets the specified requirements.
- **Project Stakeholders:** Stakeholders, including product managers, business analysts, and investors, will refer to this document to understand the scope and functionality of the Automated Logistics Transportation System.
- **Future Maintenance Teams:** In the future, maintenance teams may refer to this document to understand the original specifications of the system and make informed updates or enhancements.

1.2 Scope

1.2.1 Software Product(s) Identification

The software product covered by this Software Requirements Specification (SRS) is the Automated Logistics Transportation System.

1.2.2 Functionality Overview

Automated Logistics Transportation System is designed to facilitate seamless interactions between users and truck owners, streamlining the logistics transfer process. The software product will:

- ✓ Enable users to register and create logistics transfer requests.
- ✓ Provide a platform for truck owners to view and accept user requests based on availability.
- ✓ Incorporate real-time tracking features for users to monitor the location of their cargo during transit.
- ✓ Implement a rating and review system for users and truck owners to provide feedback.

1.2.3 Application of the Software

The application of the system extends to:

- **Users:** Individuals or businesses in need of logistics transfer services.
- **Truck Owners/Drivers:** Individuals or businesses owning trucks and providing logistics transfer services.

1.2.4 Objectives and Goals

The primary objectives and goals of the Automated Logistics Transportation System are to:

- ✓ Enhance the efficiency of logistics operations by connecting users with nearby truck owners.
- ✓ Reduce wait times for users and provide a reliable income source for truck owners.
- ✓ Improve transparency through real-time tracking.
- ✓ Foster a positive user experience through a user-friendly interface and a robust feedback system.

1.2.5 Exclusions

This SRS focuses on the core functionality Automated Logistics Transportation System. Excluded from this document are detailed design specifications and implementation specifics, as this document is intended to guide development without delving into design details.

1.3 Document Conventions

This Software Requirements Specification (SRS) adheres to the following standards and typographical conventions:

1.3.1 Formatting

- **Section Headings:** Major section headings are formatted in bold and use a hierarchical numbering system for clarity (e.g., 1, 1.1, 1.1.1).
- **Subheadings:** Subheadings within sections are formatted in bold italic.
- **Text Emphasis:** Important terms or concepts are italicized for emphasis.

1.3.2 Requirements Prioritization

- **Priority Levels:** Each requirement statement includes a priority level to indicate its relative importance. Priority levels are defined as follows:
 - ✓ **High (H):** Critical features that must be implemented for the system's core functionality.
 - ✓ **Medium (M):** Important features that enhance the system's overall performance and user experience.
 - ✓ **Low (L):** Desirable features that contribute to the system's completeness but are not critical.

1.3.3 Visual Conventions

- ✓ **Code Snippets:** Any code snippets, commands, or technical details are presented in a monospaced font for clear distinction.

1.3.4 Inheritance of Priorities

- ✓ **Priority Inheritance:** Priorities assigned to higher-level requirements are assumed to be inherited by detailed requirements unless explicitly stated otherwise. This ensures consistency and alignment with the overall project goals.

1.3.5 Revision Tracking

- ✓ **Document Versioning:** The document is versioned, and changes are tracked to maintain a clear record of updates and modifications. The version number and date are specified at the beginning of the document.

1.4 Intended Audience and Reading Suggestions

1.4.1 Intended Audience

This Software Requirements Specification (SRS) is tailored for various stakeholders involved in the development, testing, and management of the Automated Logistics Transportation System. The intended audience includes:

- ❖ **Developers:** To understand the technical specifications and requirements necessary for implementing the "TransitFlow" system.
- ❖ **Project Managers:** To gain insights into the overall scope, objectives, and priorities of the project, facilitating effective project planning and management.
- ❖ **Testers:** To create test cases and verify that the developed system meets the specified requirements.
- ❖ **Marketing Staff:** To grasp the key features and benefits of the system for promotional and marketing purposes.
- ❖ **Users:** For a high-level understanding of the app's functionality and capabilities.
- ❖ **Documentation Writers:** To gather information for creating user manuals and technical documentation.

1.4.2 Reading Suggestions

The document is organized in a logical sequence to facilitate efficient reading. Readers are suggested to follow this sequence:

- ❖ **Introduction and Purpose (Section 1):** Provides an overview of the document's purpose, scope, and intended audience.
- ❖ **Scope (Section 1.2):** Describes the scope of Automated Logistics Transportation System, including what the software will and will not do, its application, and objectives.
- ❖ **Document Conventions (Section 1.3):** Outlines the standards and conventions followed in the document, including formatting, prioritization, and versioning.
- ❖ **Intended Audience and Reading Suggestions (Section 1.4):** Identifies the target audience and provides recommendations for how different stakeholders can approach the document.

- ❖ **Overall Description (Section 2):** Offers a high-level overview of the software product, including its perspective, user classes, and operating environment.
- ❖ **Functional Requirements (Section 3):** Details the specific functional requirements of the "TransitFlow" app, such as user registration, logistics request management, real-time tracking, and feedback systems.
- ❖ **Non-Functional Requirements (Section 4):** Highlights the non-functional aspects, including performance, security, usability, and reliability.
- ❖ **External Interfaces (Section 5):** Describes the interfaces with other systems, such as user interfaces and API integrations.
- ❖ **Other Requirements (Section 6):** Covers additional aspects such as documentation and testing.
- ❖ **Conclusion (Section 7):** Summarizes the SRS and emphasizes its importance in guiding the development of the Automated Logistics Transportation System.

1.5 Definitions, Acronyms, and Abbreviations.

1.5.1 Definitions

- ❖ **Transit Flow:** The logistics transfer app under development, connecting users and truck owners for efficient and Automated logistics services.
- ❖ **User:** An individual or business entity utilizing the system to request logistics transfer services.
- ❖ **Truck Owner/Driver:** An individual or business entity that owns a truck and provides logistics transfer services through the Automated Logistics Transportation System.
- ❖ **Real-time Tracking:** A feature allowing users to monitor the live location of their cargo during transit.
- ❖ **Rating and Review System:** A feature enabling users and truck owners to provide feedback on their experiences, contributing to the overall reputation of the participants.

1.5.2 Acronyms and Abbreviations

- ✓ **SRS:** Software Requirements Specification
- ✓ **API:** Application Programming Interface

- ✓ **QA:** Quality Assurance
- ✓ **iOS:** Apple Operating System
- ✓ **Android:** Mobile Operating System developed by Google
- ✓ **GPS:** Global Positioning System

1.5.3 Reference Documents

- ✓ **[Reference Document 1]:** Detailed technical documentation for external interfaces.
- ✓ **[Reference Document 2]:** User manuals and guides for end-users.
- ✓ **[Reference Document 3]:** API documentation for developers.

2.Overall Description

2.1 Product Perspective

The Automated Logistics Transportation System serves as a bridge between users and truck owners, fostering efficient and seamless goods delivery. In a dynamic and interconnected world, this platform seeks to simplify logistics operations by intelligently connecting those in need of transport services with available truck owners. By providing an intuitive interface for users to place orders and implementing a sophisticated driver matching algorithm, the system aims to optimize the entire logistics process.

2.1.1 System Interfaces

1. Mapping Service Interface

- **Functionality:** Provides geolocation services, calculates distances, offers real-time tracking, and optimizes route planning.
- **Interface Description:** The system will utilize a mapping service API to obtain geolocation data, calculate distances between pickup and delivery locations, track the progress of transportation requests, and optimize route planning for efficient deliveries.

2. Push Notification Service Interface:

- **Functionality:** Sends real-time updates and notifications to users and truck owners regarding order status.
- **Description:** The system integrates with a push notification service to send timely updates to users and truck owners. This interface ensures that relevant information reaches the intended recipients promptly.

4. User Authentication Interface:

- **Functionality:** Ensures secure and reliable user authentication during the registration and login processes.
- **Description:** The system interfaces with external user authentication services. This integration ensures that user credentials are validated securely and that access to the system is controlled.

2.1.2 Interfaces

1. User Interface (Mobile App):

- **Logical Characteristics:**
 - Intuitive and user-friendly graphical user interface (GUI).
 - Interactive design for easy navigation and seamless user experience.
 - Multi-platform support for both iOS and Android operating systems.
- **Optimization Aspects:**
 - Responsive design to accommodate various screen sizes and resolutions.
 - Touch-friendly controls for smooth interaction.
 - Streamlined workflows for efficient order placement and tracking.

2. Admin Panel Interface:

- **Logical Characteristics:**
 - Web-based interface accessible through standard web browsers.
 - Role-based access control to ensure secure and authorized usage.

- Comprehensive dashboard for quick oversight and management.
- **Optimization Aspects:**
 - Intuitive layout for efficient user interaction.
 - Compatibility with modern web browsers.
 - Efficient data visualization for analytics and reporting.

3. GPS Location Service Interface:

- **Logical Characteristics:**
 - Integration with device location service APIs.
 - Real-time location tracking for users and truck owners.
- **Optimization Aspects:**
 - Minimal battery consumption for extended usage.
 - Optimal accuracy settings for efficient logistics coordination.

4. Push Notification Service Interface:

- **Logical Characteristics:**
 - Integration with push notification services.
 - Real-time updates and notifications for users and truck owners.
- **Optimization Aspects:**
 - Customizable notification preferences for users.
 - Prompt delivery of critical updates.

2.1.3 Hardware Interfaces

1. Mobile Devices (iOS and Android):

- **Characteristics:** Interfaces with device components (GPS, camera).
- **Configurations:** Compatible with iOS , Android .

2. Web Servers (Hosting the System):

- Characteristics: Communicates with server hardware.
- Configurations: Standard web server.

3. Database Servers (User, Truck Owner, Admin):

- Characteristics: Interfaces with database server hardware.
- Configurations: Standard database server.

2.1.4 Software Interfaces

1. Flutter Framework

- Name: Flutter
- Mnemonic: FLU
- Specification number: N/A
- Version number: The specific version of Flutter being used (e.g., 2.5.0)
- Source: <https://flutter.dev/>

Interface Purpose: Flutter is a framework used for building cross-platform mobile applications. It provides the development tools, libraries, and APIs necessary to create the user interface and handle interactions for the truck transportation system.

Interface Definition: The interface with Flutter involves utilizing its APIs, widgets, and development environment to implement the mobile application interface of the truck transportation system. This includes designing UI components, managing navigation, handling user input, and integrating with other system components.

2. NET Web API

- Name: .NET Web API
- Mnemonic: NETAPI
- Specification number: N/A

- Version number: The specific version of .NET being used (e.g., .NET 5.0)
- Source: <https://dotnet.microsoft.com/>

Interface Purpose: .NET Web API is a framework for building HTTP-based web services using the .NET platform. It provides the necessary tools and libraries to create the backend API that powers the functionality of the truck transportation system.

Interface Definition: The interface with .NET Web API involves designing and implementing the API endpoints, request handling, and response generation. It includes defining the message content and format, such as JSON or XML, for communication between the frontend (mobile application or web-based interface) and the backend API.

2.1.5 Communications Interfaces

1. Mobile App - Backend Communication:

- Interface: HTTPS/RESTful API
- Description: Facilitates secure communication between the Flutter mobile app and the .NET Web API backend. Utilizes standard HTTPS protocols for data transmission and follows RESTful principles for API endpoints.

2. Backend - Database Communication:

- Interface: MongoDB Native Protocol
- Description: Manages communication between the .NET Web API backend and the MongoDB database. Utilizes native MongoDB protocols for efficient data retrieval and storage.

2.1.6 Memory Constraints

1. Mobile App (Flutter):

- Primary Memory (RAM): The mobile app is optimized to function on devices with a minimum of 2GB of RAM for smooth performance. However, it is designed to be scalable and adaptive to varying device capabilities.

2. Backend Server (.NET Web API):

- Primary Memory (RAM): The .NET Web API backend should be deployed on servers with sufficient memory capacity based on the expected concurrent user load. The recommended minimum is 4GB of RAM.

3. Database Server (MongoDB):

- Primary Memory (RAM): The MongoDB database server should have adequate memory for efficient data caching and retrieval. A recommended minimum of 4GB of RAM is suggested.

2.1.7 Operations

1. Modes of Operations:

- User Modes:
 - Truck Owner Mode: Truck owners operate the app to manage and fulfill delivery requests.
 - User Mode: Users engage with the app to place delivery orders and track shipments.
- System Modes:
 - Normal Operation: The system functions regularly to facilitate order processing, driver matching, and real-time tracking.
 - Maintenance Mode: Periods where system updates or maintenance tasks are executed. Users are informed of any temporary disruptions.

2. Periods of Operations:

- Interactive Operations:
 - Users and truck owners can interact with the system during standard operational hours, allowing for order placement, tracking, and communication.

- Unattended Operations:
 - The system performs unattended operations, such as driver matching algorithms and order processing, continuously. However, regular maintenance activities may briefly interrupt user interactions.

3. Data Processing Support Functions:

- Order Processing:
 - The system processes user requests for goods delivery, matching them with available truck owners based on location and capacity.
- Real-Time Tracking:
 - Continuous tracking of the delivery status to provide users and truck owners with up-to-date information.
- User Authentication:
 - Secure user authentication during registration, login, and financial transactions.

4. Backup and Recovery Operations:

- Regular Backup:
 - The system performs scheduled backups of critical data, ensuring data integrity and availability.
- Recovery Mechanisms:
 - In the event of a system failure, the system implements recovery mechanisms to restore functionality swiftly. This may involve data rollback or redundancy strategies.
- Notification of Downtime:
 - Users and truck owners are informed of planned maintenance activities or potential system downtime to manage expectations.

2.1.8 Site Adaptation Requirements

The Automated Logistic Transport System requires specific data and initialization sequences for

successful deployment at customer sites. Upon installation, essential data, including user roles, preferences, and access permissions, must be initialized in the user database. Similarly, the truck owner database needs to be set up with details such as truck specifications, availability status, and geographic information. To adapt the software to a particular installation, the system should offer site-specific features that allow administrators to configure geographic parameters, pricing models.

2.2 Product Functions

1. User Management:

- Registration and authentication for both users and truck owners.
- User profile management, allowing users to update personal information and preferences.

2. Order Placement and Tracking:

- Users can place delivery orders, specifying pickup and delivery locations.
- Real-time tracking functionality for users to monitor the status and location of their shipments

3. Truck Owner Operations:

- Truck owners can register their vehicles, providing details on capacity and availability.
- Accepting or rejecting delivery requests based on availability and location.

4. Admin Panel Operations:

- Comprehensive dashboard for administrators to oversee system activities and monitor performance.

- User and truck owner management, with the ability to modify roles and permissions.

5. Communication and Notifications:

- In-app messaging system for communication between users and truck owners.
- Push notifications for real-time updates on order status, confirmations, and other relevant information.

6. Data Backup and Recovery:

- Scheduled backups of critical data to ensure data integrity.
- Recovery mechanisms to swiftly restore functionality in case of system failure.

2.3 User Classes and Characteristics

1. User Class: Regular Users

- Frequency of Use: High, as users will regularly place delivery orders.
- Subset of Functions Used: Primarily order placement, tracking, and communication features.
- Technical Expertise: Varied, but generally moderate technology proficiency.
- Security/Privilege Levels: Standard user privileges with access to order-related functionalities.
- Educational Level/Experience: Diverse educational backgrounds and experiences.

2. User Class: Truck Owners

- Frequency of Use: Frequent, especially during available hours for order acceptance.
- Subset of Functions Used: Mainly truck owner operations, including order acceptance/rejection.
- Technical Expertise: Moderate to high, as truck owners need proficiency in using the app for business operations.
- Security/Privilege Levels: Enhanced privileges for order management and vehicle registration.
- Educational Level/Experience: Varied, with a focus on business operations and logistics.

3. User Class: Administrators

- Frequency of Use: Regular, for overseeing system activities and performing administrative tasks.
- Subset of Functions Used: Admin panel operations, system monitoring, and user/truck owner management.
- Technical Expertise: High, with in-depth knowledge of system functionality and administration.
- Security/Privilege Levels: Highest privileges for system control and configuration.
- Educational Level/Experience: Technical education and experience in system administration.

2.4 Operating Environment

The Automated Logistics Transportation System is designed to operate in the following environment:

1. Hardware Platform:

- Mobile Devices (for Users and Truck Owners):
 - Compatible with iOS and Android platforms.
 - Recommended hardware specifications: 2GB RAM, and GPS capability.
- Web Servers (Hosting .NET Web API):
 - Standard web server hardware with sufficient resources for handling concurrent requests.
- Database Servers (Hosting MongoDB):
 - Server hardware with adequate processing power and storage capacity.

2. Software Components and Applications:

- Mobile App (Users and Truck Owners):
 - Developed using Flutter framework (compatible with Flutter-supported IDEs).

- Backend Server (.NET Web API):
 - Developed using .NET Core framework (compatible with ASP.NET Core-supported IDEs).

3. Other Considerations:

- **Network Requirements:** The system requires a stable internet connection for real-time communication and data synchronization.
- **Security Software:** Implementation of standard security practices and protocols to ensure data integrity and user privacy.

2.5 Design and Implementation Constraints

1. Corporate and Regulatory Policies:

- Adherence to corporate policies regarding data privacy, security, and ethical business practices.
- Compliance with relevant regulatory frameworks, such as data protection laws and financial transaction regulations.

2. Security Considerations:

- Implementation of encryption protocols for secure data transfer and storage.
- User authentication and authorization mechanisms to ensure data privacy and system security.

3. Language Requirements:

- The system must support local languages to cater to a diverse user base.
- Language selection and localization must be implemented for user interfaces and communications.

4. Interfaces to Other Applications:

- Integration with specific third-party services, such as map integration and Firebase Cloud Messaging (FCM) for push notifications.

2.6 User Documentation

User Manuals:

- In-depth manuals for regular users, truck owners, and administrators, covering system functionalities, navigation guides, and troubleshooting procedures.

2.7 Constraints

1. **Regulatory Policies:** Compliance with data protection laws, and privacy regulations.
2. **Hardware Limitations:** Adherence to specified hardware requirements for mobile devices, servers, and databases to maintain optimal performance.
3. **Interface to Other Applications:** Integration constraints with third-party services, such as map integration and push notification services.
4. **Parallel Operation:** Consideration of concurrent operations, especially in handling multiple delivery requests and real-time tracking updates.
5. **Control Functions:** Restrictions on control functions to prevent unauthorized access and manipulation of critical system settings.
6. **Safety and Security Considerations:** Prioritization of safety and security measures to protect user data, and ensure a secure environment for all system users.

2.8 Assumptions and Dependencies

Assumptions:

1. **Third-Party Service Availability:** Assuming the continued availability and reliability of third-party services, google map API and Firebase Cloud Messaging (FCM) for push notifications.
2. **Stable Internet Connection:** Assuming users will have a stable internet connection for real-time communication and data synchronization.
3. **Hardware and Operating System Consistency:** Assuming users' devices meet specified hardware, requirements and run compatible operating systems.

Dependencies:

1. **Third-Party Integrations:** Dependency on successful integration with third-party services, such as Stripe for secure payment transactions.
2. **Development Frameworks:** Dependency on the stability and updates of development frameworks, such as Flutter for the mobile app and .NET Core for the backend.

2.9 Apportioning of Requirements

In consideration of project constraints and resource availability, certain requirements may be apportioned to future versions of the Automated Logistics Transportation System.

Prioritization will be based on customer input, project planning, and iterative development considerations. The apportioning of requirements is as follows:

Phase 1 - Initial Release:

1. **User Management and Order Placement:** Initial focus on user registration, authentication, and basic order placement functionalities for both regular users and truck owners.
2. **Real-Time Tracking and Communication:** Implementation of core features such as real-time tracking of shipments and basic in-app communication between users and truck owners.
3. **Basic Admin Panel Operations:** Implementation of fundamental admin panel operations for system monitoring and user/truck owner management.

Phase 2 - Iterative Enhancements:

1. **Enhanced User Features:** Introduction of advanced user features, including profile customization and order history tracking.
2. **Truck Owner Operations Refinement:** Iterative improvements to truck owner operations, such as enhanced order acceptance/rejection mechanisms and detailed vehicle management.
3. **Advanced Admin Panel Functionality:** Expansion of admin panel capabilities, including detailed analytics, reporting tools, and refined system control.
4. **Multilingual Support:** Implementation of support for multiple languages to enhance accessibility.

Future Phases - Continuous Improvement:

1. **Optimization for Various Devices:** Ongoing optimization and testing for compatibility with a broader range of mobile devices and operating system versions.
2. **Machine Learning Algorithms:** Exploration and integration of machine learning algorithms for more efficient driver matching and route optimization.
3. **Expanded Geographic Configuration:** Iterative expansion of geographic configuration options for administrators to tailor the system to diverse markets.
4. **Additional Third-Party Integrations:** Exploration and incorporation of additional third-party services for expanded functionality.

3. Specific Requirement

3.1 External Interface Requirements

3.1.1 User Interfaces

1. Logical Characteristics:
 - The mobile app interface for regular users shall follow a clean and intuitive design, emphasizing ease of use and accessibility.
 - Screen layouts shall be designed to provide a seamless and engaging user experience.
 - Graphical elements, such as icons and images, shall be chosen to enhance user understanding and navigation.
2. GUI Standards and Style Guides:
 - The mobile app shall adhere to established GUI standards for iOS and Android platforms, ensuring a consistent look and feel across devices.
 - Any relevant product family style guides, if applicable, shall be followed to maintain brand consistency.
3. Screen Layout Constraints:

- Screen layouts should be optimized for various screen sizes and resolutions, ensuring responsiveness on different devices.
4. Standard Buttons and Functions:
 - Common buttons such as "Submit," "Cancel," and "Back" shall be consistently placed for familiarity.
 - Standard functions like "Help" or "Contact Support" shall be easily accessible on every screen.
 5. Software Components Requiring UI:
 - User Registration Module
 - Order Placement and Tracking Module
 - Notifications and Communication Module

3.1.2 Hardware Interfaces

The hardware interfaces describe the logical and physical characteristics of the interactions between the Automated Logistics Transportation System and the underlying hardware components. These interfaces encompass supported device types, data and control interactions, and communication protocols.

1. **Supported Device Types:** The software product shall be compatible with a range of mobile devices.
2. **Data and Control Interactions:** The software shall communicate with the device hardware to facilitate various functionalities, such as GPS for real-time tracking, and secure storage for user data.
3. Communication Protocols:
 - Data transmission between the software product and the device hardware shall adhere to industry-standard communication protocols.
 - The mobile app shall use HTTPS for secure data transfer over the network.

- Device-specific protocols for accessing hardware components, such as location services, shall be implemented.
4. **Physical Characteristics:** The software shall optimize its performance based on the physical characteristics of the device, considering factors such as screen size, resolution, and processing power.
 5. **Network Connectivity:** The system shall operate seamlessly over both Wi-Fi and mobile data networks, adapting to variable network conditions.

3.1.3 Software Interfaces

The software interfaces define the connections between the Automated Logistics Transportation System and other specific software components, encompassing databases, operating systems, tools, libraries, and integrated commercial components.

3.1.3.1 Database Interface

1. **Database Management System:** The Automated Logistics Transportation System shall interact with MongoDB, version 4.4, as the chosen database management system.
2. **Data Items and Messages:**
 - Incoming:** User registration details, order information, delivery tracking data.
 - Outgoing:** Confirmation messages, system logs, and analytical data for reporting.
3. **Purpose:** The database interface is responsible for storing and retrieving user-related data, order details, and system logs to support the functionality of the Automated Logistics Transportation System

3.1.3.2 Operating System Interface

1. Supported Operating Systems: The mobile app shall interface with iOS and Android operating systems.

2. Data Items and Messages:

Incoming: User interactions, device-specific information.

Outgoing: Request for system functionalities, device control signals.

3. Purpose:

The operating system interface facilitates communication between the Automated Logistics Transportation System and the mobile device, ensuring compatibility and optimal performance.

3.1.4 Communications Interfaces

1. Network Server Communication:

The Automated Logistics Transportation System shall utilize HTTPS (Hypertext Transfer Protocol Secure) for secure communication between the mobile app and the backend server.

Communication protocols, including RESTful APIs, shall be employed for data exchange.

2. Data Transfer Rates:

The system shall support high-speed data transfer rates to ensure real-time updates and a responsive user experience.

3.2 Functional Requirements

The functional requirements outline the major services and system features provided by the Automated Logistics Transportation System. These requirements are organized based on the logical flow of the product, ensuring a comprehensive understanding of the system's capabilities.

3.2.1 User Registration and Authentication

1. Feature Description:

The system shall provide a user registration module for both regular users and truck owners.

2. Functional Requirements:

- 3.2.1.1 User Registration:
 - Regular users and truck owners can register with the system by providing required details, including name, contact information, and address.
 - User registration shall include account verification through email or SMS.
- 3.2.1.2 User Authentication:
 - Authenticated access to the system shall be ensured through secure login credentials.

3.2.2 Order Placement and Tracking

1. Feature Description:

- The system shall facilitate users to place orders for item delivery and track the status in real-time.

2. Functional Requirements:

- 3.2.2.1 Order Placement:
 - Users can initiate order requests by specifying pickup and delivery locations, item details, and preferred delivery time.
- 3.2.2.2 Real-Time Tracking:
 - Users and truck owners can track the real-time status and location of the delivery.

3.2.5 Admin Panel Operations

1. Feature Description:

- The system shall provide an admin panel for system monitoring and user/truck owner management.

2. Functional Requirements:

- 3.2.5.1 User Management:
 - Admins can manage user accounts, including approval, suspension, and deletion.
- 3.2.5.2 Analytics and Reporting:
 - The admin panel shall provide analytics and reporting tools for system performance, order trends, and user behavior.

3.3 Nonfunctional Requirements

3.3.1 Performance Requirements

1. **Response Time:** The system shall respond to user interactions within 5 seconds, providing a seamless and responsive user experience.
2. **Data Transfer Speed:** The mobile app shall support a minimum data transfer rate of 3G for mobile network connections and broadband for Wi-Fi to ensure optimal performance.

3.3.2 Reliability and Availability

- 4 **System Uptime:** The system shall strive for 98.0% uptime, ensuring availability for users at all times.
- 5 **Fault Tolerance:** The system shall incorporate fault-tolerant mechanisms to handle unexpected failures gracefully and minimize service disruptions.

3.3.3 Security Requirements

User Authentication:

- The system shall implement a secure user authentication mechanism to verify the identity of users during login.
- Authentication shall be based on unique credentials, username and password.