Software Requirements Specification (SRS)

Travectio Fleet Management System

**Document Version:** 1.0  
**Date:** August 12, 2025  
**Project:** Travectio Solutions Fleet Management Platform  
**Classification:** Enterprise SaaS Application

Table of Contents

1. [Introduction](https://replit.com/@rrivera7/TravectioSolutions#1-introduction)
2. [Overall Description](https://replit.com/@rrivera7/TravectioSolutions#2-overall-description)
3. [System Features](https://replit.com/@rrivera7/TravectioSolutions#3-system-features)
4. [External Interface Requirements](https://replit.com/@rrivera7/TravectioSolutions#4-external-interface-requirements)
5. [Non-Functional Requirements](https://replit.com/@rrivera7/TravectioSolutions#5-non-functional-requirements)
6. [Technical Requirements](https://replit.com/@rrivera7/TravectioSolutions#6-technical-requirements)
7. [Security Requirements](https://replit.com/@rrivera7/TravectioSolutions#7-security-requirements)
8. [Database Requirements](https://replit.com/@rrivera7/TravectioSolutions#8-database-requirements)
9. [Integration Requirements](https://replit.com/@rrivera7/TravectioSolutions#9-integration-requirements)
10. [Deployment Requirements](https://replit.com/@rrivera7/TravectioSolutions#10-deployment-requirements)

1. Introduction

1.1 Purpose

This document specifies the software requirements for Travectio Fleet Management System, a comprehensive enterprise platform for trucking companies to manage fleet operations, load profitability, and operational analytics.

1.2 Scope

Travectio Solutions provides real-time fleet management capabilities including:

* Multi-tenant truck fleet management
* Load tracking and profitability analysis
* Automated operational cost calculations
* Real-time analytics and reporting
* ELD integration for compliance tracking
* Load board integration for market opportunities

1.3 Definitions and Acronyms

* **CPM**: Cost Per Mile
* **ELD**: Electronic Logging Device
* **HOS**: Hours of Service
* **SaaS**: Software as a Service
* **API**: Application Programming Interface
* **UI/UX**: User Interface/User Experience

1.4 References

* FMCSA Hours of Service Regulations
* DOT Commercial Vehicle Safety Requirements
* Industry Standard Cost Accounting Practices

2. Overall Description

2.1 Product Perspective

Travectio is a standalone web-based SaaS platform designed for trucking companies of all sizes, from owner-operators to large fleets. The system integrates with existing industry tools while providing comprehensive operational oversight.

2.2 Product Functions

Core Functions:

* Fleet asset management and tracking
* Load management and profitability calculation
* Operational cost analysis and reporting
* Driver assignment and management
* Real-time analytics dashboard
* Multi-provider integration support

Administrative Functions:

* Multi-tenant user management
* Role-based access control
* System analytics and monitoring
* Customer support tools

2.3 User Classes

2.3.1 Founder Users

* System-wide administrative access
* Customer support capabilities
* Business analytics and insights
* User management and oversight

2.3.2 Customer Users (Fleet Operators)

* Fleet management access
* Load tracking and analysis
* Cost management tools
* Operational reporting

2.3.3 Demo Users

* Limited access to sample data
* Feature demonstration capabilities
* Restricted to demo fleet data

2.4 Operating Environment

Client Environment:

* Modern web browsers (Chrome, Firefox, Safari, Edge)
* Mobile devices (responsive design)
* Internet connectivity required

Server Environment:

* Node.js 20.19.3 LTS runtime
* PostgreSQL database
* Cloud hosting (Render.com)
* SSL/TLS encryption

3. System Features

3.1 Authentication and Authorization

3.1.1 Description

Secure multi-tenant authentication system with role-based access control.

3.1.2 Functional Requirements

* **REQ-AUTH-001**: System shall support OpenID Connect authentication
* **REQ-AUTH-002**: System shall maintain secure session management
* **REQ-AUTH-003**: System shall implement role-based access control
* **REQ-AUTH-004**: System shall support founder, customer, and demo user types
* **REQ-AUTH-005**: System shall enforce data isolation between tenants

3.1.3 Priority

**High** - Critical for system security and multi-tenant operation

3.2 Fleet Management

3.2.1 Description

Comprehensive truck fleet management with detailed cost tracking and operational metrics.

3.2.2 Functional Requirements

* **REQ-FLEET-001**: System shall allow truck registration with equipment details
* **REQ-FLEET-002**: System shall track fixed and variable operational costs
* **REQ-FLEET-003**: System shall calculate real-time cost-per-mile metrics
* **REQ-FLEET-004**: System shall support truck editing and deletion
* **REQ-FLEET-005**: System shall maintain truck operational history
* **REQ-FLEET-006**: System shall support driver assignment to trucks

3.2.3 Priority

**High** - Core business functionality

3.3 Load Management

3.3.1 Description

Complete load tracking system with profitability analysis and route optimization.

3.3.2 Functional Requirements

* **REQ-LOAD-001**: System shall record load details (origin, destination, cargo)
* **REQ-LOAD-002**: System shall calculate load profitability automatically
* **REQ-LOAD-003**: System shall track deadhead miles and costs
* **REQ-LOAD-004**: System shall provide load editing capabilities
* **REQ-LOAD-005**: System shall maintain load history and analytics
* **REQ-LOAD-006**: System shall calculate distance using geographic coordinates

3.3.3 Priority

**High** - Essential for operational profitability

3.4 Analytics and Reporting

3.4.1 Description

Real-time analytics dashboard with comprehensive operational metrics and business intelligence.

3.4.2 Functional Requirements

* **REQ-ANALYTICS-001**: System shall provide real-time operational dashboards
* **REQ-ANALYTICS-002**: System shall calculate fleet-wide performance metrics
* **REQ-ANALYTICS-003**: System shall support time-based filtering (weekly, bi-weekly, monthly, quarterly, bi-annual, and annually)
* **REQ-ANALYTICS-004**: System shall track revenue, costs, and profit margins
* **REQ-ANALYTICS-005**: System shall provide cross-tab data synchronization
* **REQ-ANALYTICS-006**: System shall support data export capabilities

3.3.3 Priority

**High** - Critical for business decision-making

3.5 Load Board Integration

3.5.1 Description

Multi-provider load board integration for market opportunity identification.

3.5.2 Functional Requirements

* **REQ-LOADBOARD-001**: System shall integrate with DAT load board
* **REQ-LOADBOARD-002**: System shall integrate with Truckstop.com
* **REQ-LOADBOARD-003**: System shall integrate with 123Loadboard
* **REQ-LOADBOARD-004**: System shall integrate with SuperDispatch
* **REQ-LOADBOARD-005**: System shall provide unified load search interface
* **REQ-LOADBOARD-006**: System shall support per-truck load board preferences

3.5.3 Priority

**Medium** - Important for operational efficiency

3.6 ELD Integration

3.6.1 Description

Electronic Logging Device integration for Hours of Service compliance and driver management.

3.6.2 Functional Requirements

* **REQ-ELD-001**: System shall integrate with Samsara ELD platform
* **REQ-ELD-002**: System shall integrate with KeepTruckin/Motive
* **REQ-ELD-003**: System shall integrate with Garmin ELD solutions
* **REQ-ELD-004**: System shall track HOS compliance status
* **REQ-ELD-005**: System shall monitor driver duty status
* **REQ-ELD-006**: System shall support per-truck ELD provider configuration

3.6.3 Priority

**Medium** - Important for regulatory compliance

4. External Interface Requirements

4.1 User Interfaces

4.1.1 General UI Requirements

* **REQ-UI-001**: Interface shall be responsive and mobile-friendly
* **REQ-UI-002**: Interface shall support dark theme for professional appearance
* **REQ-UI-003**: Interface shall provide intuitive navigation with sidebar
* **REQ-UI-004**: Interface shall display real-time data updates
* **REQ-UI-005**: Interface shall support offline functionality with data synchronization

4.1.2 Dashboard Requirements

* **REQ-UI-006**: Dashboard shall display key operational metrics prominently
* **REQ-UI-007**: Dashboard shall provide quick access to core functions
* **REQ-UI-008**: Dashboard shall support customizable metric displays
* **REQ-UI-009**: Dashboard shall show fleet status at a glance

4.2 Hardware Interfaces

* **REQ-HW-001**: System shall operate on standard web-capable devices
* **REQ-HW-002**: System shall support mobile device touch interfaces
* **REQ-HW-003**: System shall function with standard internet connectivity

4.3 Software Interfaces

4.3.1 Database Interface

* **REQ-DB-001**: System shall interface with PostgreSQL database
* **REQ-DB-002**: System shall use Drizzle ORM for database operations
* **REQ-DB-003**: System shall support connection pooling for performance

4.3.2 External API Interfaces

* **REQ-API-001**: System shall integrate with OpenAI API for intelligent features
* **REQ-API-002**: System shall interface with multiple load board APIs
* **REQ-API-003**: System shall integrate with ELD provider APIs
* **REQ-API-004**: System shall handle API rate limiting and error recovery

5. Non-Functional Requirements

5.1 Performance Requirements

* **REQ-PERF-001**: Page load times shall not exceed 3 seconds
* **REQ-PERF-002**: API response times shall be under 1 second for standard queries
* **REQ-PERF-003**: System shall support 50+ concurrent users without degradation
* **REQ-PERF-004**: Database queries shall be optimized for sub-second response
* **REQ-PERF-005**: System shall handle 1000+ fleet vehicles per customer

5.2 Reliability Requirements

* **REQ-REL-001**: System uptime shall exceed 99.9% availability
* **REQ-REL-002**: System shall handle graceful degradation during outages
* **REQ-REL-003**: Data backup shall occur automatically and continuously
* **REQ-REL-004**: System shall recover from failures within 5 minutes

5.3 Availability Requirements

* **REQ-AVAIL-001**: System shall be available 24/7 with minimal downtime
* **REQ-AVAIL-002**: Maintenance windows shall be scheduled during low-usage periods
* **REQ-AVAIL-003**: System shall provide status monitoring and alerting

5.4 Scalability Requirements

* **REQ-SCALE-001**: System shall scale horizontally to support growth
* **REQ-SCALE-002**: Database shall support partitioning for large datasets
* **REQ-SCALE-003**: System shall handle increased load through cloud auto-scaling

5.5 Usability Requirements

* **REQ-USABILITY-001**: New users shall complete basic tasks within 15 minutes
* **REQ-USABILITY-002**: Interface shall follow industry-standard UX patterns
* **REQ-USABILITY-003**: System shall provide contextual help and guidance
* **REQ-USABILITY-004**: Error messages shall be clear and actionable

6. Technical Requirements

6.1 Frontend Technology Stack

* **REQ-TECH-001**: Frontend shall use React 18.3.1 framework
* **REQ-TECH-002**: Frontend shall use TypeScript for type safety
* **REQ-TECH-003**: Frontend shall use Tailwind CSS for styling
* **REQ-TECH-004**: Frontend shall use React Query for state management
* **REQ-TECH-005**: Frontend shall use Vite for build tooling

6.2 Backend Technology Stack

* **REQ-TECH-006**: Backend shall use Node.js 20.19.3 LTS
* **REQ-TECH-007**: Backend shall use Express.js framework
* **REQ-TECH-008**: Backend shall use TypeScript throughout
* **REQ-TECH-009**: Backend shall implement RESTful API architecture

6.3 Database Requirements

* **REQ-TECH-010**: Database shall use PostgreSQL 16+
* **REQ-TECH-011**: Database shall use Drizzle ORM for schema management
* **REQ-TECH-012**: Database shall implement proper indexing for performance
* **REQ-TECH-013**: Database shall support ACID transactions

7. Security Requirements

7.1 Authentication Security

* **REQ-SEC-001**: System shall use OpenID Connect for secure authentication
* **REQ-SEC-002**: Passwords shall be securely hashed and salted
* **REQ-SEC-003**: Session tokens shall expire after reasonable time periods
* **REQ-SEC-004**: System shall implement multi-factor authentication support

7.2 Data Security

* **REQ-SEC-005**: All data transmission shall use HTTPS/TLS encryption
* **REQ-SEC-006**: Sensitive data shall be encrypted at rest
* **REQ-SEC-007**: System shall implement data isolation between tenants
* **REQ-SEC-008**: API keys and secrets shall be securely stored

7.3 Access Control

* **REQ-SEC-009**: System shall implement role-based access control (RBAC)
* **REQ-SEC-010**: User permissions shall be validated on every request
* **REQ-SEC-011**: System shall log all administrative actions
* **REQ-SEC-012**: Failed authentication attempts shall be monitored and limited

7.4 Privacy Requirements

* **REQ-PRIVACY-001**: User data shall be protected according to privacy regulations
* **REQ-PRIVACY-002**: Users shall have control over their data sharing preferences
* **REQ-PRIVACY-003**: System shall support data deletion requests
* **REQ-PRIVACY-004**: Analytics data shall respect user privacy settings

8. Database Requirements

8.1 Data Models

8.1.1 Core Entities

* **Users**: Authentication and profile management
* **Trucks**: Fleet vehicle information and specifications
* **Loads**: Freight load details and tracking
* **Drivers**: Driver information and assignments
* **HOS Logs**: Hours of Service compliance tracking

8.1.2 Analytics Entities

* **Fleet Metrics**: Operational performance data
* **Load Calculations**: Profitability analysis results
* **User Analytics**: System usage tracking (privacy-controlled)
* **Session Management**: User session tracking and security

8.1.3 Integration Entities

* **Load Board Data**: External market opportunities
* **ELD Integration**: Electronic logging device connections
* **System Configuration**: Application settings and preferences

8.2 Data Integrity

* **REQ-DATA-001**: Database shall enforce referential integrity
* **REQ-DATA-002**: Data validation shall occur at application and database levels
* **REQ-DATA-003**: Audit trails shall track all data modifications
* **REQ-DATA-004**: Backup and recovery procedures shall be automated

9. Integration Requirements

9.1 Load Board Integrations

9.1.1 DAT Integration

* **REQ-INT-001**: System shall connect to DAT Power API
* **REQ-INT-002**: System shall sync load opportunities in real-time
* **REQ-INT-003**: System shall handle DAT rate limiting and quotas

9.1.2 Multi-Provider Support

* **REQ-INT-004**: System shall support simultaneous connections to multiple load boards
* **REQ-INT-005**: System shall normalize data formats across providers
* **REQ-INT-006**: System shall handle provider-specific authentication methods

9.2 ELD Provider Integrations

9.2.1 Samsara Integration

* **REQ-INT-007**: System shall connect to Samsara Fleet API
* **REQ-INT-008**: System shall retrieve HOS data and vehicle locations
* **REQ-INT-009**: System shall handle Samsara webhook notifications

9.2.2 Multi-ELD Support

* **REQ-INT-010**: System shall support major ELD providers simultaneously
* **REQ-INT-011**: System shall provide unified interface for HOS data
* **REQ-INT-012**: System shall handle provider-specific data formats

10. Deployment Requirements

10.1 Production Environment

* **REQ-DEPLOY-001**: System shall deploy to Render.com cloud platform
* **REQ-DEPLOY-002**: System shall use US East region for optimal performance
* **REQ-DEPLOY-003**: System shall implement automated deployment pipelines
* **REQ-DEPLOY-004**: System shall support blue-green deployment strategies

10.2 Environment Configuration

* **REQ-DEPLOY-005**: System shall support multiple environments (dev, staging, production)
* **REQ-DEPLOY-006**: Environment variables shall be securely managed
* **REQ-DEPLOY-007**: System shall implement health checks and monitoring
* **REQ-DEPLOY-008**: System shall support horizontal scaling capabilities

10.3 Monitoring and Logging

* **REQ-MONITOR-001**: System shall implement comprehensive application logging
* **REQ-MONITOR-002**: System shall provide real-time performance monitoring
* **REQ-MONITOR-003**: System shall alert on critical system issues
* **REQ-MONITOR-004**: System shall maintain audit logs for compliance

Appendix A: Technical Architecture

System Architecture Overview

┌─────────────────┐ ┌─────────────────┐ ┌─────────────────┐

│ React Client │────│ Express.js API │────│ PostgreSQL │

│ (Frontend) │ │ (Backend) │ │ (Database) │

└─────────────────┘ └─────────────────┘ └─────────────────┘

│ │ │

│ ┌─────────────────┐ │

└──────────────│ External APIs │──────────────┘

│ (Load Boards, │

│ ELD Providers) │

└─────────────────┘

Technology Stack

* **Frontend**: React 18.3.1, TypeScript, Tailwind CSS, React Query
* **Backend**: Node.js 20.19.3, Express.js, TypeScript
* **Database**: PostgreSQL with Drizzle ORM
* **Deployment**: Render.com cloud platform
* **Security**: OpenID Connect, HTTPS/TLS, Role-based access control

Appendix B: Compliance Requirements

Industry Standards

* FMCSA Hours of Service Regulations compliance
* DOT Commercial Vehicle Safety Standards
* Industry cost accounting best practices
* Data privacy and security standards

Regulatory Compliance

* Electronic Logging Device (ELD) mandate compliance
* Commercial Driver's License (CDL) tracking requirements
* Vehicle inspection and maintenance record keeping
* Load documentation and audit trail requirements

**Document Control:**

* **Version**: 1.0
* **Last Updated**: August 12, 2025
* **Approved By**: Development Team
* **Next Review**: September 12, 2025

This Software Requirements Specification defines the complete functional and non-functional requirements for the Travectio Fleet Management System, providing a comprehensive foundation for development, testing, and deployment activities.