**Software Requirements Specification (SRS)**

**Fleet Tracking and Delivery Management System**

**Document Information**

* **Version: 1.0**
* **Date: June 16, 2025**
* **Author: Abdurazzok Tursunov**
* **Course: CSE 360 - Introduction to Software Engineering**
* **Instructor: Dr. Rajan**
* **Project: Fleet Tracking and Delivery Management System**

**Table of Contents**

1. [**Introduction**](https://claude.ai/chat/c196847b-8937-4446-9d41-ab516881aa8f#1-introduction)
2. [**Overall Description**](https://claude.ai/chat/c196847b-8937-4446-9d41-ab516881aa8f#2-overall-description)
3. [**Functional Requirements**](https://claude.ai/chat/c196847b-8937-4446-9d41-ab516881aa8f#3-functional-requirements)
4. [**Non-Functional Requirements**](https://claude.ai/chat/c196847b-8937-4446-9d41-ab516881aa8f#4-non-functional-requirements)
5. [**External Interface Requirements**](https://claude.ai/chat/c196847b-8937-4446-9d41-ab516881aa8f#5-external-interface-requirements)
6. [**System Features**](https://claude.ai/chat/c196847b-8937-4446-9d41-ab516881aa8f#6-system-features)
7. [**Data Requirements**](https://claude.ai/chat/c196847b-8937-4446-9d41-ab516881aa8f#7-data-requirements)
8. [**Assumptions and Dependencies**](https://claude.ai/chat/c196847b-8937-4446-9d41-ab516881aa8f#8-assumptions-and-dependencies)
9. [**Acceptance Criteria**](https://claude.ai/chat/c196847b-8937-4446-9d41-ab516881aa8f#9-acceptance-criteria)

**1. Introduction**

**1.1 Purpose**

**This document specifies the complete software requirements for a real-time fleet tracking and delivery management system. The system integrates with Motive's fleet management API to provide live vehicle monitoring, delivery appointment management, and intelligent late-delivery risk assessment for logistics companies.**

**1.2 Scope**

**The Fleet Tracking System enables logistics companies to:**

* **Monitor vehicle locations and status in real-time**
* **Manage delivery appointments with natural language input**
* **Calculate travel distances and estimated arrival times**
* **Assess delivery risk and identify potential delays proactively**
* **Track load numbers and delivery completion status**
* **Generate fleet performance statistics and insights**

**1.3 Definitions, Acronyms, and Abbreviations**

* **API: Application Programming Interface**
* **ETA: Estimated Time of Arrival**
* **GPS: Global Positioning System**
* **REST: Representational State Transfer**
* **SPA: Single Page Application**
* **UI: User Interface**
* **UX: User Experience**
* **SDLC: Software Development Life Cycle**
* **Motive: Fleet management platform providing vehicle tracking services**
* **Mapbox: Geographic information system providing mapping and routing services**

**1.4 References**

* **Motive API Documentation: https://developers.motive.com**
* **Mapbox API Documentation: https://docs.mapbox.com**
* **React.js Documentation: https://reactjs.org/docs**
* **IEEE Std 830-1998: IEEE Recommended Practice for Software Requirements Specifications**

**1.5 Overview**

**This SRS document is organized into nine main sections covering introduction, system description, functional requirements, non-functional requirements, interfaces, features, data requirements, dependencies, and acceptance criteria. The document follows IEEE 830-1998 standards for software requirements specification.**

**2. Overall Description**

**2.1 Product Perspective**

**The Fleet Tracking System is a web-based application that serves as a centralized dashboard for fleet operations. It integrates with multiple external services to provide comprehensive fleet visibility and operates as a standalone system that enhances existing fleet management workflows.**

**System Context:**

* **Frontend: React-based single-page application**
* **Backend Services: Integration with Motive and Mapbox APIs**
* **Data Storage: Browser-based local storage for user preferences and appointments**
* **Deployment: Web-based application accessible via modern browsers**

**2.2 Product Functions**

**The system provides the following major functions:**

* **Real-Time Vehicle Tracking: Live GPS location monitoring via Motive API**
* **Delivery Appointment Scheduling: Natural language appointment input and management**
* **Distance Calculation: Driving distance and ETA calculation using Mapbox API**
* **Risk Assessment: Intelligent analysis of delivery delay probability**
* **Load Number Tracking: Assignment and tracking of delivery load numbers**
* **Visual Status Filtering: Filter and sort vehicles by various status criteria**
* **Fleet Statistics: Real-time dashboard with key performance indicators**

**2.3 User Classes and Characteristics**

**Primary Users:**

1. **Fleet Managers**
   * **Technical Expertise: Moderate computer skills**
   * **Responsibilities: Overall fleet oversight, performance monitoring**
   * **Usage Frequency: Daily, 4-8 hours**
   * **Key Needs: High-level fleet visibility, exception reporting**
2. **Dispatchers**
   * **Technical Expertise: Basic to moderate computer skills**
   * **Responsibilities: Route planning, driver communication, load assignment**
   * **Usage Frequency: Daily, 6-8 hours**
   * **Key Needs: Real-time vehicle status, appointment management**
3. **Operations Staff**
   * **Technical Expertise: Basic computer skills**
   * **Responsibilities: Monitoring specific routes, customer communication**
   * **Usage Frequency: Daily, 2-4 hours**
   * **Key Needs: Simple interface, clear status indicators**

**2.4 Operating Environment**

**Client Environment:**

* **Browsers: Chrome 90+, Firefox 88+, Safari 14+, Edge 90+**
* **Operating Systems: Windows 10+, macOS 10.14+, iOS 13+, Android 8+**
* **Screen Resolution: Minimum 1024x768, optimized for 1920x1080**
* **Internet Connection: Broadband connection required for real-time updates**

**Development Environment:**

* **Runtime: Node.js 16+**
* **Framework: React 18+ with TypeScript**
* **Styling: Tailwind CSS 3+**
* **Build Tool: Vite 4+**

**2.5 Design and Implementation Constraints**

**Technical Constraints:**

* **Must use Motive API for vehicle data (rate limit: 1000 requests/hour)**
* **Must use Mapbox API for geocoding (free tier: 50,000 requests/month)**
* **Browser localStorage capacity limit (~5-10MB)**
* **Client-side only - no backend database**

**Business Constraints:**

* **Development timeline: 2-3 weeks**
* **Budget: Utilize free API tiers only**
* **Maintenance: Self-contained system requiring minimal ongoing support**

**Regulatory Constraints:**

* **Must comply with data privacy best practices**
* **No storage of sensitive driver or customer information**

**2.6 User Documentation**

**The system will include:**

* **User Manual: Complete guide for all user types**
* **Quick Start Guide: Getting started in under 15 minutes**
* **API Integration Guide: Documentation for system administrators**
* **Troubleshooting Guide: Common issues and solutions**

**2.7 Assumptions and Dependencies**

**Listed in detail in Section 8.**

**3. Functional Requirements**

**3.1 Vehicle Tracking Requirements**

**REQ-001: Real-Time Vehicle Location Display**

**Description: The system shall display current GPS locations of all fleet vehicles in real-time. Priority: High Source: Fleet Manager requirement for live vehicle visibility**

**Inputs:**

* **Motive API vehicle location data**
* **GPS coordinates (latitude, longitude)**
* **Vehicle identification information**

**Processing:**

* **Parse API responses from multiple Motive endpoints**
* **Handle coordinate format variations (lat/lng vs lat/lon)**
* **Validate coordinate ranges and data quality**
* **Update display every 60 seconds automatically**

**Outputs:**

* **Vehicle location table with GPS coordinates**
* **Vehicle status indicators (Moving, Idle, Stationary, Offline)**
* **Current speed and direction information**
* **Last update timestamps in Eastern Time**

**Error Handling:**

* **Graceful degradation when API is unavailable**
* **Display cached data when real-time data fails**
* **Clear error messages for invalid coordinates**

**REQ-002: Vehicle Status Classification**

**Description: The system shall automatically classify vehicle status based on GPS and speed data. Priority: High**

**Status Categories:**

* **Moving: Speed > 5 mph for 2+ consecutive minutes**
* **Idle: Speed 1-5 mph (loading/unloading activities)**
* **Stationary: Speed = 0 mph for 5+ consecutive minutes**
* **Offline: No GPS data received for 10+ minutes**

**Processing Logic:**

* **Implement state machine for status transitions**
* **Apply hysteresis to prevent rapid status changes**
* **Consider speed trends over time windows**

**3.2 Appointment Management Requirements**

**REQ-003: Natural Language Appointment Input**

**Description: The system shall accept delivery appointments in natural language format. Priority: High Source: Dispatcher requirement for ease of use**

**Supported Input Formats:**

* **Standard dates: "06/16/2025 9:00 AM"**
* **Date with time range: "06/16/2025 9AM-2PM"**
* **Relative dates: "Tomorrow 10AM", "Monday 2PM"**
* **Time ranges: "Today 1PM-4PM"**
* **Combined format: "Next Friday 9AM at Walmart Distribution Center"**

**Processing Requirements:**

* **Parse date, time, and location components**
* **Convert all times to Eastern Time Zone**
* **Validate parsed components for logical consistency**
* **Provide format examples for invalid inputs**

**Error Handling:**

* **Display format examples for parsing failures**
* **Highlight invalid date/time components**
* **Suggest corrections for common mistakes**

**REQ-004: Multiple Appointment Support**

**Description: The system shall support multiple delivery appointments per vehicle. Priority: Medium**

**Functionality:**

* **Allow 2-5 appointments per vehicle simultaneously**
* **Display next appointment prominently**
* **Show appointment count and status summary**
* **Enable easy addition and removal of appointments**

**3.3 Distance Calculation Requirements**

**REQ-005: Driving Distance Calculation**

**Description: The system shall calculate accurate driving distances to delivery locations. Priority: High Source: Operations requirement for delivery planning**

**Inputs:**

* **Current vehicle GPS coordinates**
* **Destination address from appointments**
* **Route preferences (fastest, shortest)**

**Processing:**

* **Geocode destination addresses using Mapbox API**
* **Calculate driving routes and distances**
* **Estimate travel time considering traffic patterns**
* **Cache geocoding results for 2 hours**

**Outputs:**

* **Distance in miles with one decimal precision**
* **Estimated travel time in hours and minutes**
* **Estimated time of arrival (ETA) in Eastern Time**

**Performance Requirements:**

* **Complete distance calculations within 30 seconds**
* **Batch process multiple vehicles efficiently**
* **Implement 30-minute auto-refresh cycle**

**REQ-006: API Usage Optimization**

**Description: The system shall optimize API usage to stay within free tier limits. Priority: High**

**Optimization Strategies:**

* **Implement 30-minute automatic calculation intervals**
* **Cache geocoding results for frequently used addresses**
* **Provide manual "Calculate Now" override option**
* **Track and display API usage statistics**

**Usage Monitoring:**

* **Count API calls per day/month**
* **Display usage warnings at 80% of limits**
* **Gracefully degrade when limits approached**

**3.4 Risk Assessment Requirements**

**REQ-007: Delivery Risk Analysis**

**Description: The system shall analyze and classify delivery risk status for each vehicle. Priority: High Source: Fleet Manager requirement for proactive management**

**Risk Categories:**

* **🔴 Late: Current time > appointment time**
* **🟡 At Risk: ETA + buffer time > appointment time**
* **🟢 On Time: ETA + buffer time < appointment time**
* **⚪ No Data: Missing appointment or location information**

**Analysis Algorithm:**

**Risk Status = f(Current Time, Vehicle Location, Appointment Time, Traffic Buffer)**

**Buffer Time = 30 minutes (configurable)**

**ETA = Travel Time + Current Time + Buffer Time**

**Processing Requirements:**

* **Perform risk analysis every 60 seconds**
* **Consider time ranges for appointment windows**
* **Account for traffic and loading time buffers**
* **Update statistics and visual indicators**

**3.5 Load Management Requirements**

**REQ-008: Load Number Tracking**

**Description: The system shall track load numbers assigned to vehicles. Priority: Low**

**Functionality:**

* **Inline editable load number field**
* **Persistent storage across browser sessions**
* **Load assignment statistics**
* **Quick visual indicators for assigned vs unassigned loads**

**4. Non-Functional Requirements**

**4.1 Performance Requirements**

**PERF-001: Response Time**

**Requirement: The system shall respond to user interactions within specified time limits. Measurements:**

* **Initial page load: ≤ 3 seconds**
* **Vehicle data refresh: ≤ 2 seconds**
* **Distance calculations: ≤ 30 seconds**
* **Appointment input processing: ≤ 1 second**

**PERF-002: Throughput**

**Requirement: The system shall handle expected user loads efficiently. Measurements:**

* **Support 58 vehicles simultaneously**
* **Process 10 appointment additions per minute**
* **Handle 5 concurrent users without degradation**

**PERF-003: Resource Usage**

**Requirement: The system shall operate within browser resource constraints. Measurements:**

* **Memory usage: ≤ 100MB after 8-hour session**
* **Local storage: ≤ 5MB total data**
* **CPU usage: ≤ 10% during idle periods**

**4.2 Reliability Requirements**

**REL-001: Availability**

**Requirement: The system shall maintain high availability during business hours. Target: 99% uptime during 6 AM - 8 PM Eastern Time**

**REL-002: Error Recovery**

**Requirement: The system shall recover gracefully from failures. Specifications:**

* **Automatic retry for failed API calls**
* **Fallback to cached data when APIs unavailable**
* **Clear error messages with recovery suggestions**
* **No data loss during temporary network issues**

**REL-003: Data Persistence**

**Requirement: User-entered data shall persist across browser sessions. Scope: Appointments, load numbers, user preferences**

**4.3 Usability Requirements**

**USE-001: Learning Curve**

**Requirement: New users shall become productive quickly. Target: 95% of users can add appointments successfully within 15 minutes**

**USE-002: Interface Design**

**Requirement: The interface shall be intuitive and professional. Specifications:**

* **Responsive design for desktop and tablet**
* **Color-blind friendly status indicators**
* **Consistent navigation and layout**
* **Professional color scheme and typography**

**USE-003: Accessibility**

**Requirement: The system shall be accessible to users with disabilities. Standards: WCAG 2.1 Level AA compliance where feasible**

**4.4 Security Requirements**

**SEC-001: Data Protection**

**Requirement: The system shall protect sensitive information. Specifications:**

* **API keys secured in environment variables**
* **No transmission of sensitive driver information**
* **HTTPS-only API communications**
* **No persistent storage of confidential data**

**SEC-002: Input Validation**

**Requirement: All user inputs shall be validated and sanitized. Scope: Appointment text, load numbers, filter selections**

**4.5 Compatibility Requirements**

**COMP-001: Browser Support**

**Requirement: The system shall work on major modern browsers. Supported Browsers:**

* **Chrome 90+ (primary development target)**
* **Firefox 88+**
* **Safari 14+**
* **Edge 90+**

**COMP-002: Device Support**

**Requirement: The system shall provide responsive design. Target Devices:**

* **Desktop: 1920x1080+ (primary)**
* **Tablet: 1024x768+ (secondary)**
* **Mobile: 375x667+ (limited support)**

**5. External Interface Requirements**

**5.1 User Interfaces**

**UI-001: Main Dashboard**

**Description: Primary interface showing fleet overview and statistics. Components:**

* **Statistics cards showing fleet metrics**
* **Vehicle table with real-time data**
* **Filter controls for status-based views**
* **Refresh indicators and connection status**

**Layout Requirements:**

* **Header with system title and refresh controls**
* **Statistics grid (2x4 cards on desktop)**
* **Filterable vehicle table with sortable columns**
* **Footer with last update timestamp**

**UI-002: Vehicle Management Interface**

**Description: Detailed vehicle information and controls. Components:**

* **Inline editable appointment fields**
* **Load number assignment controls**
* **Risk status indicators**
* **Historical appointment list**

**UI-003: Mobile-Responsive Design**

**Description: Adapted interface for tablet and mobile devices. Adaptations:**

* **Collapsible navigation menu**
* **Stacked statistics cards**
* **Horizontal scrolling for wide tables**
* **Touch-friendly button sizes**

**5.2 Hardware Interfaces**

**HW-001: GPS Tracking Devices**

**Description: Integration with Motive-compatible vehicle tracking hardware. Requirements:**

* **Compatible with Motive OBD-II devices**
* **Support for cellular and GPS connectivity**
* **Real-time location reporting capability**

**HW-002: User Devices**

**Description: Client hardware requirements for optimal performance. Minimum Specifications:**

* **Processor: Dual-core 1.5GHz or equivalent**
* **Memory: 4GB RAM**
* **Storage: 100MB available browser cache**
* **Network: Broadband internet connection**

**5.3 Software Interfaces**

**SW-001: Motive Fleet Management API**

**Description: Primary data source for vehicle information. Interface Specifications:**

* **Protocol: HTTPS REST API**
* **Authentication: API key-based**
* **Endpoints:** 
  + **/vehicles - Vehicle list and basic information**
  + **/vehicles/{id}/locations - Real-time location data**
  + **/vehicles/{id}/status - Vehicle status information**
* **Data Format: JSON**
* **Rate Limits: 1000 requests per hour**
* **Error Handling: HTTP status codes with JSON error messages**

**Data Exchange:**

**{**

**"vehicles": [**

**{**

**"id": "vehicle\_12345",**

**"name": "Truck 15",**

**"location": {**

**"latitude": 40.712776,**

**"longitude": -74.005974**

**},**

**"status": "moving",**

**"speed": 45.2,**

**"last\_seen": "2025-06-15T14:30:15Z"**

**}**

**]**

**}**

**SW-002: Mapbox Geocoding and Routing API**

**Description: Geographic services for address resolution and route calculation. Interface Specifications:**

* **Protocol: HTTPS REST API**
* **Authentication: Access token-based**
* **Endpoints:** 
  + **/geocoding/v5/mapbox.places/{address} - Address geocoding**
  + **/directions/v5/mapbox/driving/{coordinates} - Route calculation**
* **Data Format: JSON**
* **Rate Limits: 50,000 requests per month (free tier)**

**SW-003: Browser Local Storage**

**Description: Client-side data persistence for user preferences and appointments. Interface Specifications:**

* **API: HTML5 localStorage interface**
* **Capacity: ~5-10MB depending on browser**
* **Data Format: JSON strings**
* **Persistence: Survives browser restarts, cleared manually or by browser cleanup**

**5.4 Communication Interfaces**

**COMM-001: Network Protocols**

**Requirements:**

* **Primary Protocol: HTTPS for all external API communications**
* **Fallback: HTTP not supported for security reasons**
* **WebSocket: Future enhancement for real-time updates**

**COMM-002: Data Formats**

**Standards:**

* **API Communication: JSON (RFC 7159)**
* **Date/Time: ISO 8601 format with timezone information**
* **Coordinates: Decimal degrees format (WGS84)**

**6. System Features**

**6.1 Real-Time Fleet Dashboard**

**6.1.1 Description**

**Central command center providing comprehensive fleet visibility with live updates and key performance indicators.**

**6.1.2 Functional Requirements**

* **REQ-001: Vehicle location display**
* **REQ-002: Vehicle status classification**
* **REQ-007: Risk assessment display**

**6.1.3 Priority**

**High - Core system functionality**

**6.1.4 Inputs**

* **Motive API vehicle data**
* **User filter selections**
* **Refresh interval preferences**

**6.1.5 Processing**

* **Aggregate vehicle statistics**
* **Apply status-based filtering**
* **Calculate fleet-wide metrics**
* **Update displays in real-time**

**6.1.6 Outputs**

* **Statistics cards with fleet metrics**
* **Vehicle table with current status**
* **Visual status indicators**
* **Connection and update status**

**6.2 Smart Appointment Management**

**6.2.1 Description**

**Natural language appointment scheduling system with intelligent parsing and validation.**

**6.2.2 Functional Requirements**

* **REQ-003: Natural language input processing**
* **REQ-004: Multiple appointment support**

**6.2.3 Priority**

**High - Key differentiating feature**

**6.2.4 Processing Logic**

1. **Parse user input using natural language processing**
2. **Extract date, time, and location components**
3. **Validate parsed components for consistency**
4. **Convert to standard format and timezone**
5. **Store in local storage with vehicle association**

**6.3 Intelligent Risk Assessment**

**6.3.1 Description**

**Proactive analysis system for identifying potential delivery delays before they occur.**

**6.3.2 Functional Requirements**

* **REQ-005: Distance calculation**
* **REQ-007: Risk analysis algorithm**

**6.3.3 Priority**

**High - Business value feature**

**6.3.4 Risk Calculation Algorithm**

**function calculateRisk(vehicle, appointment, currentTime) {**

**const travelTime = calculateDrivingTime(vehicle.location, appointment.destination);**

**const bufferTime = 30; // minutes**

**const eta = currentTime + travelTime + bufferTime;**

**if (currentTime > appointment.time) {**

**return "Late";**

**} else if (eta > appointment.time) {**

**return "At Risk";**

**} else {**

**return "On Time";**

**}**

**}**

**7. Data Requirements**

**7.1 Data Entities**

**7.1.1 Vehicle Entity**

**Description: Represents a fleet vehicle and its current state.**

**Attributes:**

* **vehicle\_id (string, primary key): Unique vehicle identifier**
* **name (string): Human-readable vehicle name**
* **latitude (float): Current GPS latitude (-90 to 90)**
* **longitude (float): Current GPS longitude (-180 to 180)**
* **status (enum): Current vehicle status (moving, idle, stationary, offline)**
* **speed (float): Current speed in mph**
* **driver\_id (string): Assigned driver identifier**
* **last\_update (datetime): Timestamp of last data update**

**Constraints:**

* **Coordinates must be valid GPS ranges**
* **Status must be one of defined enumeration values**
* **Speed must be non-negative**

**7.1.2 Appointment Entity**

**Description: Represents a delivery appointment for a specific vehicle.**

**Attributes:**

* **appointment\_id (string, primary key): Unique appointment identifier**
* **vehicle\_id (string, foreign key): Associated vehicle**
* **address (string): Delivery destination address**
* **appointment\_datetime (datetime): Scheduled appointment time**
* **time\_range\_start (datetime, optional): Window start time**
* **time\_range\_end (datetime, optional): Window end time**
* **status (enum): Appointment status (pending, completed, missed)**
* **coordinates (object): Geocoded destination coordinates**
* **created\_at (datetime): Appointment creation timestamp**

**7.1.3 Load Assignment Entity**

**Description: Tracks load numbers assigned to vehicles.**

**Attributes:**

* **vehicle\_id (string, primary key): Associated vehicle**
* **load\_number (string): Assigned load identifier**
* **assigned\_at (datetime): Assignment timestamp**
* **status (enum): Load status (assigned, in\_transit, delivered)**

**7.2 Data Storage**

**7.2.1 Local Storage Strategy**

**Implementation: HTML5 localStorage for client-side persistence**

**Stored Data:**

* **User-entered appointments**
* **Load number assignments**
* **User preferences and settings**
* **Geocoding cache**

**Data Organization:**

**{**

**"appointments": {**

**"vehicle\_123": [**

**{**

**"id": "appt\_001",**

**"address": "Walmart Distribution Center",**

**"datetime": "2025-06-16T14:00:00-05:00",**

**"status": "pending"**

**}**

**]**

**},**

**"loadNumbers": {**

**"vehicle\_123": "LOAD\_456789"**

**},**

**"preferences": {**

**"refreshInterval": 60,**

**"defaultView": "all"**

**}**

**}**

**7.2.2 Cache Management**

**Geocoding Cache:**

* **Purpose: Reduce API calls for repeated addresses**
* **Duration: 2 hours time-to-live**
* **Size Limit: 100 most recent addresses**
* **Invalidation: Automatic expiration and manual clear option**

**7.3 Data Validation**

**7.3.1 Input Validation Rules**

**Appointment Data:**

* **Date must be current date or future**
* **Time must be valid 24-hour format**
* **Address must be non-empty string**
* **Time ranges must have start < end**

**Vehicle Data:**

* **Coordinates must be within valid GPS ranges**
* **Speed must be non-negative numeric value**
* **Status must match predefined enumeration**

**7.3.2 Data Integrity**

**Consistency Checks:**

* **Appointment times logical relative to current time**
* **Vehicle associations valid and existing**
* **Load numbers unique per assignment period**

**8. Assumptions and Dependencies**

**8.1 Assumptions**

**8.1.1 User Environment Assumptions**

* **Users have reliable broadband internet connectivity**
* **Users operate modern web browsers with JavaScript enabled**
* **Users have basic computer literacy for web applications**
* **Operating environment is primarily desktop/tablet, not mobile-first**

**8.1.2 Data Quality Assumptions**

* **Motive API provides accurate, up-to-date vehicle location data**
* **GPS tracking devices are properly installed and functional**
* **Vehicle status information is reliable and consistently formatted**
* **Network connectivity allows for regular API communication**

**8.1.3 Business Process Assumptions**

* **Delivery appointments are scheduled in advance, not ad-hoc**
* **Fleet operations occur primarily during standard business hours**
* **Load assignments follow existing company procedures**
* **Risk assessment enhances rather than replaces human judgment**

**8.2 Dependencies**

**8.2.1 External Service Dependencies**

**Motive Fleet Management Platform:**

* **Availability: 99.9% uptime during business hours**
* **API Stability: Consistent endpoint behavior and data formats**
* **Rate Limits: Sufficient quota for real-time monitoring**
* **Data Quality: Accurate GPS coordinates and vehicle status**

**Mapbox Geographic Services:**

* **Geocoding Service: Address-to-coordinate conversion**
* **Routing Service: Driving distance and time calculations**
* **Rate Limits: Free tier sufficient for expected usage (50K requests/month)**
* **Data Accuracy: Reliable routing and traffic consideration**

**8.2.2 Technology Dependencies**

**Browser Compatibility:**

* **Modern JavaScript: ES6+ feature support**
* **Local Storage: HTML5 localStorage availability**
* **Network APIs: Fetch API and Promise support**
* **CSS Grid/Flexbox: Modern layout capabilities**

**Development Framework Dependencies:**

* **React 18+: Component framework and hooks**
* **TypeScript: Type safety and development tooling**
* **Tailwind CSS: Utility-first styling framework**
* **Vite: Build tool and development server**

**8.2.3 Infrastructure Dependencies**

**Development Environment:**

* **Node.js 16+: Runtime environment for development tools**
* **Git: Version control system**
* **GitHub: Repository hosting and collaboration**

**Deployment Environment:**

* **Web Server: Static file serving capability**
* **HTTPS Support: Secure communication for API calls**
* **Domain/Hosting: Accessible URL for user access**

**8.3 Risk Mitigation**

**8.3.1 API Dependency Risks**

**Risk: Motive API becomes unavailable Mitigation:**

* **Implement graceful degradation to cached data**
* **Provide clear user notification of data staleness**
* **Design fallback displays for offline scenarios**

**Risk: Mapbox usage exceeds free tier limits Mitigation:**

* **Implement usage monitoring and warnings**
* **Cache geocoding results aggressively**
* **Provide manual distance calculation override**

**8.3.2 Technical Risks**

**Risk: Browser compatibility issues Mitigation:**

* **Test on multiple browsers during development**
* **Use progressive enhancement for advanced features**
* **Provide clear browser requirement documentation**

**Risk: Local storage capacity limits Mitigation:**

* **Implement data cleanup for old appointments**
* **Monitor storage usage and warn users**
* **Design for graceful degradation if storage full**

**9. Acceptance Criteria**

**9.1 Functional Acceptance Criteria**

**9.1.1 Vehicle Tracking Acceptance**

* **ACCEPT-001: All 58 fleet vehicles display current location within 60 seconds of system load**
* **ACCEPT-002: Vehicle status updates automatically every 60 seconds without user intervention**
* **ACCEPT-003: Vehicle status accurately reflects movement state (moving, idle, stationary, offline)**
* **ACCEPT-004: GPS coordinates are displayed with appropriate precision and validation**
* **ACCEPT-005: System gracefully handles vehicles with missing or invalid GPS data**

**9.1.2 Appointment Management Acceptance**

* **ACCEPT-006: Natural language appointment input parses correctly for standard formats:** 
  + **"06/16/2025 9:00 AM" ✓**
  + **"Tomorrow 2PM" ✓**
  + **"Monday 9AM-2PM" ✓**
  + **"Next Friday at Walmart" ✓**
* **ACCEPT-007: Invalid appointment formats display helpful error messages with examples**
* **ACCEPT-008: Multiple appointments per vehicle are supported (minimum 3 appointments)**

**ACCEPT-009: Appointments persist across browser sessions and page refreshes**

* **ACCEPT-010: Next appointment is prominently displayed for each vehicle**

**9.1.3 Distance Calculation Acceptance**

* **ACCEPT-011: Distance calculations complete within 30 seconds for all vehicles with appointments**
* **ACCEPT-012: Driving distances are accurate within 10% of actual routes**
* **ACCEPT-013: ETAs account for current time and include reasonable buffer**
* **ACCEPT-014: Geocoding results are cached to minimize API usage**
* **ACCEPT-015: Manual "Calculate Distances" button provides immediate updates**

**9.1.4 Risk Assessment Acceptance**

* **ACCEPT-016: Risk status accurately categorizes deliveries as Late, At Risk, or On Time**
* **ACCEPT-017: Risk assessment updates automatically when vehicle positions change**
* **ACCEPT-018: Filter functionality correctly displays vehicles by risk status**
* **ACCEPT-019: Risk indicators are visually distinct and color-blind accessible**
* **ACCEPT-020: Risk calculations include appropriate buffer time (30 minutes default)**

**9.2 Performance Acceptance Criteria**

**9.2.1 Response Time Acceptance**

* **PERF-ACCEPT-001: Initial dashboard loads completely within 3 seconds on broadband connection**
* **PERF-ACCEPT-002: Vehicle data refresh completes within 2 seconds**
* **PERF-ACCEPT-003: Appointment input processing responds within 1 second**
* **PERF-ACCEPT-004: Filter operations complete within 500 milliseconds**
* **PERF-ACCEPT-005: Distance calculations for 10 vehicles complete within 30 seconds**

**9.2.2 Resource Usage Acceptance**

* **PERF-ACCEPT-006: Memory usage remains under 100MB after 8-hour continuous session**
* **PERF-ACCEPT-007: Local storage usage stays under 5MB total**
* **PERF-ACCEPT-008: CPU usage below 10% during idle periods**
* **PERF-ACCEPT-009: No memory leaks detected during extended usage testing**

**9.3 Usability Acceptance Criteria**

**9.3.1 User Experience Acceptance**

* **UX-ACCEPT-001: 95% of test users successfully add appointment within 2 minutes of first use**
* **UX-ACCEPT-002: Interface adapts responsively to tablet screen sizes (1024x768+)**
* **UX-ACCEPT-003: All interactive elements have appropriate hover states and feedback**
* **UX-ACCEPT-004: Error messages are clear and provide actionable guidance**
* **UX-ACCEPT-005: Loading states are indicated for operations taking >1 second**

**9.3.2 Accessibility Acceptance**

* **ACCESS-001: Color-blind users can distinguish all status indicators**
* **ACCESS-002: Keyboard navigation works for all interactive elements**
* **ACCESS-003: Screen reader compatibility for major interface elements**
* **ACCESS-004: Minimum contrast ratios meet WCAG 2.1 guidelines**

**9.4 Technical Acceptance Criteria**

**9.4.1 Browser Compatibility Acceptance**

* **TECH-ACCEPT-001: Full functionality in Chrome 90+ (primary target)**
* **TECH-ACCEPT-002: Core functionality in Firefox 88+, Safari 14+, Edge 90+**
* **TECH-ACCEPT-003: Graceful degradation in unsupported browsers**
* **TECH-ACCEPT-004: No JavaScript errors in browser console during normal operation**

**9.4.2 API Integration Acceptance**

* **TECH-ACCEPT-005: Motive API integration successfully retrieves vehicle data**
* **TECH-ACCEPT-006: Mapbox API integration provides accurate geocoding and routing**
* **TECH-ACCEPT-007: API error scenarios handled gracefully with user notification**
* **TECH-ACCEPT-008: API usage stays within free tier limits with monitoring alerts**
* **TECH-ACCEPT-009: Network failures result in fallback to cached data, not system crash**

**9.4.3 Data Persistence Acceptance**

* **DATA-ACCEPT-001: User appointments persist across browser restarts**
* **DATA-ACCEPT-002: Load number assignments retained between sessions**
* **DATA-ACCEPT-003: User preferences saved and restored correctly**
* **DATA-ACCEPT-004: Data corruption scenarios handled with clear error messages**
* **DATA-ACCEPT-005: Local storage cleanup prevents unlimited growth**

**9.5 Security Acceptance Criteria**

**9.5.1 Data Protection Acceptance**

* **SEC-ACCEPT-001: API keys not exposed in client-side code or network requests**
* **SEC-ACCEPT-002: All external API communications use HTTPS protocol**
* **SEC-ACCEPT-003: No sensitive driver or customer information stored locally**
* **SEC-ACCEPT-004: Input validation prevents injection attacks**
* **SEC-ACCEPT-005: Error messages do not reveal sensitive system information**

**9.6 Business Acceptance Criteria**

**9.6.1 Operational Value Acceptance**

* **BIZ-ACCEPT-001: System provides proactive identification of late delivery risks**
* **BIZ-ACCEPT-002: Fleet managers can monitor 58 vehicles efficiently from single dashboard**
* **BIZ-ACCEPT-003: Dispatchers can add appointments faster than previous manual methods**
* **BIZ-ACCEPT-004: Risk assessment enables proactive customer communication**
* **BIZ-ACCEPT-005: Real-time visibility improves operational decision-making**

**9.6.2 Cost Effectiveness Acceptance**

* **BIZ-ACCEPT-006: System operates within free API tier limits for expected usage**
* **BIZ-ACCEPT-007: No additional hardware or infrastructure costs required**
* **BIZ-ACCEPT-008: Minimal training required for user adoption**
* **BIZ-ACCEPT-009: Return on investment evident within first month of usage**

**9.7 Acceptance Testing Process**

**9.7.1 Test Environment Setup**

**Prerequisites:**

* **Clean browser environment with cleared cache and storage**
* **Stable internet connection (minimum 10 Mbps)**
* **Access to Motive API test environment**
* **Valid Mapbox API credentials for testing**

**Test Data Requirements:**

* **Minimum 10 active vehicles with valid GPS coordinates**
* **Variety of vehicle statuses (moving, idle, stationary)**
* **Sample delivery addresses for distance calculation testing**
* **Edge cases: invalid coordinates, offline vehicles**

**9.7.2 Acceptance Test Execution**

**Phase 1: Functional Testing**

1. **Execute all functional acceptance criteria**
2. **Document pass/fail status for each criterion**
3. **Capture screenshots for failed tests**
4. **Record performance metrics during testing**

**Phase 2: User Acceptance Testing**

1. **Recruit representatives from each user class**
2. **Provide minimal training (15-minute overview)**
3. **Execute realistic scenarios without assistance**
4. **Collect feedback on usability and functionality**

**Phase 3: Integration Testing**

1. **Test with live API endpoints and real data**
2. **Verify error handling with simulated API failures**
3. **Test edge cases and boundary conditions**
4. **Validate performance under realistic load**

**9.7.3 Acceptance Criteria Evaluation**

**Pass Criteria:**

* **100% of critical functional requirements met**
* **95% of performance requirements within targets**
* **90% of usability requirements satisfied**
* **Zero critical security vulnerabilities**
* **Positive user feedback from acceptance testing**

**Conditional Pass Criteria:**

* **Minor performance issues with mitigation plan**
* **Non-critical usability improvements identified**
* **Documentation updates required**
* **Minor bug fixes needed before deployment**

**Fail Criteria:**

* **Any critical functional requirement not met**
* **Security vulnerabilities identified**
* **Performance significantly below targets**
* **User acceptance testing shows major usability issues**
* **System instability or data loss scenarios**

**9.8 Sign-off Requirements**

**9.8.1 Technical Sign-off**

**Required Approvals:**

* **Development Team: All acceptance criteria verified**
* **Quality Assurance: Testing completed and documented**
* **System Architecture: Design requirements satisfied**
* **Security Review: No significant vulnerabilities identified**

**9.8.2 Business Sign-off**

**Required Approvals:**

* **Fleet Manager Representative: Operational requirements met**
* **Dispatcher Representative: Usability requirements satisfied**
* **Operations Representative: System meets daily workflow needs**
* **Project Sponsor: Business value delivered as specified**

**9.8.3 Deployment Readiness**

**Final Checklist:**

* **All acceptance criteria passed or conditionally passed**
* **User documentation completed and reviewed**
* **Training materials prepared for each user class**
* **Support procedures documented**
* **Deployment plan approved and scheduled**

**Appendices**

**Appendix A: Glossary**

**At Risk: Delivery status indicating potential for late arrival based on current progress and traffic conditions.**

**Buffer Time: Additional time allowance (default 30 minutes) added to travel time calculations to account for loading, traffic, and other delays.**

**ETA (Estimated Time of Arrival): Calculated time when vehicle is expected to reach destination based on current location, route, and traffic conditions.**

**Fleet Manager: Primary user responsible for overall fleet operations, performance monitoring, and strategic decision-making.**

**Geocoding: Process of converting street addresses into geographic coordinates (latitude and longitude).**

**Idle Status: Vehicle state indicating minimal movement (1-5 mph), typically during loading/unloading operations.**

**Late Status: Delivery status indicating appointment time has already passed.**

**Load Number: Unique identifier assigned to delivery shipments for tracking and reference purposes.**

**Moving Status: Vehicle state indicating active travel (speed > 5 mph).**

**Natural Language Input: User interface feature allowing appointment entry in conversational format rather than structured forms.**

**On Time Status: Delivery status indicating sufficient time to reach destination before appointment.**

**Risk Assessment: Algorithmic analysis of delivery completion probability based on current vehicle position, destination, and appointment timing.**

**Stationary Status: Vehicle state indicating no movement (speed = 0) for extended period (5+ minutes).**

**Appendix B: References**

1. **IEEE Std 830-1998: IEEE Recommended Practice for Software Requirements Specifications**
2. **Motive Developer Documentation: https://developers.motive.com/reference**
3. **Mapbox API Documentation: https://docs.mapbox.com/api/**
4. **React.js Official Documentation: https://reactjs.org/docs/getting-started.html**
5. **Web Content Accessibility Guidelines (WCAG) 2.1: https://www.w3.org/WAI/WCAG21/**
6. **RFC 7159: The JavaScript Object Notation (JSON) Data Interchange Format**
7. **ISO 8601: Date and time format standard**

**Appendix C: Change History**

| **Version** | **Date** | **Author** | **Changes** |
| --- | --- | --- | --- |
| **1.0** | **June 16, 2025** | **Abdurazzok Tursunov** | **Initial document creation** |

***This Software Requirements Specification document represents the complete functional and non-functional requirements for the Fleet Tracking and Delivery Management System as developed for CSE 360 Final Exam project. The document follows IEEE 830-1998 standards and provides comprehensive coverage of all system aspects necessary for successful implementation and deployment.***