Vehicle Transport Management System (VTMS)

Software Requirements Specification (SRS)

Prepared by: VTMS Development Team

	Name	Roll No
1	Tahira Hanif	NUM-BSCS-2024-76
2	Momina Umer	NUM-BSCS-2024-36
3	Mubashir Aman	NUM-BSCS-2024-37

Department of Computer Science
Namal University Mianwali

January 8, 2026

Contents

1 Introduction **3**
- 1.1 Purpose . . . 3
- 1.2 Scope . . . 3
  - 1.2.1 Product Identification . . . 3
  - 1.2.2 Product Features . . . 3
- 1.3 Definitions, Acronyms, and Abbreviations . . . 4
- 1.4 References . . . 4
  - 1.4.1 Standards . . . 4
  - 1.4.2 Technical Documentation . . . 4
  - 1.4.3 Development Tools . . . 4
- 1.5 Overview . . . 4

2 Overall Description **5**
- 2.1 Product Perspective . . . 5
- 2.2 Product Functions . . . 5
- 2.3 User Classes and Characteristics . . . 5
- 2.4 Operating Environment . . . 5
- 2.5 Design and Implementation Constraints . . . 6
- 2.6 Assumptions and Dependencies . . . 6
  - 2.6.1 Assumptions . . . 6
  - 2.6.2 Dependencies . . . 6

3 Specific Requirements **6**
- 3.1 External Interface Requirements . . . 6
  - 3.1.1 User Interfaces . . . 6
  - 3.1.2 Hardware Interfaces . . . 7
  - 3.1.3 Software Interfaces . . . 7
  - 3.1.4 Communication Interfaces . . . 7
- 3.2 Functional Requirements . . . 7
  - 3.2.1 FR1: Casual Vehicle Type Selection (Priority: Critical) . . . 7
  - 3.2.2 FR2: Buyer Information Collection (Priority: Critical) . . . 8
  - 3.2.3 FR3: Advance Payment Processing (Priority: Critical) . . . 9
  - 3.2.4 FR4: Shipment Tracking Dashboard (Priority: Critical) . . . 9
  - 3.2.5 FR5: Transport Manager Status Updates (Priority: High) . . . 10
  - 3.2.6 FR6: Final Payment Processing (Priority: Critical) . . . 10
  - 3.2.7 FR7: Delivery Confirmation (Priority: High) . . . 11
  - 3.2.8 FR8: Order Cancellation - 50% Refund Policy (Priority: High) 11
  - 3.2.9 FR9: Automated Notification System (Priority: Critical) . . . 11
  - 3.2.10 FR10: Administrator Functions (Priority: High) . . . 12
  - 3.2.11 FR11: Finance Manager Functions (Priority: Medium) . . . 12
  - 3.2.12 FR12: Vehicle Type Configuration (Priority: Medium) . . . 12
- 3.3 Performance Requirements . . . 13
- 3.4 Design Constraints . . . 13
- 3.5 Software System Attributes . . . 13
  - 3.5.1 Security Requirements . . . 13
  - 3.5.2 Reliability Requirements . . . 14
  - 3.5.3 Availability Requirements . . . 14

3.5.4 Maintainability Requirements 14
3.5.5 Usability Requirements 14
3.6 Other Requirements 15
3.6.1 Database Requirements 15

4 Supporting Information **15**
4.1 System Workflow 15
4.2 Future Enhancements (Post v1.0) 15
4.3 Appendix A: Context Diagram 15
4.4 Appendix B: Usecase Diagram 15

1 Introduction

1.1 Purpose

This Software Requirements Specification defines complete functional and non-functional requirements for the Vehicle Transport Management System (VTMS). VTMS automates vehicle export operations through a streamlined workflow eliminating manual processes that cause data loss, billing errors, and delivery delays.

1.2 Scope

1.2.1 Product Identification

VTMS provides:

- Direct vehicle type selection from casual dropdown (NO browsing/catalog)
- Terms acceptance → Vehicle type → Buyer info → 30% advance payment → Tracking ID
- Real-time shipment tracking with exact export timestamps to last shipment port
- Automated notifications between 4 actors (Vehicle Buyer, Admin, Transport Manager, Finance Manager)
- Strict cancellation policy requiring 3 mandatory checkboxes for 50% advance refund

1.2.2 Product Features

VTMS Will Provide:

- Direct vehicle type selection via dropdown
- Automated workflow from order to delivery
- Payment processing (30% advance, 70% final)
- Shipment tracking with timestamps
- Role-based access control for 4 user types
- Automated notification system
- Cancellation with 50% refund policy

VTMS Will Not Include (Initial Release):

- Physical vehicle inventory management
- Shopping cart functionality
- Vehicle browsing/search capabilities
- Multi-vehicle ordering per session
- Mobile applications (web-only v1.0)

1.3 Definitions, Acronyms, and Abbreviations

Term/Acronym	Definition
VTMS	Vehicle Transport Management System
Tracking ID	Unique identifier: VTMS-YYYYMMDD-XXXX
Advance Payment	30% of total order value
Final Payment	70% balance due after export confirmation
Last Shipment Port	Final export port before ocean transport
RBAC	Role-Based Access Control
SRS	System Requirements Specification

1.4 References

1.4.1 Standards

1. IEEE Std 830-1998: Recommended Practice for Software Requirements Specifications
2. IEEE Std 1002-1987: Software Engineering Standards Taxonomy

1.4.2 Technical Documentation

1. University Software Engineering Project Guidelines

1.4.3 Development Tools

1. LaTeX for documentation
2. GitHub
3. Figma

1.5 Overview

This document follows IEEE Std 830-1998 structure:

- **Section 2** provides high-level system description
- **Section 3** details all functional and non-functional requirements
- **Section 4** includes supporting technical information

2 Overall Description

2.1 Product Perspective

VTMS replaces error-prone manual vehicle export processes with an automated web system ensuring:

- Reliable payment processing (30% advance, 70% final)
- Accurate shipment tracking with timestamps
- Automated notifications between stakeholders
- Strict 50% refund policy enforcement

2.2 Product Functions

Core Workflow:

1. **BUYER:** Select vehicle type → Terms → Info → Pay 30% → Receive Tracking ID
2. **TRANSPORT:** Update "Exported to Last Port" → System notifies buyer for final payment
3. **BUYER:** Pay 70% → TRANSPORT confirms delivery
4. **Optional:** BUYER cancels (50% refund via 3-checkbox policy)

2.3 User Classes and Characteristics

Actor	**Technical Level**	**Primary Functions**
Vehicle Buyer	Basic web users	Order, track, pay 30%/70%, cancel
Administrator	Advanced technical	User/order/vehicle type management
Transport Manager	Intermediate	Status updates, delivery confirmation
Finance Manager	Intermediate	Payment verification, reports

2.4 Operating Environment

- **CLIENT:** Modern browsers (Chrome 90+, Firefox 88+, Safari 15+, Edge 90+)
- **SERVER:** Linux/Windows, Apache/Nginx, PHP/Node.js/Python

- **DATABASE:** PostgreSQL 12+ / MySQL 8.0+
- **NETWORK:** HTTPS/TLS 1.3, stable internet (2Mbps)
- **EXTERNAL APIs:** Payment gateway, SMTP, SMS services

2.5 Design and Implementation Constraints

- Web application only (no native mobile v1.0)
- IEEE 830-1998 document format compliance
- Role-Based Access Control mandatory
- 50% advance refund policy hardcoded
- Payment gateway integration required

2.6 Assumptions and Dependencies

2.6.1 Assumptions

- Users have stable internet connectivity
- Transport managers update status within 24 hours
- Payment gateway uptime 99.9%

2.6.2 Dependencies

- Third-party payment gateway (Stripe/PayPal)
- Email service (SendGrid/Gmail SMTP)
- SMS service (Twilio)

3 Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

- **HOMEPAGE:** Single prominent "Order Vehicle" button
- **TERMS MODAL:** Mandatory "I Agree" checkbox (non-dismissible)
- **VEHICLE SELECTION:** Casual type dropdown + auto-price calculation
- **ORDER FORM:** Buyer details + pre-filled vehicle info
- **PAYMENT:** Secure gateway redirect
- **TRACKING DASHBOARD:** Status timeline + Cancel button (pre-delivery only)
- **ADMIN DASHBOARD:** Role-specific CRUD operations

3.1.2 Hardware Interfaces

- Standard web-enabled devices (desktop/laptop/tablet)
- Printer support for receipts/reports (PDF generation)

3.1.3 Software Interfaces

- **PAYMENT GATEWAY:** REST API (Stripe/PayPal/Razorpay)
- **EMAIL SERVICE:** SMTP protocol (SendGrid/SES)
- **SMS SERVICE:** REST API (Twilio/Nexmo)
- **DATABASE:** JDBC/ODBC compliant relational database

3.1.4 Communication Interfaces

- **PROTOCOL:** HTTPS 1.3 with TLS 1.3 minimum
- **DATA FORMAT:** JSON over RESTful APIs
- **RATE LIMITING:** 100 requests/minute per IP
- **SESSION:** JWT tokens (30-minute expiry)

3.2 Functional Requirements

3.2.1 FR1: Casual Vehicle Type Selection (Priority: Critical)

Introduction: Vehicle buyer selects from admin-configured casual vehicle type dropdown with standard pricing.

Inputs:

- Vehicle Type Dropdown:
  - Sedan (Standard $25,000)
  - SUV (Standard $35,000)
  - Pickup Truck ($45,000)
  - Hatchback ($20,000)
  - Minivan ($30,000)
  - Sports Car ($55,000)
- Additional Fields:
  - Color preference (free text)
  - Model year preference (2020-2026)
  - Quantity (1-3 vehicles)

Processing:

1. Populate dropdown from vehicle_types table

2. On selection → Auto-fill standard price
3. Calculate: total_cost = price × quantity
4. advance_payment = total_cost × 0.3
5. Generate Order ID: VTMS-ORD-YYYYMMDD-XXXX

Outputs:

- Order Summary:
  - Vehicle Type: Sedan ($25,000 standard)
  - Color: White — Quantity: 1 — Total: $25,000
  - 30% Advance Due: $7,500

Terms & Conditions [Proceed to Payment]

3.2.2 FR2: Buyer Information Collection (Priority: Critical)

Introduction: Collect complete buyer information post-vehicle selection.

Inputs:

- Required Fields:
  - Full Name (100 chars, alphabetic)
  - Email (valid format)
  - Phone (E.164 format: +1234567890)
  - Delivery Address (200 chars)

Processing:

1. Client + server-side validation
2. Email format verification (regex)
3. Phone E.164 validation
4. Store complete order with vehicle_type_id

Outputs:

- Validation Confirmation:
  - Order VTMS-ORD-20260108-0001 ready for payment
  - Total: $25,000 — Advance: $7,500

y Now → Secure Gateway

3.2.3 FR3: Advance Payment Processing (Priority: Critical)

Introduction: Process 30% advance payment and generate tracking ID.
Inputs:

- Order ID (VTMS-ORD-YYYYMMDD-XXXX)
- Payment amount (must = 30% of total)
- Payment gateway callback (success/failure)

Processing:

- IF payment_success AND amount == expected_advance:
  1. Generate Tracking ID: VTMS-YYYYMMDD-XXXX
  2. Create shipment record: status="Payment Received"
  3. Update order status: "Awaiting Export"
  4. Send confirmation Email/SMS to buyer
  5. Notify Transport Manager of new order
- ELSE:
  - Display payment failure + retry option

Outputs:

- SUCCESS:
  - ORDER CONFIRMED
  - Tracking ID: VTMS-20260108-0001
  - Vehicle: Sedan (White)
  - Advance Paid: $7,500 / $25,000 (30%)
  - Status: Awaiting Export
  - Track Progress → [Dashboard Link]

3.2.4 FR4: Shipment Tracking Dashboard (Priority: Critical)

Introduction: Real-time tracking interface accessible via Tracking ID.
Inputs: Tracking ID (VTMS-YYYYMMDD-XXXX format)
Processing:

1. Validate Tracking ID exists
2. Retrieve complete shipment timeline
3. Display current status with timestamps
4. Show "Cancel Order" button (pre-final-payment only)

Outputs:

- TRACKING DASHBOARD:
  - Tracking: VTMS-20260108-0001
  - Payment Received 2026-01-08 14:30 UTC
  - Exported to Port [Pending]
  - In Transit [Pending]
  - Final Payment Due [Pending]
  - Delivered [Pending]

Cancel Order [Refresh Status]

3.2.5 FR5: Transport Manager Status Updates (Priority: High)

Introduction: Transport Manager updates shipment progress triggering notifications.

Inputs:

- Status Options:
  - Exported to Last Shipment Port (timestamp required)
  - In Transit
  - Arrived at Destination Port
  - Ready for Final Payment

Processing:

1. Verify Transport Manager role (RBAC)
2. Update shipment status + record timestamp
3. IF status="Exported to Last Shipment Port":
   - Auto-notify buyer: "Pay final 70%: [payment link]"
4. Notify Admin + Finance Manager
5. Audit log: user_id + timestamp + status_change

3.2.6 FR6: Final Payment Processing (Priority: Critical)

Introduction: Process 70% final payment after export confirmation.

Inputs:

- Tracking ID
- Final payment amount (= 70% of total)
- Payment gateway response

Processing:

1. Verify vehicle_exported_status = true

2. Validate payment_amount == remaining_balance
3. Update order_status = "Full Payment Received"
4. Notify Transport Manager: "Arrange delivery"

3.2.7 FR7: Delivery Confirmation (Priority: High)

Introduction: Transport Manager confirms successful delivery.
Inputs: Tracking ID + Delivery confirmation checkbox
Processing:

1. Verify final_payment_received = true
2. Update status = "Delivered "
3. Record delivery_timestamp
4. Archive to completed_orders table

3.2.8 FR8: Order Cancellation - 50% Refund Policy (Priority: High)

Introduction: Buyer cancellation requires mandatory 3-checkbox verification.
Inputs:

- Mandatory Checkboxes (ALL REQUIRED):
  - 1. Cancel Order
  - 2. Request Refund
  - 3. Bank Account/Credit Card: [_____]

Processing:

- IF all_3_checkboxes_selected AND order_pre_delivery:
  1. Calculate refund = advance_payment × 0.5
  2. Process refund via payment gateway
  3. Update status = "Cancelled (50% Refund)"
  4. Notify Finance Manager + Administrator
- ELSE:
  - Error: "Complete all cancellation requirements"

3.2.9 FR9: Automated Notification System (Priority: Critical)

Introduction: Automatic notifications for all critical workflow events.

Event	Recipients	Method	Priority
Order Confirmed	Buyer	Email + SMS	Critical
Exported to Port	Buyer + Transport	Email + SMS	High
Final Payment Due	Buyer	Email + SMS	Critical
Full Payment Received	All actors	Email	Medium
Order Cancelled	All actors	Email	High
Delivered	All actors	Email	Medium

3.2.10 FR10: Administrator Functions (Priority: High)

- User Management: CRUD operations for all 4 roles
- Vehicle Type Configuration: Casual dropdown maintenance
- Order Management: View/modify/cancel any order
- System Configuration: All business rules
- Reports: Orders, payments, shipments

3.2.11 FR11: Finance Manager Functions (Priority: Medium)

- Complete payment visibility (advance + final)
- Refund processing approval
- Financial reporting and reconciliation
- Payment audit trails

3.2.12 FR12: Vehicle Type Configuration (Priority: Medium)

Introduction: Administrator configures casual vehicle type dropdown options.
Inputs:

- Vehicle Type Name (Sedan, SUV, etc.)
- Standard Price ($10K-$100K range)
- Description (optional)
- Display Order (dropdown position)

Req ID	Description	Specification	Measurement
PR1	Page Load Time	2 seconds (95th percentile)	Load testing
PR2	Payment Processing	5 seconds	Gateway logs
PR3	Concurrent Users	500 simultaneous	Stress testing
PR4	Database Queries	2 seconds (10K records)	Query monitoring
PR5	System Availability	99.5% uptime	Uptime monitoring
PR6	Notification Delivery	30 seconds	Service logs

ID	Constraint	Rationale
DC1	Web-only deployment	Project timeline
DC2	IEEE 830-1998 format	Academic requirement
DC3	HTTPS/TLS 1.3 required	Security compliance
DC4	RBAC mandatory	Data protection
DC5	50% refund policy	Company policy

3.3 Performance Requirements

3.4 Design Constraints

3.5 Software System Attributes

3.5.1 Security Requirements

AUTHENTICATION:

- MFA required for Admin/Finance roles
- Password complexity: 12+ chars (UC/LC/Numbers/Symbols)
- Session timeout: 30 minutes inactivity
- Failed login lockout: 5 attempts → 15 minute lockout

AUTHORIZATION:

- Strict RBAC with 4 distinct roles
- Least privilege principle enforced
- Admin-only functions protected

DATA PROTECTION:

- PII encrypted AES-256 at rest
- All traffic TLS 1.3

- Payment data PCI-DSS compliant
- Full audit trail (who/what/when)

3.5.2 Reliability Requirements

FAULT TOLERANCE:

- Graceful error recovery
- Automatic payment retry (3 attempts)
- Database transaction rollback on failure

BACKUP/RECOVERY:

- Daily automated backups (00:00-01:00)
- RTO 4 hours, RPO 1 hour
- Offsite backup storage
- Point-in-time recovery capability

3.5.3 Availability Requirements

- UPTIME: 99.5% ($\leq$ 3.65 hours downtime/year)
- MTBF: $>$ 1000 hours
- MTTR: $<$ 2 hours (95th percentile)
- MAINTENANCE: $<$ 4 hours/month (02:00–06:00 local time)

3.5.4 Maintainability Requirements

- MODULARITY: Microservices architecture
- TEST COVERAGE: $\geq$ 80% unit tests
- DOCUMENTATION: API docs (OpenAPI 3.0), user manuals
- DEPLOYMENT: Zero-downtime updates

3.5.5 Usability Requirements

- INTERFACE: Mobile-responsive (320px-1920px)
- ACCESSIBILITY: WCAG 2.1 AA compliant
- FORMS: Real-time validation + help text
- WORKFLOW: 3-step guided process (Select$\rightarrow$Info$\rightarrow$Pay)

3.6 Other Requirements

3.6.1 Database Requirements

CORE TABLES:

1. users: id, role, email, password_hash
2. vehicle_types: id, type_name, standard_price
3. orders: id, tracking_id, vehicle_type_id, status
4. payments: id, order_id, amount, type, status
5. shipments: id, order_id, status, timestamps
6. notifications: id, event_type, recipients, status

4 Supporting Information

4.1 System Workflow

Workflow:

4.2 Future Enhancements (Post v1.0)

PHASE 2 (Q2 2026):

- Mobile applications (iOS/Android)
- Real-time GPS tracking integration
- Advanced analytics dashboard

PHASE 3 (Q4 2026):

- Multi-language support
- International currency/payment methods
- Predictive delivery time estimates

4.3 Appendix A: Context Diagram

Description: The context diagram illustrates the high-level interactions between the Vehicle Transport Management System (VTMS) and external entities, including customers, transport staff, administrators, and payment gateways (finance manager).

4.4 Appendix B: Usecase Diagram

Description: The use case diagram represents the functional interactions between system users and the Vehicle Transport Management System (VTMS). It highlights major system functionalities from the user perspective.

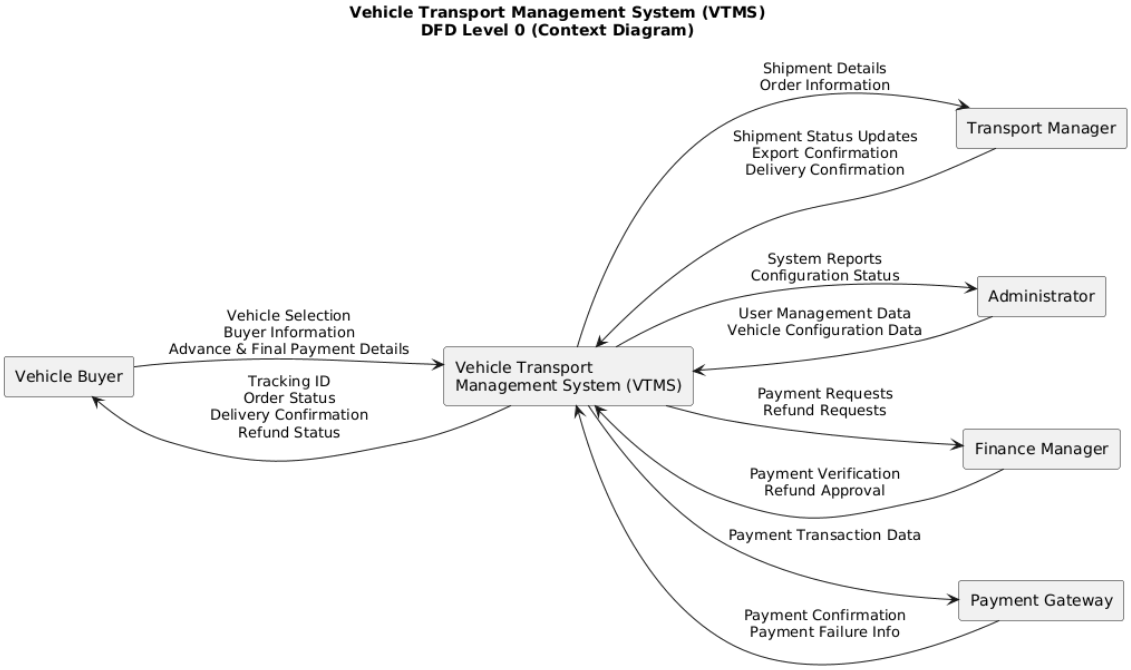

Figure 1: High-level context diagram of VTMS

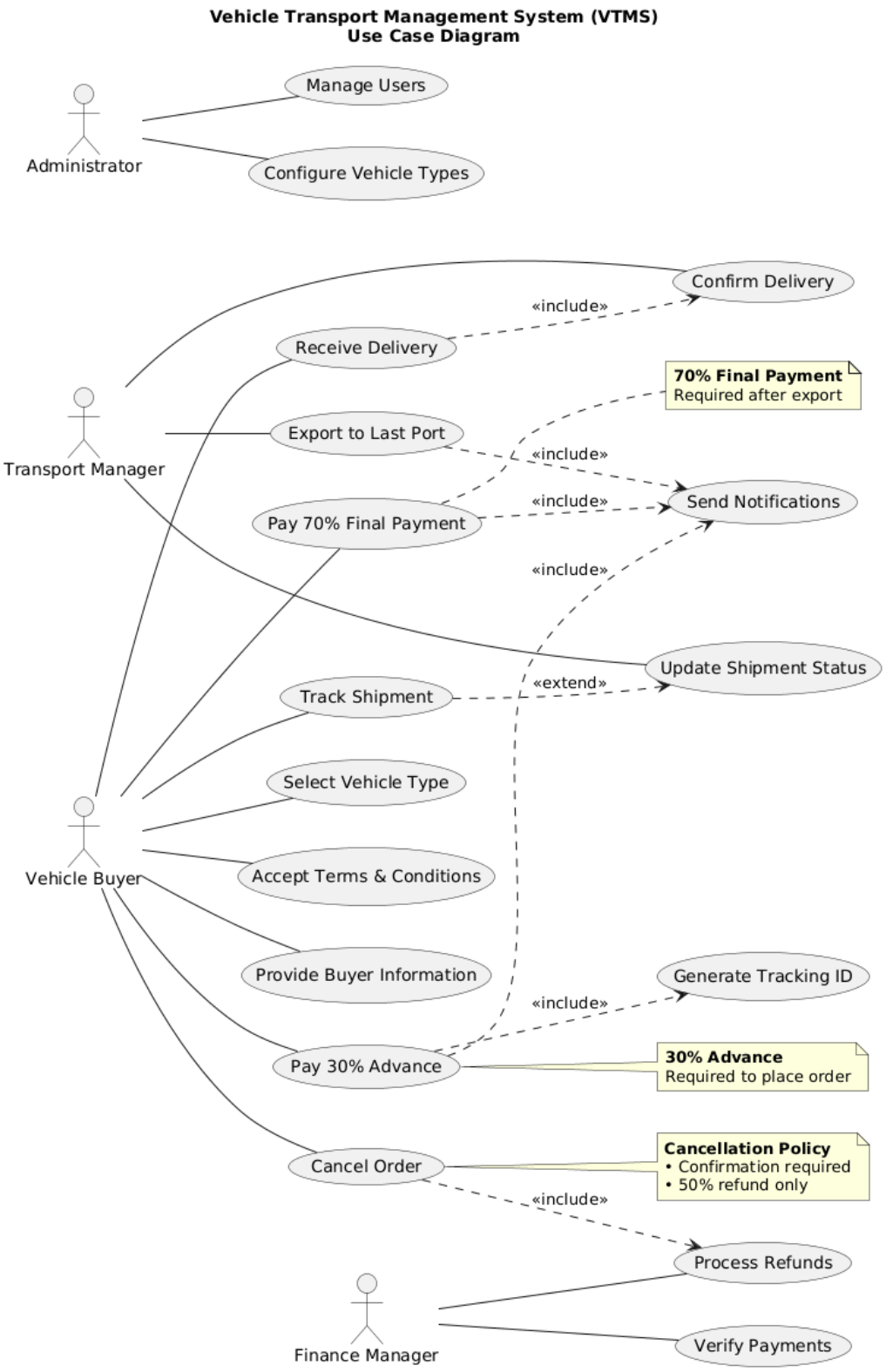

Figure 2: Use Case Diagram of VTMS