**Detailed Report**

**System Requirements**

**Purpose**

The purpose of this project is to develop a comprehensive Transportation Management System (TMS) designed to manage and streamline various aspects of transportation logistics. The system aims to efficiently manage customers, orders, shipments, vehicles, drivers, routes, warehouses, and inventory, ensuring seamless operations and accurate tracking.

**Functional Requirements**

1. **Customer Management**:
   * Create, read, update, and delete (CRUD) operations for customer information.
   * Store customer contact details and addresses.
2. **Order Management**:
   * Place and track orders.
   * Associate orders with customers and shipments.
   * Handle special instructions for orders.
3. **Shipment Tracking**:
   * Track shipment status from pickup to delivery.
   * Record pickup and estimated delivery dates.
   * Manage shipment stops and routes.
4. **Vehicle and Driver Records**:
   * Maintain vehicle information including type, capacity, and maintenance records.
   * Track driver details, license information, and availability status.
   * Assign drivers to vehicles.
5. **Route and Stop Planning**:
   * Plan and optimize delivery routes.
   * Manage stops within each route.
   * Associate routes with shipments and warehouses.
6. **Warehouse and Inventory Management**:
   * Manage warehouse details including location and storage capacity.
   * Track inventory items within warehouses.
   * Record quantities and types of inventory items.
7. **Expense Recording**:
   * Record and categorize expenses related to shipments.
   * Associate expenses with specific shipments for detailed cost tracking.

**Non-Functional Requirements**

1. **Scalability**:
   * The system should be able to handle an increasing amount of data as the business grows.
   * Support for scaling out to multiple servers or cloud infrastructure as needed.
2. **Data Integrity and Security**:
   * Ensure data accuracy through validation and constraints.
   * Implement security measures to protect sensitive information, including encryption and access controls.
   * Regular backups and data recovery procedures.
3. **Usability**:
   * Provide a user-friendly interface for managing data.
   * Ensure the system is easy to navigate and operate for end-users with varying levels of technical expertise.
   * Include help and documentation features.
4. **Performance**:
   * Ensure the system performs efficiently with minimal latency.
   * Optimize database queries and operations for speed.
5. **Reliability**:
   * Ensure high availability and uptime of the system.
   * Implement robust error handling and logging mechanisms.
6. **Compliance**:
   * Adhere to relevant legal and regulatory requirements for data protection and transportation logistics.

**Business Rules**

1. **Customer Orders**:
   * A customer can place multiple orders, establishing a 1:N relationship between customers and orders.
2. **Order Shipments**:
   * An order can be part of multiple shipments, establishing a 1:N relationship between orders and shipments.
3. **Shipment Expenses**:
   * A shipment can incur multiple expenses, establishing a 1:N relationship between shipments and expenses.
4. **Shipment Stops**:
   * A shipment can include multiple stops, establishing a 1:N relationship between shipments and stops.
5. **Vehicle Shipments**:
   * A vehicle can transport multiple shipments, and a driver can drive multiple vehicles through the **VehicleDriver** junction table.
6. **Inventory Storage**:
   * Inventory items are stored in warehouses, establishing a 1:N relationship between warehouses and inventory items.
7. **Routes and Stops**:
   * Routes include multiple stops, establishing a 1:N relationship between routes and stops.
   * Routes are associated with warehouses, establishing a 1:N relationship between warehouses and routes.

**Complete Entity Relationship (ER) Model**

The ER model includes the following entities, attributes, keys, cardinality, connectivity, and relationships.

**Entities and Attributes**

* **Person**: **PersonID**, **Name**, **ContactInformation**
* **Customer**: **PersonID** (FK), **Address**
* **Driver**: **PersonID** (FK), **LicenseInformation**, **AvailabilityStatus**
* **Vehicle**: **VehicleID**, **LicensePlateNumber**, **Capacity**
* **Truck**: **VehicleID** (FK), **TruckType**, **MaximumLoad**
* **Van**: **VehicleID** (FK), **VanType**, **FuelType**
* **Order**: **OrderID**, **CustomerID** (FK), **Order\_Date**, **Items**, **Destination\_Address**, **Special\_Instructions**
* **Shipment**: **ShipmentID**, **OrderID** (FK), **Pickup\_Date**, **Estimated\_DeliveryDate**, **Current\_Status**, **VehicleID** (FK)
* **Expense**: **ExpenseID**, **ShipmentID** (FK), **ExpenseType**, **Amount**
* **Stop**: **StopID**, **RouteID** (FK), **ShipmentID** (FK), **SequenceNumber**, **DeliveryAddress**
* **Route**: **RouteID**, **WarehouseID** (FK), **DestinationWarehouse**, **EstimatedTravelTime**, **Distance**
* **Warehouse**: **WarehouseID**, **Location**, **Storage\_Capacity**
* **Inventory**: **InventoryID**, **WarehouseID** (FK), **ItemType**, **Quantity**
* **VehicleDriver**: **VehicleID** (FK), **PersonID** (FK)

**Keys**

* **Primary Keys**: **PersonID**, **VehicleID**, **OrderID**, **ShipmentID**, **ExpenseID**, **StopID**, **RouteID**, **WarehouseID**, **InventoryID**
* **Foreign Keys**:
  + **PersonID** in **Customer** references **Person**
  + **PersonID** in **Driver** references **Person**
  + **VehicleID** in **Truck** and **Van** reference **Vehicle**
  + **CustomerID** in **Order** references **Customer**
  + **OrderID** in **Shipment** references **Order**
  + **VehicleID** in **Shipment** references **Vehicle**
  + **ShipmentID** in **Expense** references **Shipment**
  + **RouteID** and **ShipmentID** in **Stop** reference **Route** and **Shipment** respectively
  + **WarehouseID** in **Route** and **Inventory** reference **Warehouse**
  + **VehicleID** and **PersonID** in **VehicleDriver** reference **Vehicle** and **Driver** respectively

**Cardinality and Connectivity**

* **Customer and Order**: 1:N (one customer places many orders)
* **Order and Shipment**: 1:N (one order can be part of many shipments)
* **Shipment and Expense**: 1:N (one shipment can have many expenses)
* **Shipment and Stop**: 1:N (one shipment can include many stops)
* **Route and Stop**: 1:N (one route includes many stops)
* **Warehouse and Route**: 1:N (one warehouse can have many routes)
* **Warehouse and Inventory**: 1:N (one warehouse stores many inventory items)
* **Vehicle and Driver**: N:1 (one vehicle can be driven by many drivers through the **VehicleDriver** table)
* **Vehicle and Shipment**: 1:1 (one shipment is transported by one vehicle)

**Assumptions**

* Each customer, driver, and vehicle is uniquely identifiable by their respective IDs.
* Vehicles are classified as either trucks or vans.
* Each order is associated with a single customer but can be included in multiple shipments.
* Each shipment is transported by a single vehicle and includes multiple stops.
* Expenses are directly related to shipments.

**SQL DDL and DML Statements**

**SQL DDL Statements (Data Definition Language)**

-- Supertype: PERSON

CREATE TABLE Person (

PersonID INT PRIMARY KEY,

Name VARCHAR(100) NOT NULL,

ContactInformation VARCHAR(150) NOT NULL

);

-- Subtype: CUSTOMER

CREATE TABLE Customer (

PersonID INT PRIMARY KEY,

Address VARCHAR(150) NOT NULL,

FOREIGN KEY (PersonID) REFERENCES Person(PersonID)

);

-- Subtype: DRIVER

CREATE TABLE Driver (

PersonID INT PRIMARY KEY,

LicenseInformation VARCHAR(100) UNIQUE NOT NULL,

AvailabilityStatus VARCHAR(50) NOT NULL,

FOREIGN KEY (PersonID) REFERENCES Person(PersonID)

);

-- Subtype: ADMINISTRATOR

CREATE TABLE Administrator (

PersonID INT PRIMARY KEY,

Department VARCHAR(100) NOT NULL,

FOREIGN KEY (PersonID) REFERENCES Person(PersonID)

);

-- Subtype: DISPATCHER

CREATE TABLE Dispatcher (

PersonID INT PRIMARY KEY,

Shift VARCHAR(50) NOT NULL,

FOREIGN KEY (PersonID) REFERENCES Person(PersonID)

);

-- Supertype: VEHICLE

CREATE TABLE Vehicle (

VehicleID INT PRIMARY KEY,

LicensePlateNumber VARCHAR(20) UNIQUE NOT NULL,

Capacity INT NOT NULL

);

-- ENTITY: WAREHOUSE

CREATE TABLE Warehouse (

WarehouseID INT PRIMARY KEY,

Location VARCHAR(150) NOT NULL,

Storage\_Capacity INT NOT NULL

);

-- ENTITY: INVENTORY

CREATE TABLE Inventory (

InventoryID INT PRIMARY KEY,

WarehouseID INT NOT NULL,

ItemType VARCHAR(100) NOT NULL,

Quantity INT NOT NULL,

FOREIGN KEY (WarehouseID) REFERENCES Warehouse(WarehouseID)

);

-- ENTITY: ROUTE

CREATE TABLE Route (

RouteID INT PRIMARY KEY,

WarehouseID INT NOT NULL,

DestinationWarehouse VARCHAR(150) NOT NULL,

EstimatedTravelTime INT NOT NULL,

Distance DECIMAL(10, 2) NOT NULL,

FOREIGN KEY (WarehouseID) REFERENCES Warehouse(WarehouseID)

);

-- ENTITY: ORDER

CREATE TABLE `Order` (

OrderID INT PRIMARY KEY,

CustomerID INT NOT NULL,

Order\_Date DATE NOT NULL,

Items TEXT NOT NULL,

Destination\_Address VARCHAR(150) NOT NULL,

Special\_Instructions TEXT,

FOREIGN KEY (CustomerID) REFERENCES Customer(PersonID)

);

-- ENTITY: SHIPMENT

CREATE TABLE Shipment (

ShipmentID INT PRIMARY KEY,

OrderID INT NOT NULL,

Pickup\_Date DATE NOT NULL,

Estimated\_DeliveryDate DATE NOT NULL,

Current\_Status VARCHAR(50) NOT NULL,

VehicleID INT NOT NULL,

FOREIGN KEY (OrderID) REFERENCES `Order`(OrderID),

FOREIGN KEY (VehicleID) REFERENCES Vehicle(VehicleID)

);

-- ENTITY: EXPENSE

CREATE TABLE Expense (

ExpenseID INT PRIMARY KEY,

ShipmentID INT NOT NULL,

ExpenseType VARCHAR(50) NOT NULL,

Amount DECIMAL(10, 2) NOT NULL,

FOREIGN KEY (ShipmentID) REFERENCES Shipment(ShipmentID)

);

-- ENTITY: STOP

CREATE TABLE Stop (

StopID INT PRIMARY KEY,

RouteID INT NOT NULL,

ShipmentID INT NOT NULL,

SequenceNumber INT NOT NULL,

DeliveryAddress VARCHAR(150) NOT NULL,

FOREIGN KEY (RouteID) REFERENCES Route(RouteID),

FOREIGN KEY (ShipmentID) REFERENCES Shipment(ShipmentID)

);

-- RELATION: VEHICLE and DRIVER (many-to-many)

CREATE TABLE VehicleDriver (

VehicleID INT NOT NULL,

PersonID INT NOT NULL,

PRIMARY KEY (VehicleID, PersonID),

FOREIGN KEY (VehicleID) REFERENCES Vehicle(VehicleID),

FOREIGN KEY (PersonID) REFERENCES Driver(PersonID)

);

-- Subtype: TRUCK

CREATE TABLE Truck (

VehicleID INT PRIMARY KEY,

TruckType VARCHAR(50) NOT NULL,

MaximumLoad INT NOT NULL,

FOREIGN KEY (VehicleID) REFERENCES Vehicle(VehicleID)

);

-- Subtype: VAN

CREATE TABLE Van (

VehicleID INT PRIMARY KEY,

VanType VARCHAR(50) NOT NULL,

FuelType VARCHAR(50) NOT NULL,

FOREIGN KEY (VehicleID) REFERENCES Vehicle(VehicleID)

);

**SQL DML Statements (Data Manipulation Language)**

-- Insert data into Person

INSERT INTO Person (PersonID, Name, ContactInformation) VALUES (1, 'John Doe', 'john.doe@example.com');

-- Insert data into Customer

INSERT INTO Customer (PersonID, Address) VALUES (1, '123 Main St, Springfield');

-- Insert data into Driver

INSERT INTO Driver (PersonID, LicenseInformation, AvailabilityStatus) VALUES (2, 'DL123456', 'Available');

-- Insert data into Vehicle

INSERT INTO Vehicle (VehicleID, LicensePlateNumber, Capacity) VALUES (1, 'XYZ123', 1000);

-- Insert data into Truck

INSERT INTO Truck (VehicleID, TruckType, MaximumLoad) VALUES (1, 'Semi', 20000);

-- Insert data into Warehouse

INSERT INTO Warehouse (WarehouseID, Location, Storage\_Capacity) VALUES (1, 'Warehouse A', 5000);

-- Insert data into Inventory

INSERT INTO Inventory (InventoryID, WarehouseID, ItemType, Quantity) VALUES (1, 1, 'Electronics', 100);

-- Insert data into Route

INSERT INTO Route (RouteID, WarehouseID, DestinationWarehouse, EstimatedTravelTime, Distance) VALUES (1, 1, 'Warehouse B', 120, 50.5);

-- Insert data into Order

INSERT INTO `Order` (OrderID, CustomerID, Order\_Date, Items, Destination\_Address, Special\_Instructions) VALUES (1, 1, '2024-05-23', 'Item1, Item2', '456 Another St, Springfield', 'Leave at door');

-- Insert data into Shipment

INSERT INTO Shipment (ShipmentID, OrderID, Pickup\_Date, Estimated\_DeliveryDate, Current\_Status, VehicleID) VALUES (1, 1, '2024-05-24', '2024-05-25', 'In Transit', 1);

-- Insert data into Expense

INSERT INTO Expense (ExpenseID, ShipmentID, ExpenseType, Amount) VALUES (1, 1, 'Fuel', 150.00);

-- Insert data into Stop

INSERT INTO Stop (StopID, RouteID, ShipmentID, SequenceNumber, DeliveryAddress) VALUES (1, 1, 1, 1, '456 Another St, Springfield');

-- Insert data into VehicleDriver

INSERT INTO VehicleDriver (VehicleID, PersonID) VALUES (1, 2);

**Data Dictionary**

|  |  |  |
| --- | --- | --- |
| Table | Column | Description |
| Person | PersonID | Unique identifier for a person |
| Person | Name | Name of the person |
| Person | ContactInformation | Contact details of the person |
| Customer | PersonID | Unique identifier for a customer (FK to Person) |
| Customer | Address | Address of the customer |
| Driver | PersonID | Unique identifier for a driver (FK to Person) |
| Driver | LicenseInformation | License information of the driver |
| Driver | AvailabilityStatus | Availability status of the driver |
| Vehicle | VehicleID | Unique identifier for a vehicle |
| Vehicle | LicensePlateNumber | License plate number of the vehicle |
| Vehicle | Capacity | Capacity of the vehicle |
| Truck | VehicleID | Unique identifier for a truck (FK to Vehicle) |
| Truck | TruckType | Type of the truck |
| Truck | MaximumLoad | Maximum load capacity of the truck |
| Van | VehicleID | Unique identifier for a van (FK to Vehicle) |
| Van | VanType | Type of the van |
| Van | FuelType | Fuel type of the van |
| Warehouse | WarehouseID | Unique identifier for a warehouse |
| Warehouse | Location | Location of the warehouse |
| Warehouse | Storage\_Capacity | Storage capacity of the warehouse |
| Inventory | InventoryID | Unique identifier for an inventory item |
| Inventory | WarehouseID | Identifier for the warehouse (FK to Warehouse) |
| Inventory | ItemType | Type of item in inventory |
| Inventory | Quantity | Quantity of the item in inventory |
| Route | RouteID | Unique identifier for a route |
| Route | WarehouseID | Identifier for the warehouse (FK to Warehouse) |
| Route | DestinationWarehouse | Destination warehouse for the route |
| Route | EstimatedTravelTime | Estimated travel time for the route |
| Route | Distance | Distance of the route |
| Order | OrderID | Unique identifier for an order |
| Order | CustomerID | Identifier for the customer (FK to Customer) |
| Order | Order\_Date | Date of the order |
| Order | Items | Items included in the order |
| Order | Destination\_Address | Destination address for the order |
| Order | Special\_Instructions | Special instructions for the order |
| Shipment | ShipmentID | Unique identifier for a shipment |
| Shipment | OrderID | Identifier for the order (FK to Order) |
| Shipment | Pickup\_Date | Pickup date for the shipment |
| Shipment | Estimated\_DeliveryDate | Estimated delivery date for the shipment |
| Shipment | Current\_Status | Current status of the shipment |
| Shipment | VehicleID | Identifier for the vehicle (FK to Vehicle) |
| Expense | ExpenseID | Unique identifier for an expense |
| Expense | ShipmentID | Identifier for the shipment (FK to Shipment) |
| Expense | ExpenseType | Type of the expense |
| Expense | Amount | Amount of the expense |
| Stop | StopID | Unique identifier for a stop |
| Stop | RouteID | Identifier for the route (FK to Route) |
| Stop | ShipmentID | Identifier for the shipment (FK to Shipment) |
| Stop | SequenceNumber | Sequence number of the stop |
| Stop | DeliveryAddress | Delivery address for the stop |
| VehicleDriver | VehicleID | Identifier for the vehicle (FK to Vehicle) |
| VehicleDriver | PersonID | Identifier for the driver (FK to Driver) |