# Order Management System - Requirements Document

**Version:** 2.1 **Date:** July 31, 2025 **Author:** Product Management

### 1. Application Overview

This document outlines the requirements for an **Order Management System**. The primary goal is to create an efficient, centralized platform for managing the entire order lifecycle, from creation and processing to shipping and delivery. The system will integrate with external enterprise systems for customer and product data, provide role-based access control, and offer real-time tracking capabilities.

### 2. General Application Settings

The application must use the **"Mentor"** theme available in the ODC tenant.

### 3. Data Model

The application will use the following data model, which includes local entities and entities sourced from external systems.

#### Entities & Attributes

* **Entity: Order**
  + This entity is stored **locally**.
  + Attributes include:
    - Id: An Identifier that serves as the Primary Key.
    - OrderNumber: Text, an auto-generated, unique order identifier.
    - OrderDate: DateTime, a timestamp of when the order was placed.
    - TotalAmount: Currency, the calculated total cost of the order.
    - ShippingAddress: Text, the delivery address for the order.
    - CustomerId: An Identifier from **Salesforce** that is a Foreign Key to the Customer entity.
    - StatusId: An Identifier that is a Foreign Key to the OrderStatus static entity.
* **Entity: OrderItem**
  + This entity is stored **locally**.
  + Attributes include:
    - Id: An Identifier that serves as the Primary Key.
    - Quantity: An Integer representing the number of units for a specific product.
    - UnitPrice: Currency, the price per unit at the time of order.
    - OrderId: An Identifier that is a Foreign Key to the Order entity.
    - ProductId: An Identifier that is a Foreign Key to the Product entity.
* **Entity: Customer**
  + This entity's data is sourced from **Salesforce**.
  + Attributes include:
    - Id: An Identifier that is the Primary Key from Salesforce.
    - Name: Text, the customer's full name or company name.
    - Email: Email, the customer's primary email address.
    - Phone: Phone Number, the customer's primary phone number.
    - BillingAddress: Text, the customer's billing address.
* **Entity: Product**
  + This entity is stored **locally**.
  + Attributes include:
    - Id: An Identifier that serves as the Primary Key.
    - SKU: Text, the Stock Keeping Unit.
    - Name: Text, the product name.
    - Description: Text, a detailed product description.
    - Price: Currency, the current price of the product.
    - StockQuantity: An Integer representing the current quantity in stock.

#### Entity Relationships

* A **Customer** can have many **Orders** (One-to-Many).
* An **Order** can have many **OrderItems** (One-to-Many).
* A **Product** can be in many **OrderItems** (One-to-Many).

### 4. Static Entities

* **Entity Name:** OrderStatus
* **Purpose:** This static entity defines the possible states of an order throughout its lifecycle.
* **Records:** The records for OrderStatus are Pending, Confirmed, Processing, Shipped, Delivered, and Cancelled.

### 5. Roles & Permissions

The application must have the following user roles with simplified access levels.

* **Role: Admin**
  + Full Access to all entities.
* **Role: Sales Rep**
  + Order: Edit Access
  + OrderItem: Edit Access
  + Customer: View Access
  + Product: View Access
* **Role: Warehouse Manager**
  + Order: View Access.
  + OrderItem: View Access.
  + Customer: View Access
  + Product: View Access.
* **Role: Customer Service**
  + Order: View Access.
  + OrderItem: View Access.
  + Customer: Edit Access.
  + Product: View Access
* **Special Permissions:**
  + Logged-in users can edit their own records in Orders entity.

### 6. Main Features & Screens

The application should include the following main screens and features:

1. **Dashboard:** A landing page displaying key metrics like total orders, pending orders, and revenue. It should also show a list of recent orders.
2. **Order Management:** This includes a screen to view a list of all orders with filtering and searching capabilities (by status, customer, date), a screen to view the details of a single order, and a form to create a new order.
3. **Order Tracking Map View:** A dedicated screen that displays a map. For orders with a status of "Shipped," a marker should be placed on the map to visualize the real-time location of the delivery. This requires integration with a shipping service API.
4. **Product Catalog:** A read-only screen that displays a list of all products sourced from SAP.
5. **Customer Directory:** A read-only screen that displays a list of all customers sourced from Salesforce.