Software Requirements Specification
Supply Chain Management System
Version 1.0

Prepared by: Group 36
210467A 210189X 210362X 210726T 210472J
Date created: 10.09.2023

Contents

1 Introduction

1.1 Purpose

The purpose of this document is to present a detailed description of the Supply Chain Management System developed for Company A. It will explain the purpose and features of the system, the interfaces of the system, what the system will do, the constraints under which it must operate and how the system will react to external stimuli. This document is intended for both the stakeholders and the developers of the system.

1.2 Intended Audience and Reading Suggestions

1. Administration
2. Store employees
3. Customers
4. Truck drivers and assistants

1.3 Product Scope

The SCMS will include the following features:

1. The ability to track the movement of goods from the factory to the customers.
2. Assignment of drivers, driver assistants, and trucks to delivery schedules.
3. The ability to generate reports for the management of the company.
4. Route planning from stores to customer addresses.

2 Overall Description

2.1 Product Perspective

The proposed system will act as the central hub for managing the product distribution process, from order placement to delivery. The system will also offer reporting and analytics capabilities to provide valuable insights for decision-making.

2.2 Product Functions

The system has a range of vital functions to facilitate the supply chain process of the Company A. These functions include:

- Customers can place orders, specify delivery dates and preferred routes, and track their orders through the customer-dedicated application.
- The system allocates products to train trips based on predefined capacities and trip schedules.
- The store employee has a dedicated application from which he/she can update the status of goods delivered by train by scanning a barcode using their smartphones.
- Stores are associated with predefined routes, enabling efficient delivery scheduling.
- Trucks, drivers, and driver assistants are assigned to specific schedules based on defined rosters.
- Ensures compliance with maximum work hour constraints for drivers and driver assistants.
- Provides comprehensive reports including quarterly sales, popular products, sales by cities and routes, Working Hours of Drivers/Driver Assistants, Used hours of Trucks, and Customer-order reports.

2.3 User Classes and Characteristics

The system caters to various user classes:

- **Administrators** : Responsible for system configuration, managing users, and overseeing overall operations.
- **Managers**: Oversee specific aspects like route planning, resource allocation, and reporting.
- **Drivers and Driver Assistants**: Execute assigned schedules, adhere to work hour constraints, and report on deliveries.
- **Store Employees**: Scan barcodes and update the status of products, such as status after picking up from the train, and the status before loading onto trucks, etc.
- **Customers**: Place orders, select delivery routes, and track order status.

2.4 Operating Environment

The system is designed to operate in a standard computing environment. It requires:

For the application dedicated to administration,

- Operating System: Compatible with Windows, MacOS, and Linux.
- Web browser: Compatible with modern browsers such as Edge, Chrome, Firefox, Safari, etc.
- Internet connection: Stable internet connectivity is essential for real-time data updates and communication.

For the application dedicated to store employees,

- Operating System: Android (Downloadable from the website dedicated to the system).
- Internet connection: Stable internet connectivity is essential for real-time data updates and communication.

For the application dedicated to end customers,

- Operating System: Android (Downloadable from Google Play).
- Internet connection: Stable internet connectivity is essential for real-time data updates and communication.

2.5 Design and Implementation Constraints

The system is subject to the following constraints:

- Capacity allocation: Orders exceeding train trip capacities must be scheduled for the subsequent trip.
- Work-hour constraints: Drivers and driver assistants have predefined maximum work-hour limits.
- Route limitations: Transportation via railway is limited to specific main cities.
- Route predefinition: Routes are predefined to ensure efficient coverage of associated areas.

2.6 User Documentation

Comprehensive user documentation will be provided, including:

- User guides: Detailed instructions for administrators, managers, store employees, drivers, driver assistants, and customers.
- Training materials: Tutorials and resources to assist store employees, drivers, and driver assistants in becoming proficient with the system.

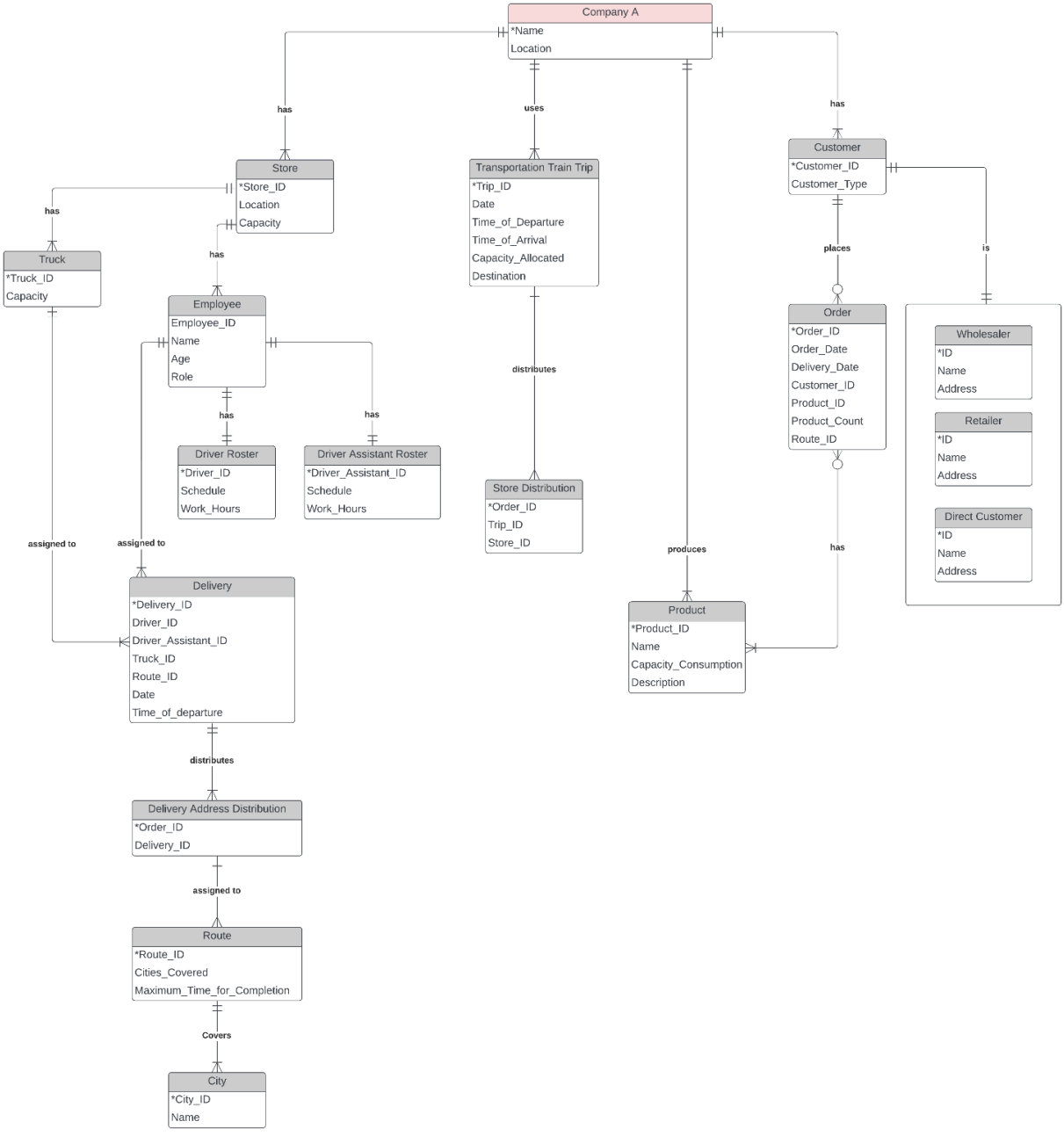

Figure 1: ER diagram of the system.

2.7 Assumptions and Dependencies

The system operates under certain assumptions and dependencies:

- Customers are expected to place orders at least 7 days before the desired delivery date.
- The system assumes a negotiated capacity allocation from the railway department for each transportation train trip.

3 External Interface Requirements

3.1 User Interfaces

3.1.1 Administration User Interface (Admin UI)

- Technology: React
- UI Library: Flowbite UI library

The Admin UI will be developed using the React framework, providing a robust and interactive web application for system administrators. It will leverage the Flowbite UI library to ensure a consistent and visually appealing user interface. This interface will serve as the central hub for administrators to manage users, permissions, system configurations, and generate reports.

3.1.2 Store Management User Interface (Store UI)

- Technology: Windows Forms
- UI Library: Windows Forms follows native UI design

The Store Management UI will be developed as a Windows Forms application, providing store managers with a desktop-based platform. It follows the native Windows Forms UI design guidelines, ensuring an intuitive and familiar user experience. This interface enables store managers to efficiently update package status, manage inventory, and monitor order fulfillment.

3.1.3 Driver Status Update User Interface (Driver UI)

- Technology: Flutter
- UI Library: Flutter Widgets

The Driver Status Update UI will be developed as a mobile application using the Flutter framework. It provides drivers with a platform to update package statuses, share real-time locations, and communicate with the central system. The UI will leverage Flutter's built-in widgets and design principles to create a clean and intuitive mobile interface

3.1.4 Customer User Interface (Customer UI)

- Technology: svelte
- UI Library: Flowbite UI library

The Customer UI will be developed as a Svelte web application, catering to end customers, wholesalers, and retail shops. It utilizes the Svelte framework for a reactive and efficient UI development process. The Flowbite UI library will be used to ensure a visually appealing and user-friendly interface. This UI enables users to place orders, track shipments, view order history, and manage their accounts seamlessly.

3.2 Hardware Interfaces

3.2.1 Server

The system will run on a DigitalOcean Droplet, which will host the backend services, databases, and web server. The server hardware specifications must meet the minimum requirements specified in the system documentation.

- Compatibility: The system is compatible with standard Digital Ocean Droplet configurations, including CPU, RAM, and storage options.
- Connection: The server must have a stable internet connection to ensure uninterrupted system operation and communication with external services.

3.2.2 Desktop / Laptop Computers

Admin users and store employees will access the system through desktop or laptop computers. These devices should meet the minimum system requirements outlined in the system documentation to ensure optimal performance.

- Operating System: Compatible with Windows, macOS, and Linux operating systems
- Browser Compatibility: The system is compatible with the latest versions of popular web browsers such as Chrome, Firefox, Safari, and Edge.

3.2.3 Wireless Barcode Reader

Store employees will utilize a USB radio wireless barcode reader for efficient inventory management and order processing. The barcode reader should support standard barcode formats and connect to devices via USB radio technology.

- Compatibility: The barcode reader must be compatible with the system's barcode scanning protocol and have USB radio connectivity.
- Connection: The barcode reader will connect to the computer via a USB port, utilizing radio frequency technology for seamless data transmission. It should be plug-and-play compatible with standard operating systems.
- Power Source: The barcode reader is powered by a rechargeable battery, providing flexibility for use in various store locations without being tethered to a power source.
- Range: The wireless range of the barcode reader should be sufficient for efficient scanning within the store environment.

3.2.4 Mobile Phones

Truck drivers and their assistants will use mobile phones to access the system for package status updates and location tracking. The mobile phones should be smartphones running Android or iOS operating systems.

- App Compatibility: The mobile application for the system must be available and compatible with the respective app stores (Google Play Store for Android, Apple App Store for iOS).
- Internet Connectivity: Mobile phones must have access to mobile data or a stable Wi-Fi connection to ensure real-time updates and communication with the system.

3.3 Software Interfaces

3.3.1 Ubuntu Server

The application will be hosted on an Ubuntu Server for deployment. It will utilize the Linux environment for runtime execution and resource management.

3.3.2 Web Browsers

The application will be accessed and utilized through standard web browsers like Chrome, Firefox, and Safari. It will be compatible with the latest versions of these browsers to ensure optimal performance and user experience.

3.3.3 MySQL

The system will rely on MySQL as the primary relational database management system. It will store structured data related to users, orders, inventory, and other critical information.

Connections: The application will establish connections to MySQL using appropriate database drivers and protocols, adhering to industry-standard SQL queries for data retrieval and manipulation.

3.3.4 Redis

Redis will be employed as an in-memory data structure store, acting as a cache to enhance data retrieval performance. It will be instrumental in reducing the load on the primary MySQL database.

Connections: The system will utilize Redis client libraries to establish connections, allowing for efficient caching and retrieval of frequently accessed data.

3.3.5 .NET Core

The application will be built on the .NET Core framework. This framework provides a robust and flexible environment for developing, deploying, and managing software applications.

Connections: .NET Core will be integrated with the application codebase, offering a wide range of libraries and runtime support for various functionalities.

3.3.6 .NET Core Identity platform

The .NET Core Identity platform will handle user authentication, authorization, and management. It will provide a secure and standardized way to manage user accounts, roles, and permissions within the system.

Connections: The main application will integrate with the .NET Core Identity platform to facilitate user-related operations, ensuring secure access and data protection.

3.3.7 Flutter

Flutter will be used to create the app for the employees. Bar-code scanning and notifying drivers will be done through a flutter app.

Connections: Employee app will communicate with backend server through web-sockets and HTTPS

3.3.8 React

React will be employed for building the Admin UI. It is a popular JavaScript library for creating dynamic and interactive user interfaces.

Connections: The Admin UI, developed using React, will communicate with the backend server through RESTful APIs over HTTPS. It will facilitate various administrative functions such as user management, system configuration, and report generation.

3.3.9 Svelte

React will be employed for building the Admin UI. It is a popular JavaScript library for creating dynamic and interactive user interfaces

Connections: The Admin UI, developed using React, will communicate with the backend server through RESTful APIs over HTTPS. It will facilitate various administrative functions such as user management, system configuration, and report generation.

3.3.10 Flowbite

Flowbite will be employed for building the Admin UI and End customer UI. It is a popular JavaScript library for creating dynamic and interactive user interfaces.

Connections: The Admin UI, developed using React, will communicate with the backend server through RESTful APIs over HTTPS. It will facilitate various administrative functions such as user management, system configuration, and report generation.

3.4 Communications Interfaces

3.4.1 HTTPS - Rest APIs for Communication

The system will primarily rely on HTTPS (Hypertext Transfer Protocol Secure) for communication between different components. This protocol ensures secure and encrypted data exchange over the internet. The communication will be facilitated through Restful APIs (Application Programming Interfaces), allowing seamless interaction between various modules of the Supply Chain Management System.

- Requirements
  - Data in JSON format.
  - HTTPS for encryption.
  - Error handling with standard codes.

3.4.2 2.4GHz Radio and USB for Barcode Reader

The system will use 2.4GHz radio communication between the wireless barcode reader and the usb dongle which will intern communicate with the computer via USB.

- Requirements
  - Standardized data format.
  - Adherence to wireless and USB protocols.
  - Plug-And-Play
  - Secure data transmission.

4 System Features

System feature of this system needs to be partitioned into three sub-systems because we have three separate interfaces for the Customers(to place the order), Employees(to update the state of the order) and Administration(to monitor the system).

4.1 Customer Interface

4.1.1 Place the order (FC-1)

Description and Priority	This feature will allow customers to place orders or to schedule the orders. Benefit - 9 — Penalty - 0 — Cost - 1 — risk - 3
Stimulus/Response Sequences	Items will be listed out for the customers to select. There will be a calendar view for them to schedule the order. Delivery details will be taken from the registration interface. In this case, customers will not have to enter details again.
Functional Requirements	After selecting the items and scheduling the date, details will be posted to the server, and the order will be placed. In case if a customer failed to enter any details or there are connection issues, a modal view with an error message will be popped up. The UI needs to be responsive because most of the customers will be placing orders through a mobile device.

4.1.2 Track the order (FC-2)

Description and Priority	Using this feature, customers will be able to track the order (current state of the order). Customers will be informed whether the order is being processed for delivery, on the train, arrived at a warehouse, or released by the warehouse. Completed orders will also be displayed here. Benefit - 9 — Penalty - 3 — Cost - 3 — risk - 7
Stimulus/Response Sequences	All the orders that have been placed by the customer will be listed here (most recent first). By clicking on an order, the customer will get the details of the order (current state, Expected delivery date, etc).
Functional Requirements	Current state needs to be updated frequently to avoid unnecessary complaints to the administration. Websockets will be used to do that. Completed orders will be stored in local storage to avoid unnecessary traffic to the main server.

4.1.3 Cancel the order (FC-3)

Description and Priority	Customers might place orders mistakenly, and a requirement can occur from their side to cancel an order. However, it must not interrupt the delivery process. If the order is still being processed in the store, customers will be able to cancel the order. Benefit - 7 — Penalty - 5 — Cost - 0 — risk - 6
Stimulus/Response Sequences	There will be a button to cancel the order in the tracking order section.
Functional Requirements	The current state will be checked after the button press. An error message will be displayed if it is not in a reversible state.

4.1.4 Registration (FC-4)

Description and Priority	It is important to save customer data using an account. This feature will take care of the registration process. Benefit - 9 — Penalty - 3 — Cost - 5 — risk - 5
Stimulus/Response Sequences	There will be form to get user data. Form data will go through a validation process.
Functional Requirements	Form validation is needed(A model view is needed to show errors). form data will be posted to the main server.

4.2 Admin Interface

4.2.1 Adding Employees (FA-1)

Description and Priority	Managing employees is a major requirement of this system. This feature will be dedicated to add new employees and update data of existing employees. Benefit - 7 — Penalty - 4 — Cost - 4 — risk - 4
Stimulus/Response Sequences	There will be a form view to add employee data and existing employee data will be listed out to do modifications
Functional Requirements	After each and every submission employee data will be posted to the server to save data. Empty fields needs to be informed with an error message.

4.2.2 Tracking Order (FA-2)

Description and Priority	Admin users will also need to track the ongoing orders. They must be able to change the state or cancel the order. Benefit - 9 — Penalty - 0 — Cost - 7 — risk - 3
Stimulus/Response Sequences	There will be a list of all the ongoing orders (recent first). Admin users must be able to filter orders by state, address, date etc. Live location of the trucks and trains also needs to be displayed here. In this view admin users will be able to change the state of an order forcefully
Functional Requirements	Data needs to be updated real time. Web sockets will be used to archive that purpose. Google maps api will be used alongside with websockets to do location tracking of vehicles.

4.2.3 Monitoring Employees (FA-3)

Description and Priority	This feature is to give the functionality of monitoring working hours of employees. Benefit - 9 — Penalty - 0 — Cost - 7 — risk - 3
Stimulus/Response Sequences	A list of employees (two columns for drivers and driving assistants) by descending order of work hours will be displayed in this view. If someone is getting near the working hour limit a warning will be displayed.
Functional Requirements	After every order a function will be called to save order start time. with that this feature will be able to calculate working hours. To get employee data quickly a caching system will be implemented.

4.2.4 Items Report (FA-4)

Description and Priority	A report monthly and annual report on sales data will be provided with this feature. Benefit - 8 — Penalty - 0 — Cost - 5 — risk - 3
Stimulus/Response Sequences	A pie chart of items will be shown in the main view. Admin users will be able to select whether they want monthly report or an annual report. they will be able to get historic data as well.
Functional Requirements	There is no need to update this data on real time. Therefore data will be fetched after clicking the refresh button.

4.2.5 Sales Report (FA-5)

Description and Priority	A proper sales report on financial data of items will be provided with this feature Benefit - 8 — Penalty - 0 — Cost - 6 — risk - 3
Stimulus/Response Sequences	Column charts of sales data with the time (month or date) and items will be shown in the main view. Admin users will be able to select whether they want monthly report or an annual report. they will be able to get historic data as well.
Functional Requirements	There is no need to update this data on real time. Therefore data will be fetched after clicking the refresh button.

4.3 Employee Interface

4.3.1 Scanning Order (FE-1)

Description and Priority	This feature is dedicated to change the state of the order each bar-code scan will step up the state of the order. As an example, if an employee in a warehouse scanned the bar-code the state of the order will be changed from "on the train" to "in the warehouse". Benefit - 9 — Penalty - 7 — Cost - 8 — risk - 7
Stimulus/Response Sequences	A camera view will be displayed. It will scan the bar codes. Employees have to scan the bar-code when they load the items or unload them.
Functional Requirements	This needs to be in a mobile app. If a false bar-code or an expired one gets scanned an error message will be popped up and administration will be informed.

4.3.2 Drivers Interface (FE-2)

Description and Priority	This feature is for the drivers and assistants. Assigning them to deliver an order will be done automatically by the system. They will be notified through this interface. They will have to update when they start the turn and end the turn. Benefit - 9 — Penalty - 3 — Cost - 5 — risk - 8
Stimulus/Response Sequences	Finished turns will be displayed in the main view. Driver will be notified by an alarm sound. Driver and assistant will be assign to a truck by id. And there will be a button to press when they start turn and end turn.
Functional Requirements	Since a function in the interface needs to be triggered by an external DB entry web-sockets are needed to be used here. Drivers will be reminded constantly if they did not respond within a specified time period.

5 Other Nonfunctional Requirements

5.1 Performance Requirements

- The database should support concurrent access by multiple users and handle a high volume of transactions efficiently.
- Queries for generating reports, such as quarterly sales reports, should execute within a reasonable time frame (e.g., under 5 seconds).
- The system should have minimal downtime for maintenance, with planned maintenance windows communicated in advance.
- Real-time updates to the database, such as order tracking and inventory adjustments, should be reflected instantly to ensure up-to-date information for users.
- Database backups and disaster recovery procedures should be in place to prevent data loss and minimize downtime in case of system failures.

5.2 Safety Requirements

- The system should enforce access controls to ensure that only authorized personnel can access sensitive data, such as customer information and financial records.
- Data encryption should be implemented to protect customer data during transmission and storage.
- Compliance with safety regulations related to transportation and storage of goods should be ensured, and the system should provide necessary documentation for auditing purposes.

- Safeguards should be in place to prevent data corruption or loss due to hardware failures or unexpected system crashes.
- Safety certifications and compliance with relevant industry standards should be maintained.

5.3 Security Requirements

- User authentication and authorization should be implemented to control access to the database and its functions.
- Audit logs should record all significant database activities for security monitoring and auditing purposes.
- Data privacy should be maintained, and personal information of customers, drivers, and employees should be securely stored and protected.
- Protection against SQL injection, cross-site scripting (XSS), and other common security vulnerabilities should be incorporated into the database design.
- Regular security assessments and penetration testing should be conducted to identify and address vulnerabilities.

5.4 Software Quality Attributes

- The system should prioritize reliability to ensure that orders are fulfilled accurately and on time.
- Database design should be maintainable, allowing for easy updates and modifications as business requirements evolve.
- Data should be stored efficiently to optimize storage space and query performance.
- The system should be highly available, with a documented disaster recovery plan in case of unexpected failures.
- Usability should be a key consideration, with an intuitive interface for users to input data and generate reports.

6 Business Rules

- Only authorized personnel, such as administrators and managers, should have the ability to modify or delete critical data, such as routes, schedules, and inventory levels.
- Business rules should be enforced to prevent overscheduling of drivers and driver assistants, ensuring compliance with work hour constraints.
- Orders should be assigned to the most suitable route based on factors like delivery location, capacity, and route availability.
- Customers should be restricted from placing orders that do not comply with the 7-day advance order placement rule.
- The system should enforce the allocation of train trips for order transportation to maintain efficient distribution.