Software Requirements Specification

Logistics System

Ege Türker
Beste Kaya
Ata Türker

INF 439 Software Engineering

Table of Contents

1.0. Introduction

1.1 Project Name & Description

The Logistics Management System is a desktop-based software application developed to manage and track shipment processes in a simple and structured manner. The system allows administrators to manage users and shipments, staff members to update delivery statuses during the distribution process, and public users to track shipments using a unique tracking number without authentication.

The main purpose of the system is to provide transparency, basic control, and ease of management for logistics operations. By centralizing shipment information and user roles in a single application, the system reduces manual tracking efforts and improves accessibility to shipment status information. The application is developed as a minimum viable product (MVP) and focuses on core logistics functionalities rather than advanced features.

1.2 Project Team Members and Roles

The Logistics Management System project is developed by a three-member team. Each team member contributed to different phases of the Software Development Life Cycle (SDLC), including requirements analysis, system design, and documentation.

- ***Ege Türker***

  Responsible for preparing the project documentation and UI mockups. Took the lead in writing the Software Requirements Specification (SRS) and designing the overall structure of the report.

- ***Beste Kaya***

  Responsible for the database design and user interface implementation. Designed the database schema and contributed to implementing the application’s user interfaces based on the defined requirements.

- ***Ata Türker***

  Responsible for the software implementation and coding tasks. Implemented the core functionalities of the system and integrated them with the database and user interface components.

All team members collaborated closely during the requirements analysis, system design, and decision-making processes. In addition, the implementation and coding tasks were carried out collaboratively, with team members supporting each other during development. Key architectural and design decisions were made jointly to ensure that the final system meets the project objectives and course requirement.

1.3 Problem Statement

Logistics and shipment tracking processes are often managed manually or through fragmented systems, making it difficult for organizations to track deliveries efficiently and provide accurate shipment information to customers. Lack of centralized control can lead to delayed updates, limited transparency, and increased operational complexity.

The problem addressed by this project is the absence of a simple, role-based logistics management system that allows different users to interact with shipment data according to their responsibilities. Administrators need a way to manage users and shipments, staff members require a controlled interface to update delivery statuses, and customers expect to track their shipments easily using a tracking number without creating an account.

The Logistics Management System aims to solve these issues by providing a centralized desktop application that supports user management, shipment tracking, and delivery status updates through clearly defined roles and use cases.

1.4 Purpose

The purpose of this document is to present a detailed description of the Logistics Management System. This document explains the purpose and main functionalities of the system, including shipment tracking, user management, and delivery status updates. It describes how different user roles such as admin, staff, and public users interact with the system.

The document outlines the functional and non-functional requirements, system constraints, and expected system behavior in response to user actions. This Software Requirements Specification (SRS) is intended for stakeholders, instructors, and developers involved in the design and implementation of the system. It serves as a reference document throughout the development process.

1.5 Scope of Project

The Logistics Management System is a desktop-based application developed using Windows Forms. The system is designed to manage shipment information, track delivery statuses, and support different user roles including admin, staff, and public users.

Admins can manage users and view all shipments, staff members can update shipment statuses during the delivery process, and public users can track shipments using a tracking number without authentication. The system focuses on basic logistics operations and does not include advanced features such as GPS tracking or automated notifications**.

1.6 Glossary

Term	Definition
Admin	User with full system management privileges
Staff	User responsible for shipment delivery operations
Customer (Public User)	User who tracks shipments using a tracking number
Shipment	A package registered in the logistics system
Tracking Number	Unique code used to track a shipment

1.7 References

- IEEE Std 830-1998 – IEEE Recommended Practice for Software Requirements Specifications
- Course materials for INF 439 Software Engineering
- Microsoft Documentation – Windows Forms

1.8 Overview of Document

This document is organized as follows:

Section 1 provides an introduction and defines the purpose and scope of the system. Section 2 describes the overall system functionalities and functional requirements. Section 3 presents detailed use case specifications. Section 4 outlines non-functional requirements. Section 5 provides a high-level system architecture overview. Section 6 describes assumptions and constraints, and Section 7 discusses possible future enhancements.

2.0 Overall Description

2.1 System Environment

The system is a Windows desktop application developed using C# Windows Forms and a relational database. It runs on Windows operating systems and supports multiple user roles with role-based access control.

2.2 Functional Requirements Specification

This section summarizes the functional capabilities of the system by user role. The functional requirements are organized based on system actors to clearly define responsibilities and system behavior.

2.2.1 Admin Use Cases

2.2.1.1. Admin Login - Allows the administrator to access the system using valid credentials.

2.2.1.2. Create New User - Allows the administrator to create new admin or staff accounts.

2.2.1.3. Delete User - Allows the administrator to remove users from the system.

2.2.1.4. View All Shipments - Allows the administrator to view all shipments in the system.

2.2.1.5. Update Shipment Status – Allows the administrator to update shipment statuses.

2.2.1.6. Logout – Allows the administrator to securely exit the system.

2.2.2 Customer (Public User) Use Cases

2.2.2.1 Access Shipment Tracking – Allows public users to access the tracking screen without login.

2.2.2.2 Track Shipment by Tracking Number – Allows users to track a shipment using a valid tracking number.

2.2.2.3 View Shipment Status – Allows users to view the current shipment status.

2.2.3 Staff Use Cases

2.2.3.1 Staff Login – Allows staff members to log into the system.

2.2.3.2 View Assigned Shipments – Allows staff to view shipments assigned to them.

2.2.3.3 Update Shipment Status – Allows staff to update delivery status information.

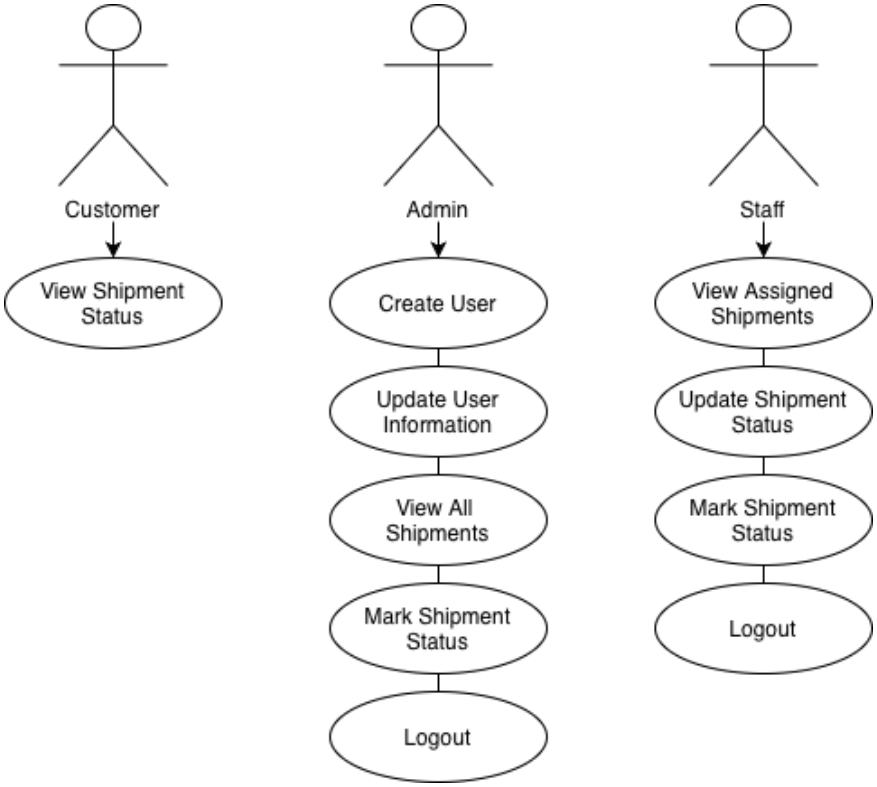
2.2.3.4 Mark Shipment as Out for Delivery – Allows staff to mark shipments as out for delivery.

2.2.3.5 Mark Shipment as Delivered – Allows staff to mark shipments as delivered.

2.2.3.6 View Delivery History – Allows staff to view completed deliveries.

2.2.3.7 Logout – Allows staff to securely exit the system.

2.2.4 Use Case Diagram



3.0 Use Case Specifications

3.1 Admin Use Case Specifications

3.1.1. Admin Login

- Actor: Admin
- Precondition: Admin account exists.
- Basic Flow: Admin enters credentials, system validates, dashboard is displayed.
- Alternative Flow: Invalid credentials result in an error message.
- Postcondition: Admin is logged in or an error is shown.

3.1.2 Create New User

- Actor: Admin
- Precondition: Admin is logged in.
- Basic Flow: Admin enters user details and saves; system creates the user.
- Postcondition: New user account is created.

3.1.3 Update User Information

- Actor: Admin
- Precondition: Target user exists.
- Basic Flow: Admin edits user details and saves changes.
- Postcondition: User information is updated.

3.1.4 Delete User

- Actor: Admin
- Precondition: Target user exists.
- Basic Flow: Admin deletes the user account.
- Postcondition: User is removed from the system.

3.1.5 View All Shipments

- Actor: Admin
- Precondition: Admin is logged in.
- Basic Flow: System displays all shipments.

3.1.6 Update Shipment Status

- Actor: Admin
- Precondition: Shipment exists.
- Basic Flow: Admin updates the shipment status.
- Postcondition: Shipment status is updated.

3.1.7 Logout

- Actor: Admin
- Basic Flow: Admin logs out and session ends.

3.2 Customer (Public User) Use Case Specifications

3.2.1 Access Shipment Tracking

- Actor: Customer
- Basic Flow: User opens the tracking screen.

3.2.2 Track Shipment by Tracking Number

- Actor: Customer
- Precondition: Valid tracking number exists.
- Basic Flow: User enters tracking number; system displays shipment information.
- Alternative Flow: Invalid number shows an error.

3.2.3 View Shipment Status

- Actor: Customer
- Basic Flow: System displays current shipment status.

3.2.4 Exit Tracking Screen

- Actor: Customer
- Basic Flow: User exits the tracking screen.

3.3 Staff Use Case Specifications

3.3.1 Staff Login

- Actor: Staff
- Basic Flow: Staff enters credentials and logs in.

3.3.2 Update Shipment Status

- Actor: Staff
- Basic Flow: Staff updates shipment status.

3.3.3 Mark Shipment as Out for Delivery

- Actor: Staff
- Precondition: Shipment is assigned to staff.
- Basic Flow: Staff marks shipment as out for delivery.

3.3.4 Mark Shipment as Delivered

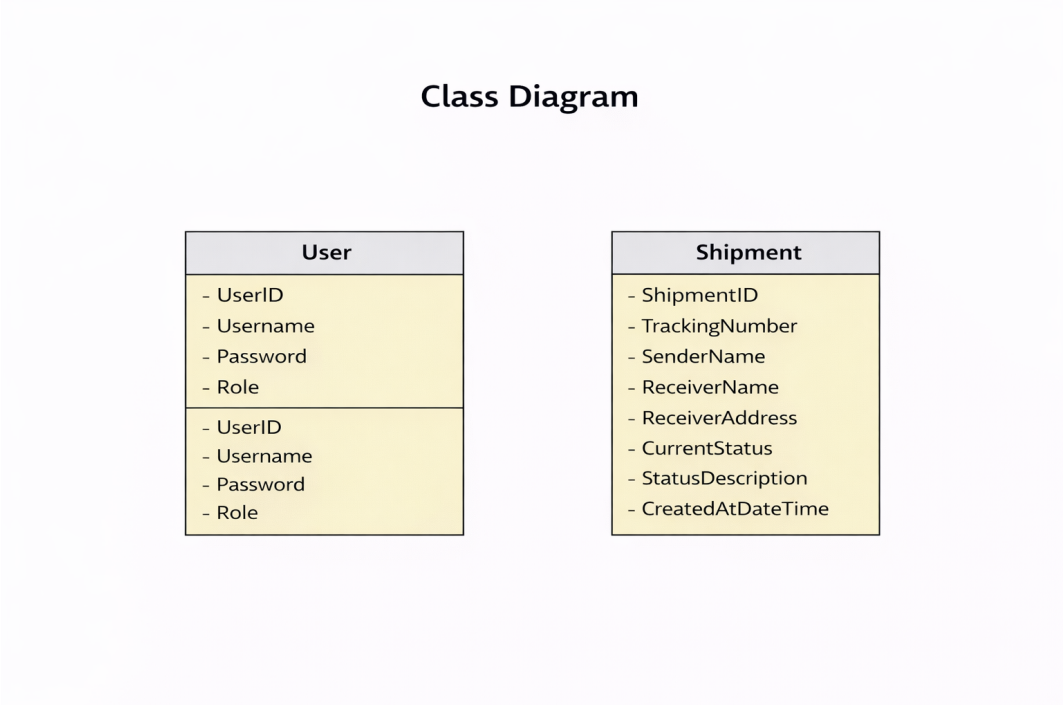
- Actor: Staff
- Precondition: Shipment is out for delivery.
- Basic Flow: Staff marks shipment as delivered.

3.3.5 Logout

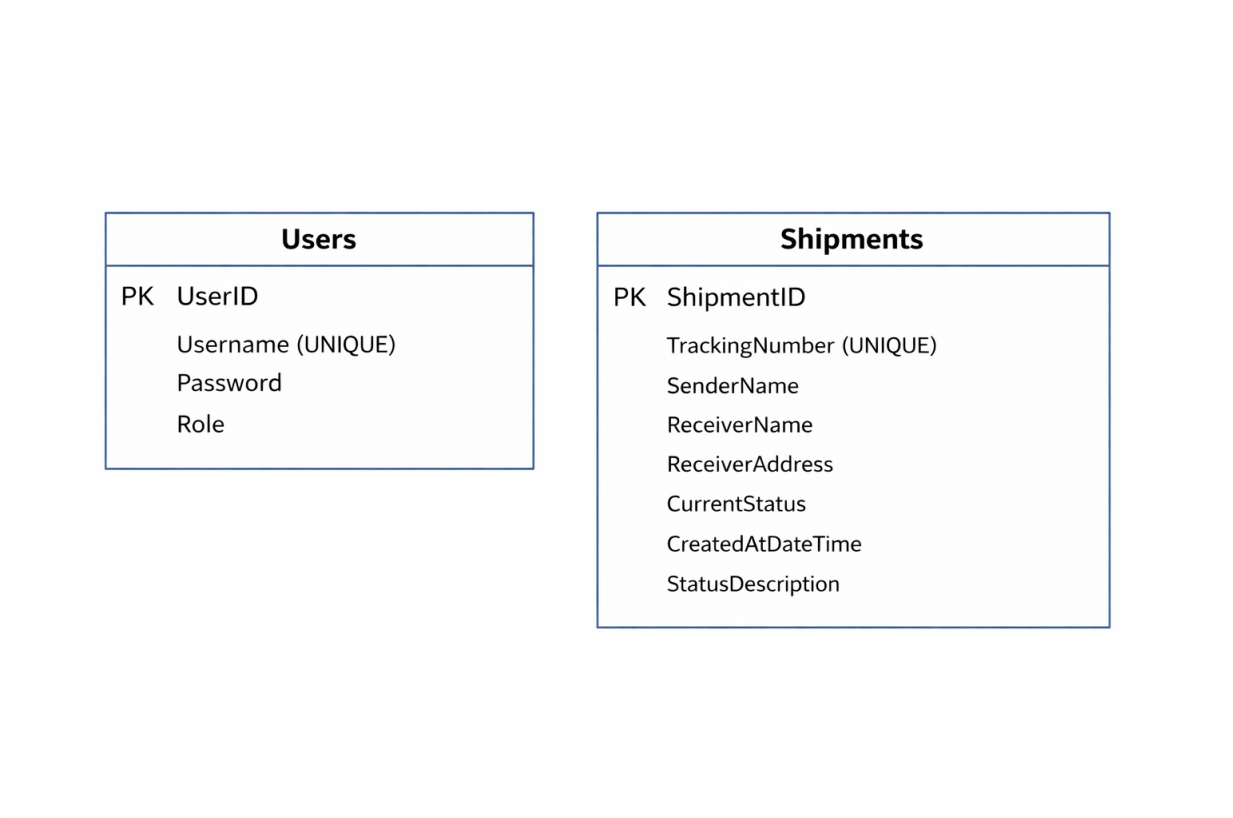
- Actor: Staff
- Basic Flow: Staff logs out.

4.0 System Design

4.1 Class Diagram

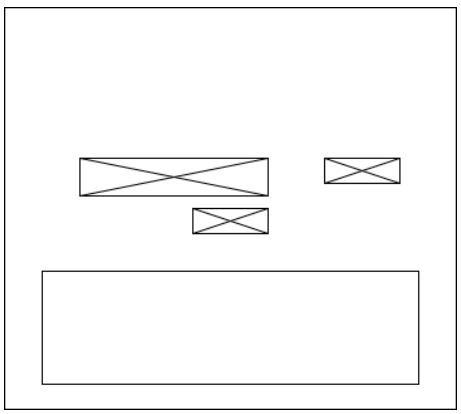


4.2 Database Design (ER Diagram)

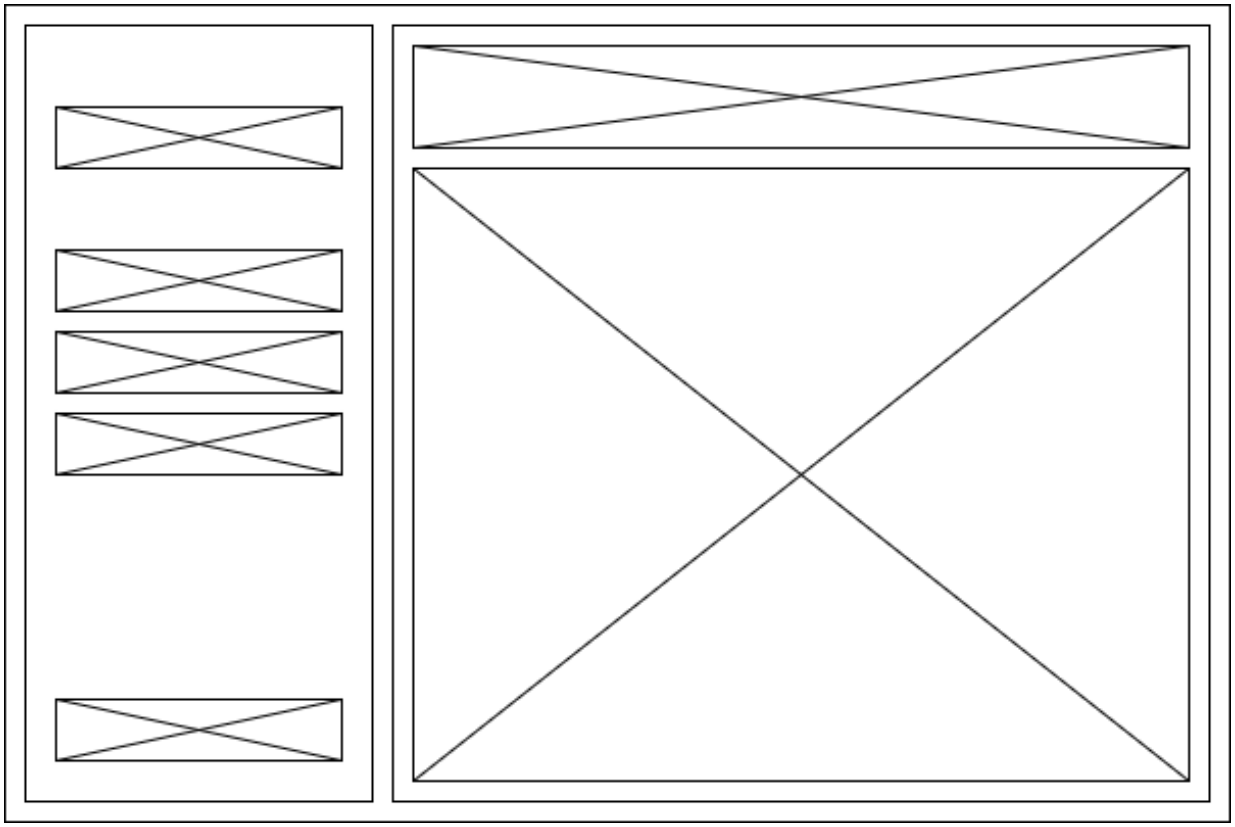


There isn’t any relation between tables.

4.3 UI Mockups



Track Screen



Admin and Staff dashboard, They have the same infrastructure we have used 2 separate panels for navigate faster.

5.0 Non-Functional Requirements

- Usability: The system shall be easy to use for non-technical users.
- Performance: Tracking results shall be displayed within 2 seconds.
- Security: Only authorized users may access restricted areas.
- Reliability: The system shall prevent data loss.
- Maintainability: The system shall support future enhancements.

6.0 Assumptions and Constraints

- Users have access to Windows OS.
- The system is developed using Windows Forms.

- The system uses mock data for demonstration purposes.

7.0 AI & Prompt Engineering Log

- ***Prompt 1***

  Task: Designing the database schema

  My Prompt:

  "I currently have two tables in my database: Users and Shipments. Do I need to define a relationship between these tables, or can they remain independent for an MVP logistics system?"

  AI Output Summary:

  The AI explained that a relationship between Users and Shipments is not mandatory if shipment records are not directly associated with a specific user. It suggested that relationships are only required when tracking ownership, assignment, or history.

  My Refinement:

  Based on the AI's explanation, I decided not to create a relationship between the Users and Shipments tables. This decision aligns with the MVP scope of the project, where user roles are used only for authorization purposes and shipment data is managed independently.

- ***Prompt 2***

  Task: Deciding which sections to include in the SRS document

  My Prompt:

  "I'm preparing an SRS document for my software engineering project and I'm a confused about the structure. Which main headings should I use while writing the SRS?"

  AI Output Summary:

  The AI suggested a standard SRS structure including an introduction, functional and non-functional requirements, use cases, and system constraints.

My Refinement:

I followed the suggested structure but kept the document simple and focused only on the main features required for the MVP, removing sections that were not necessary for the scope of the project.

- ***Prompt 3***

  Task: Designing the Class Diagram

  My Prompt:

  "I need to create a class diagram for my project, but I'm not really sure how to do it or what it should look like. Can you help me with that?"

  AI Output Summary:

  The AI explained what a class diagram is and suggested creating simple classes that represent the main entities of the system, such as users and shipments.

  My Refinement:

  I followed the guidance and created a simple class diagram that focuses on the main data structures of the system, keeping it aligned with the MVP scope of the project. However, I initially found it a bit confusing because the class diagram looked very similar to the database structure.

- ***Prompt 4***

  Task: Deciding how many forms are needed for the application

  My Prompt:

  "For a simple logistics management system, how many forms would you recommend creating? I want to keep the project simple but still functional."

  AI Output Summary:

  The AI suggested a relatively large number of forms, including separate screens for different

shipment actions and management tasks, which felt unnecessary for a simple MVP-level logistics system.

My Refinement:

We initially tried to follow the form structure suggested by the AI, but this approach ended up complicating the project and making the system harder to manage. This experience helped us realize that leaving architectural decisions entirely to AI can sometimes increase complexity instead of reducing it. After recognizing this issue, we stepped back and redesigned the software architecture ourselves, focusing on simplicity and the actual needs of the MVP.

- ***Prompt 5***

  Task: Retrieving data from the database

  My Prompt:

  "While working with the database, I needed help writing SELECT queries to retrieve user and shipment data. Can you help me with simple queries for fetching shipment status and basic user information?"

  AI Output Summary:

  The AI suggested various SELECT queries, including additional conditions and filtering options. Some of these suggestions were more detailed than necessary for the simple structure of the current database.

  My Refinement:

  I kept the queries very simple and used only basic SELECT statements that directly retrieve shipment status and user information from single tables, since the database does not include relational joins and the project follows an MVP approach.

8.0 Future Enhancements

Possible future enhancements include mobile support, notification services, GPS tracking, and advanced reporting features.