Software Requirements Specification Template

Warehouse Wizard

Software Requirements Specification

Version

05/24/2023

Mounika Annabathina

Amani Chennu

Pradeep RaoVeeramaneni

Aravind Kumar Koyyala

Anusha Bichal

Swarnalatha Yadla

Submitted in partial fulfilment.

Of the requirements of

CSIS 44-691 Graduate Directed Project 1

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Description** | **Author** | **Comments** |
| <date> | <Version 1> | <Your Name> | <First Revision> |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Document Approval

The following Software Requirements Specification has been accepted and approved by the following:

|  |  |  |  |
| --- | --- | --- | --- |
| **Signature** | **Printed Name** | **Title** | **Date** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Table of Contents**

**Table of Contents Page Number**

Table of Contents Page Number

1.Introduction

1.1. Purpose

1.2. Scope

1.3. Definitions, Acronyms, and Abbreviations

1.4. References

1.5. Overview

2.General Description

2.1. Product Perspective

2.2. Product Functions

2.3. User Characteristics

2.4. General Constraints

2.5. Assumptions and Dependencies

3.Specific Requirements

3.1. External Interface Requirements

3.1.1. User Interfaces

3.1.2. Hardware Interfaces

3.1.3. Software Interfaces

3.1.4. Communications Interface

3.2. Functional Requirements

3.3. Use Cases

3.4. Class/Objects

3.5. Non-Functional Requirements

3.5.1. Performance

3.5.2. Reliability

3.5.3. Availability

3.5.4. Security

<Project Name>

Software Requirements Specification Page iv

3.5.5. Portability

3.6. Inverse Requirements

3.7. Design Constraints

3.8. Logical Database Requirements

3.9. Other Requirements

3.10. Prototypes (for complete project)

3.11. Use Case Diagrams

4. Design

4.1. ER diagram.

4.2. GUI

5.Analysis Models

5.1. Data Flow Diagram

5.2. Sequence Diagram

**1. Introduction**

1.1. Purpose: The purpose of the warehouse management system application is to efficiently manage and track the movement of goods within a warehouse. It facilitates communication and coordination between the warehouse owner, who owns the warehouse, and customers who own products and need to apply for check-out of their items. The system aims to streamline warehouse operations, improve inventory control, and provide an efficient process for managing customer requests.

1.2. Scope: The scope of the application includes:

* Warehouse Owner Management: Storing and managing information about warehouse owners, including contact details, warehouse locations, and rental agreements.
* Customer Management: Storing and managing customer information, including contact details, addresses, and the products owned by customers.
* Request Management: Allowing customers to apply for check-out of their products based on a tracking ID, tracking the status of requests, and facilitating communication between the warehouse owner and the customer.
* Reporting and Analytics: Generating reports and providing insights into warehouse utilization, request history, and customer activity.

1.3. Definitions, Acronyms, and Abbreviations:

* ASP.NET: Active Server Pages .NET, a web development framework.
* Entity Framework: A data access technology in ASP.NET for working with databases using object-oriented programming.
* Warehouse Management System: A software application designed to manage and control warehouse operations.
* Warehouse Owner: The individual or organization that owns a warehouse.
* Customer: An individual or organization that owns products and stores them in a warehouse.
* Check-in: The process of bringing products into the warehouse and registering them.
* Check-out: The process of taking products out of the warehouse.
* Tracking ID: A unique identifier assigned to each product for tracking purposes.

1.4. References:

1. ASP.NET Official Website - Introduction to ASP.NET:
   * Link: <https://dotnet.microsoft.com/apps/aspnet>
2. Microsoft Docs - Entity Framework Core:
   * Link: <https://docs.microsoft.com/en-us/ef/core/>
3. WiseTech Global - What is a Warehouse Management System (WMS)?
   * Link: <https://www.wisetechglobal.com/products/warehouse-management-system>
4. Investopedia - Warehouse Owner:
   * Link: <https://www.investopedia.com/terms/w/warehouse-owner.asp>
5. BusinessDictionary - Customer:
   * Link: <http://www.businessdictionary.com/definition/customer.html>
6. Techopedia - Check-In:
   * Link: <https://www.techopedia.com/definition/27013/check-in>

.

1.5. Overview: The warehouse management system application is developed using ASP.NET and utilizes the Entity Framework for database operations. It provides a user-friendly interface for warehouse owners and customers to manage their respective activities related to warehouse operations. The application integrates two databases: the owner database for managing warehouse owner information and the customer database for storing customer details and their owned products.

**2. General Description**

**2.1. Product Perspective**

The warehouse management system is a web application that serves as a central platform for managing warehouse operations and facilitating communication between warehouse owners and customers. It is built using ASP.NET and utilizes the Entity Framework for database operations. The system interacts with two databases: the owner database, which stores information about warehouse owners, and the customer database, which manages customer details and their owned products. The system provides a user-friendly interface for warehouse owners and customers to perform their respective tasks.

**2.2. Product Functions**

The warehouse management system offers the following key functions:

1. Warehouse Owner Management:
   * Updating owner details such as contact information and rental agreements.
   * Managing warehouse locations and their availability.
   * Tracking warehouse occupancy and availability.
2. Customer Management:
   * Storing and managing customer information including contact details, addresses, and products owned.
   * Allowing customers to register and update their profiles.
   * Managing customer permissions and access levels.
3. Request Management:
   * Allowing customers to apply for check-out of their products based on a tracking ID.
   * Tracking the status of requests and notifying customers and warehouse owners about the progress.
   * Facilitating communication between customers and warehouse owners regarding requests.
4. Inventory Management:
   * Tracking the inventory of products stored in the warehouse.
   * Managing product details such as name, quantity, and specifications.
   * Updating inventory status upon check-in and check-out operations.
5. Reporting and Analytics:
   * Generating reports on warehouse utilization, occupancy and product inventory.
   * Providing insights into customer activity, including check-in and check-out history.
   * Analyzing data to identify trends and optimize warehouse operations.

**2.3. User Characteristics**

The warehouse management system caters to two primary user roles:

1. Warehouse Owners:
   * Owners of warehouses who need to manage their properties, rental agreements, and occupancy.
   * They have the authority to approve customer requests for check-out.
   * They can view reports and analytics related to their warehouse's utilization.
2. Customers:
   * Individuals or businesses who own products stored in the warehouse.
   * They can submit requests for check-out based on a tracking ID.
   * They can view the status of their requests and communicate with the warehouse owner regarding their products.

**2.4. General Constraints**

The warehouse management system has the following general constraints:

1. Technical Requirements:
   * The application is developed using ASP.NET, utilizing the Entity Framework for database operations.
   * It requires a web server and a compatible database server for hosting the application.
   * The system should be accessible through supported web browsers.
2. Security Measures:
   * The system should implement secure user authentication and authorization mechanisms to ensure data privacy and prevent unauthorized access.
   * Sensitive information such as customer and owner data should be encrypted and stored securely.
3. Performance Optimization:
   * The application should be optimized for performance to handle a large number of concurrent users and database operations.
   * Database indexing and caching techniques can be employed to enhance system responsiveness.

**2.5. Assumptions and Dependencies**

The warehouse management system operates under the following assumptions and dependencies:

1. Reliability of Server Infrastructure:
   * The system assumes the availability and reliability of the hosting server infrastructure where the application and databases are deployed.
   * Server maintenance and backup procedures should be in place to ensure uninterrupted operation.
2. Entity-Framework package to design database modal classes.

**3. Specific Requirements**

**3.1. External Interface Requirements**

3.1.1. User Interfaces

The user interfaces should be intuitive, user-friendly, and responsive. The application should provide separate interfaces for warehouse owners and customers. Key features of the user interfaces include:

* Login page with authorization and authentication screens for both warehouse owners and customers.
* Dashboard for warehouse owners to manage their warehouses, rental agreements, and view reports.
* Customer portal for customers to submit check-in or check-out requests, track their requests, and communicate with warehouse owners.
* Forms for entering and updating owner and customer information.

3.1.2. Hardware Interfaces

The application should be compatible with standard hardware components, including computers, laptops, tablets, and smartphones, with a compatible web browser.

3.1.3. Software Interfaces

The application should integrate with the following software components:

* ASP.NET framework for web application development.
* Entity Framework for database operations.
* Database management system (e.g., MSSQL Server) for storing owner and customer data.
* Communication libraries for sending notifications and emails.

3.1.4. Communications Interface

The application should support communication channels between warehouse owners and customers, such as email notifications and in-app messaging.

**3.2. Functional Requirements**

The functional requirements describe the specific functions and capabilities of the warehouse management system, including:

* Warehouse owner management: updating, and deleting warehouse owner information.
* Customer management: Adding, updating, and deleting customer information, including their owned products.
* Request management: Allowing customers to submit check-in or check-out requests based on tracking IDs, tracking the status of requests, and facilitating communication between customers and warehouse owners.
* Inventory management: Tracking and updating the inventory of products stored in the warehouse.
* Reporting and analytics: Generating reports on warehouse utilization, customer activity, and inventory status.

**3.3. Use Cases**

Use cases illustrate the interactions and workflows between system actors. Some use cases for the warehouse management system may include:

* Customer checks out a product.
* Warehouse owner generates a warehouse utilization report or history for check in and checkout requests.

**3.4. Class/Objects**

The class/objects represent the entities and data structures used in the system. Some classes/objects in the warehouse management system may include:

* Warehouse Owner: Represents a warehouse owner with attributes like name, contact information, and rental agreements.
* Customer: Represents a customer with attributes like name, contact information, and owned products.

Include Products info: Represents a product with attributes like name, quantity, UPC and tracking ID etc.

**3.5. Non-Functional Requirements**

3.5.1. Performance

The system should be able to handle a large number of concurrent users and database operations efficiently. It should provide fast response times for user interactions and data retrieval.

3.5.2. Reliability

The system should be reliable and available for use during normal business hours. It should minimize downtime and handle errors gracefully.

3.5.3. Availability

The system should be available to users with a high uptime percentage. Planned maintenance should be communicated to users in advance.

3.5.4. Security

The system should implement appropriate security measures to protect user data and prevent unauthorized access. This includes secure user authentication, data encryption, and access control.

3.5.5. Portability

The system should be designed and developed to be easily deployable on different platforms and environments. It should support different web browsers and devices.

**3.6. Inverse Requirements**

Inverse requirements represent the actions or behaviours that the system should not exhibit. Some inverse requirements for the warehouse management system may include:

* The system should not allow unauthorized access to sensitive data.
* The system should not perform operations that exceed the allocated system resources.
* The system should not allow conflicting check-in or check-out requests for the same warehouse space.

**3.7. Design Constraints**

Design constraints are limitations or restrictions that impact the system design. Some design constraints for the warehouse management system may include:

* Compliance with industry regulations and standards related to data security and privacy.
* Compatibility with the chosen ASP.NET and Entity Framework versions.
* Consideration of scalability to accommodate future growth and increasing data volumes.

**3.8. Logical Database Requirements**

The logical database requirements define the structure and organization of the databases used in the system. For the warehouse management system, the logical database requirements may include:

* Warehouse Owner Database:
  + Owner table: Stores information about warehouse owners, including their ID, name, contact details, and rental agreements.
  + Warehouse table: Stores information about warehouses, including their ID, location, and availability status.
* Customer Database:
  + Customer table: Stores information about customers, including their ID, name, contact details, and ownership information.
  + Product table: Stores information about products, including their ID, name, quantity, and tracking ID.

**3.9. Other Requirements**

Other requirements refer to any additional specifications or considerations for the warehouse management system. Some other requirements may include:

* Compliance with data backup and recovery procedures to ensure data integrity and availability.
* Support for multi-language and multi-currency functionalities, if required.
* Integration with third-party services or APIs for features like email notifications or payment gateways.

**3.10. Prototypes (for complete project)**

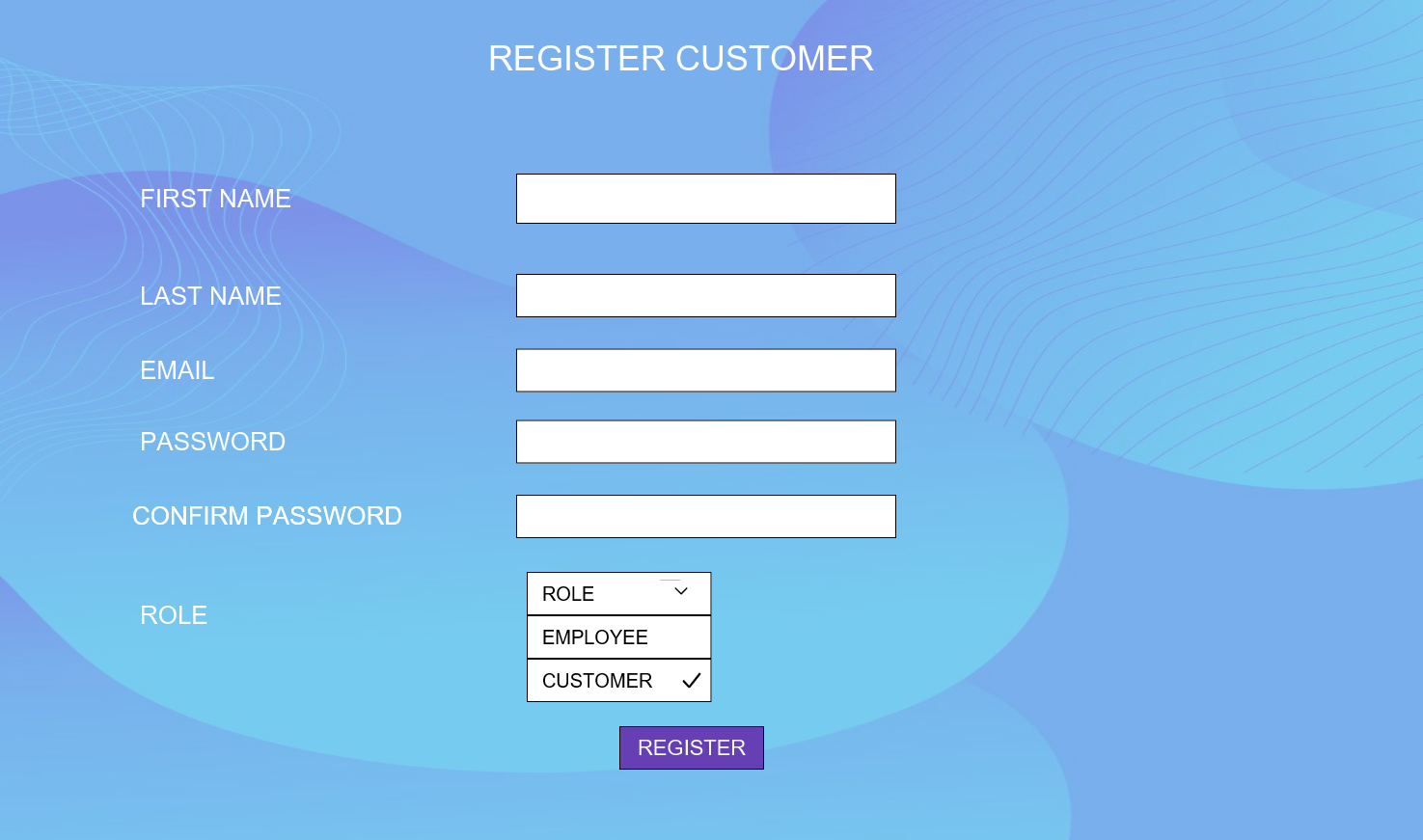
Screen 1



Screen 2

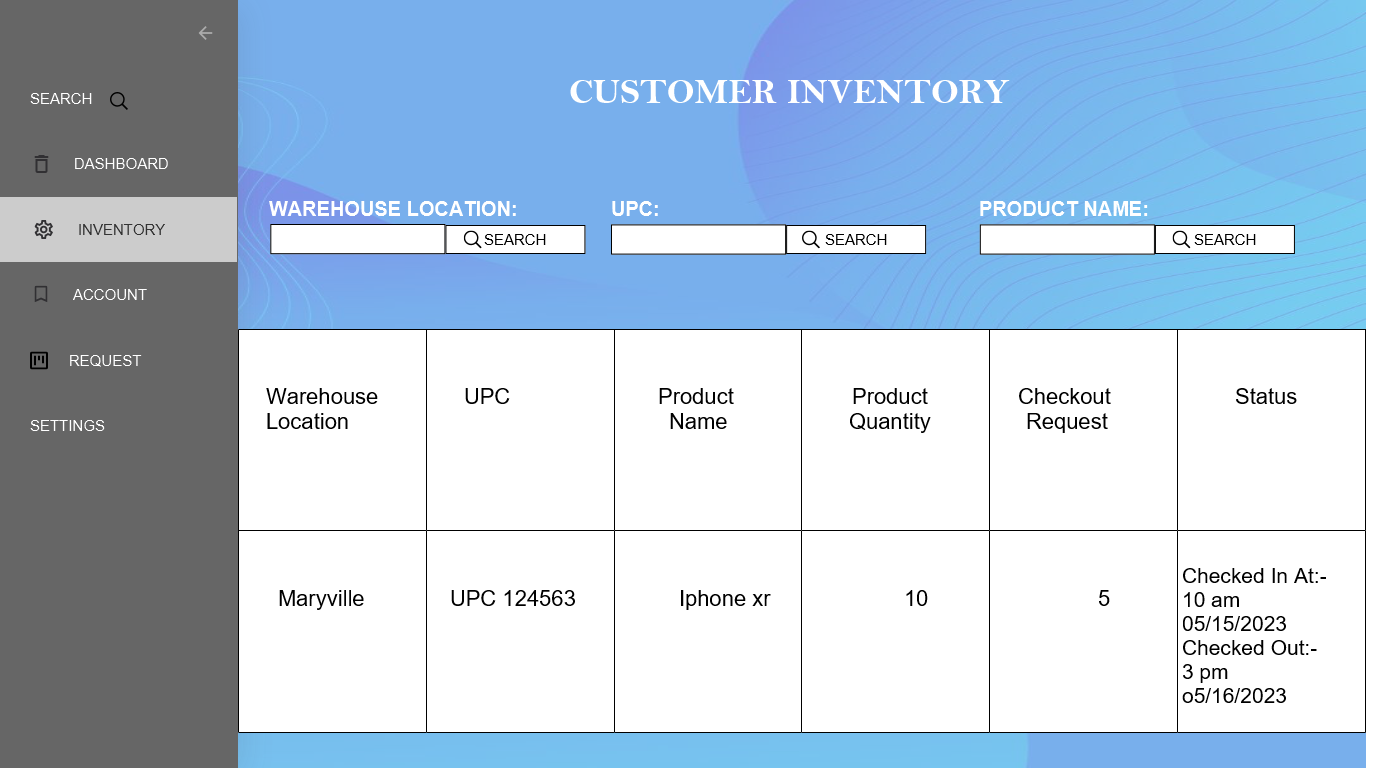


Screen 3



The Screen represents Role based Registration. Both Employee and customer can register here.

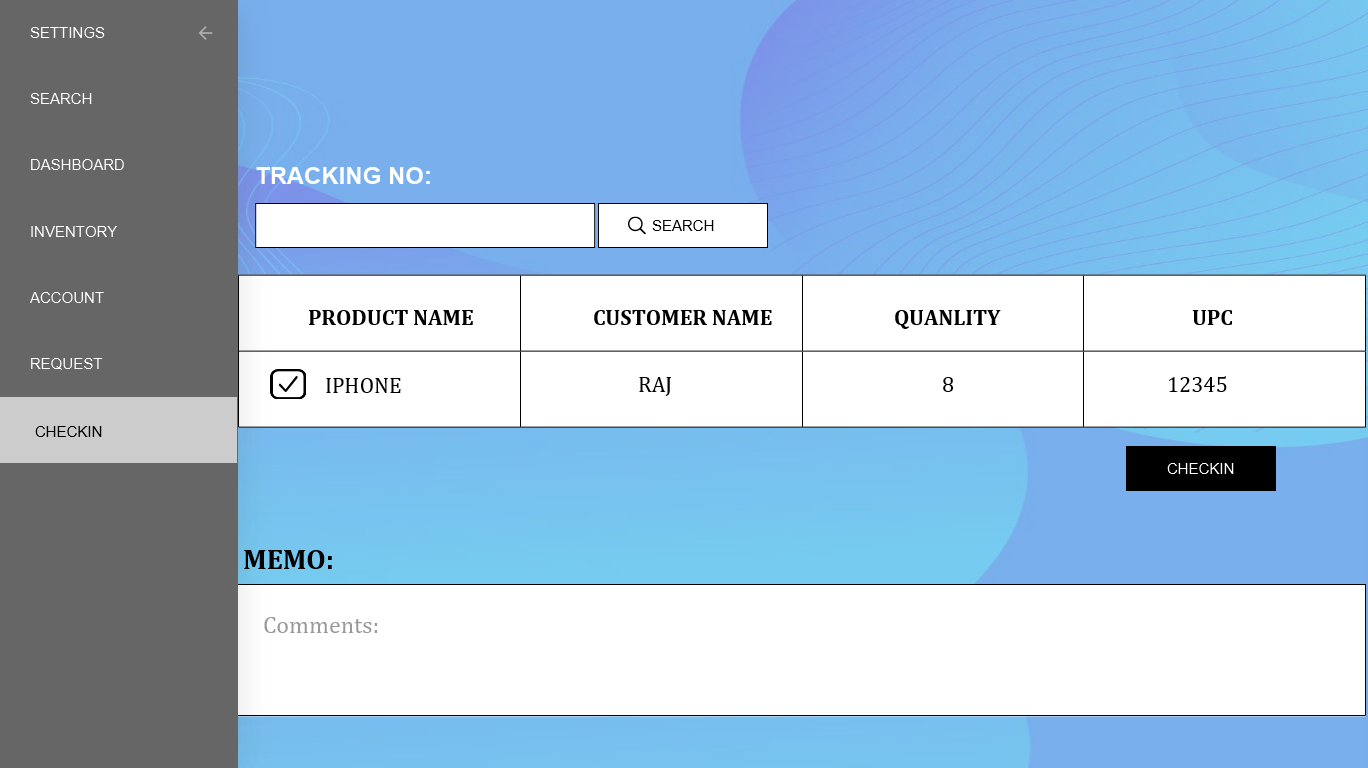
Screen 4(Represents Customer Inventory and allows Customer to put checkout request)



Screen 5(This Owner Inventory Screen Gives access to owner to check the list of customer Product details)



Screen 6(This screen shows how owner /Employee of the warehouse can check in the customers product when received)



Screen 7(This screen shows history of the checked out products of the customers to the owner )



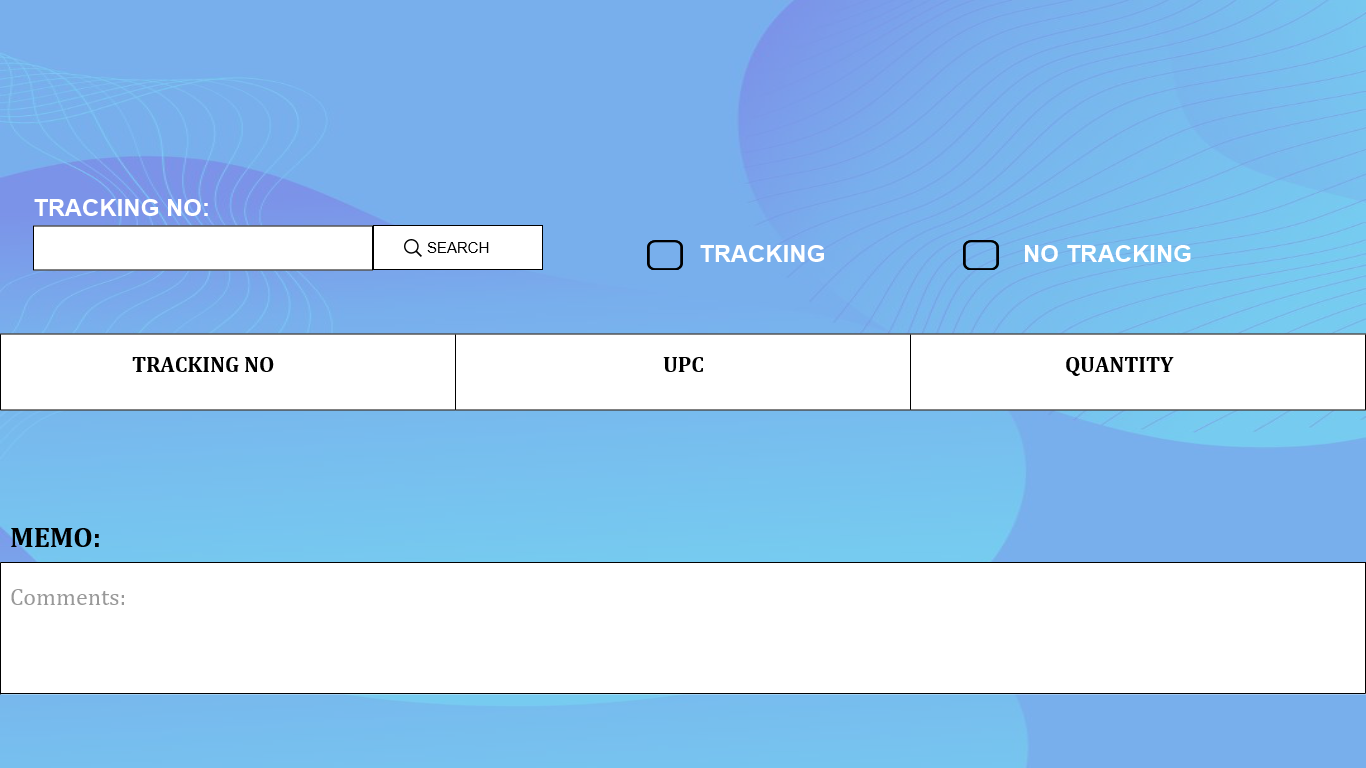
Screen 8(This screen shows Customer Checkout request which shows Pending/Processed requests)



Screen 9(This screen shows \*pending requests that customer requested for checkout)



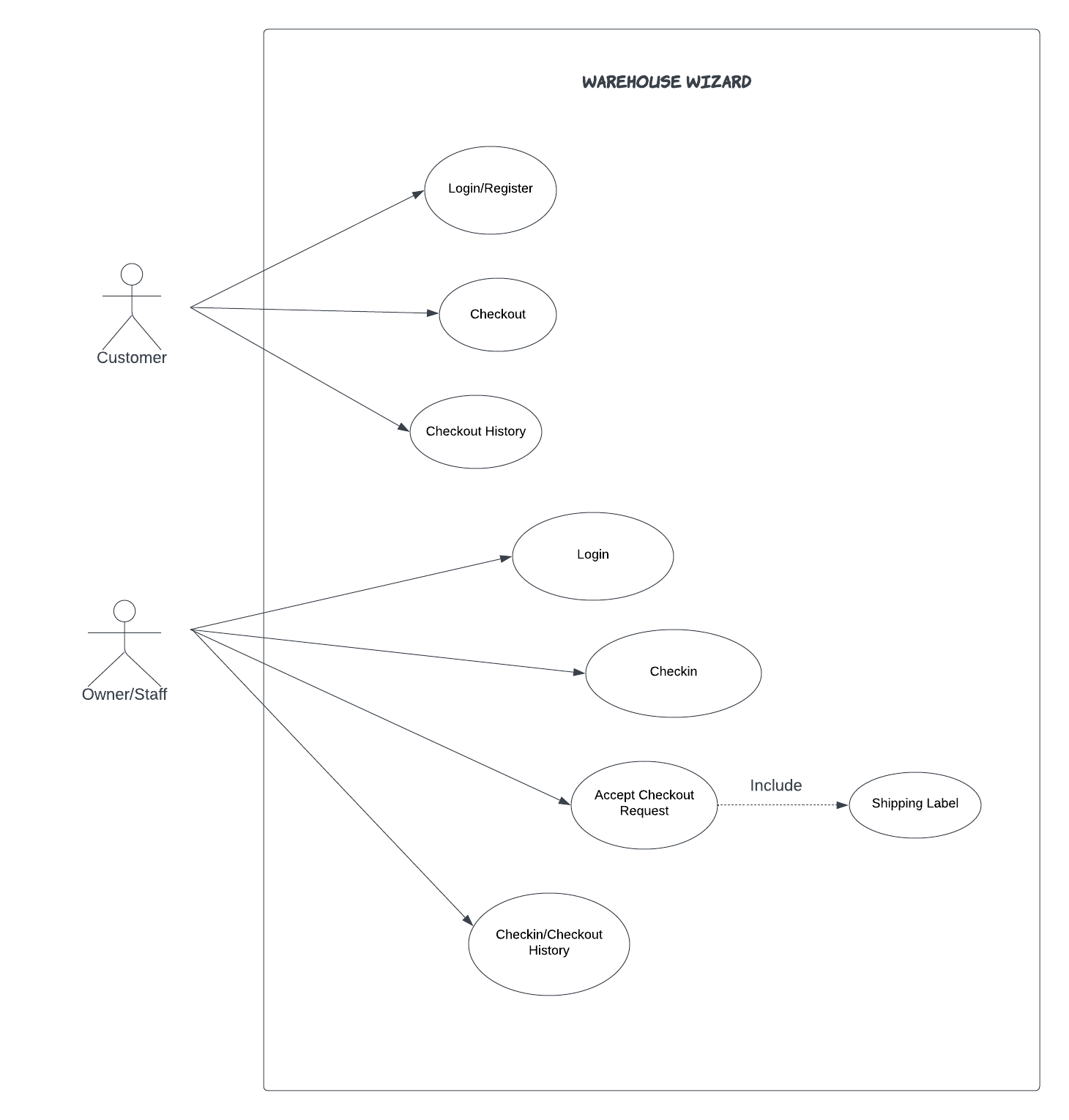
**Screen 10(This screen shows pending requests when clicked on UPC link which gives track of the customer product and allows the search function as well)**



**Screen 11(This screen shows history of the Requests that are processed that customer requested for checkout)**



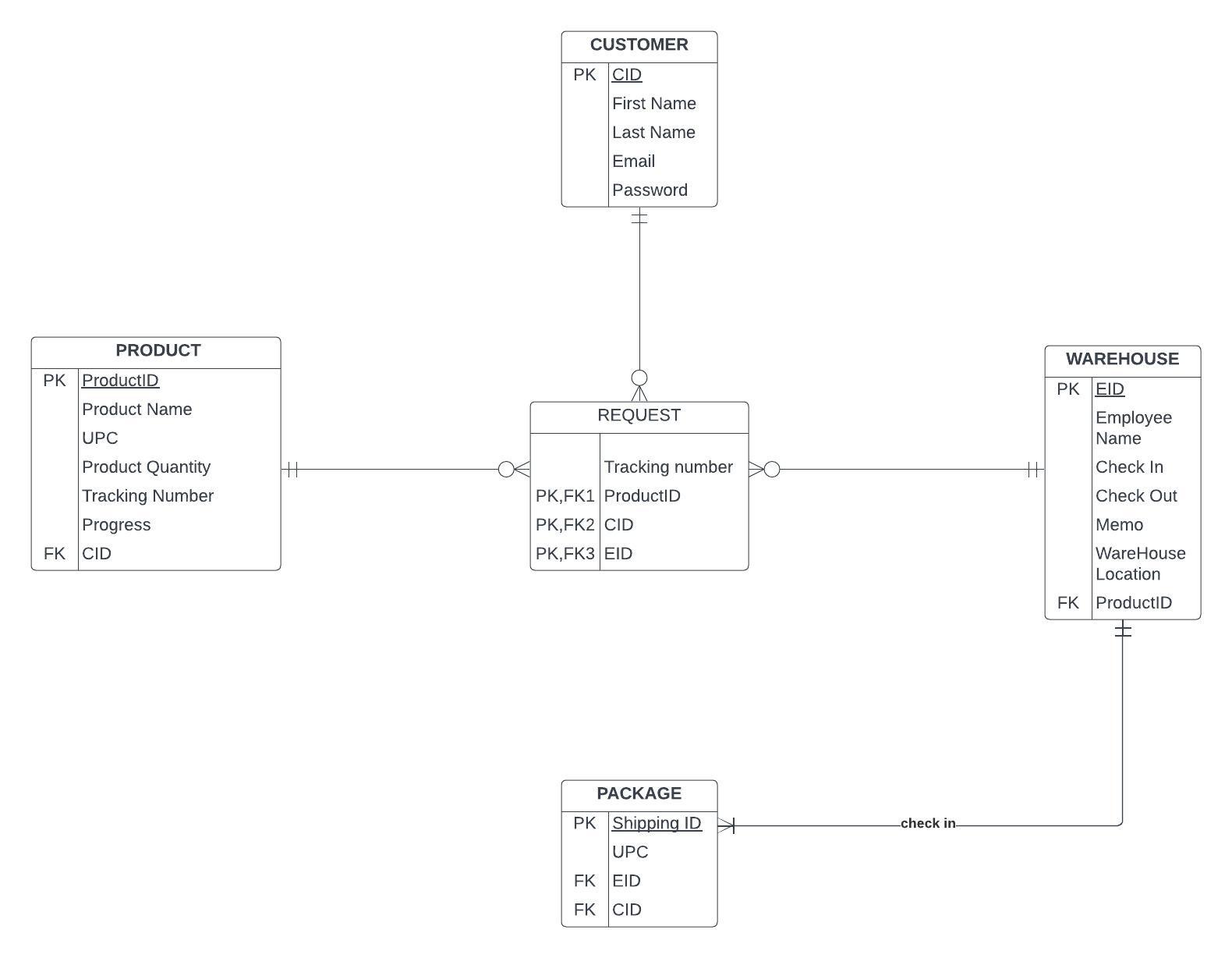
**3.11. Use Case Diagrams**



**4. Design**

## 4.1. ER diagram.

**WAREHOUSE MANAGEMENT SYSTEM ENTITY RELATIONSHIP DIAGRAM**



**5.Analysis Models**

**5.1. Data Flow Diagram**



**5.2. Sequence Diagram**

