Warehouse Management System (WMS)

CSC490 – Senior Project I

Fall 2024

Project Requirements

10/9/2024

Instructor Comments/Evaluation

Table of Contents

Abstract

The Warehouse Management System intends to help warehouse managers always track and manage their inventory; this includes ongoing shipments, deliveries, and supplies within the warehouse. While existing systems are available, they are not as affordable or rely on manual or outdated systems to track stock, leading to inefficiencies, errors, and increased operational costs. The team aims to develop a warehouse management system to simplify inventory tracking and management. It will focus on user-friendliness, compatibility, and practical use by warehouse gh/managers and workers. Implementing this system will allow these businesses to streamline operations, reduce errors, and increase productivity. In this document, we will show the development management systems and the project plan.

Introduction

Background:

WMS is crucial in the broader supply chain management process by ensuring that warehousing activities are optimized and efficient. With these systems in place, managers can track their inventory for business since products are constantly moving by being brought in, sent out, or returned. For example, let us say that you have a warehouse of inventory where a product is being sold, returned, and delivered. How would you know when a product is low, if the product has been returned, or when products were shipped and their location in the warehouse?

A system needs to be implemented to keep track of these things because it allows businesses to reduce errors and increase productivity.

Objective & Overview of Project:

The objective of our WMS is to create a system with a user-friendly interface, simplify management, and is compatible with most devices. This creates an intuitive, responsive design for warehouse managers and workers, minimizing the learning curve. It will include QR and Barcode scanning for easy product management and auditing. The project will utilize Bootstrap as the front-end framework to build a responsive interface, and AJAX and JavaScript will be used to improve user interaction. For our simplified management, a streamlined warehouse operation will be implemented, introducing role-based access to ensure data integrity and secure management of products. This project will be able to handle automating inventory updates, stock management, and reporting. Additionally, the system will contain compatibility features that ensure cross-platform compatibility. These features can be achieved using Bootstrap to create a responsive design that works across desktop and mobile, as well as by developing a Docker environment to simplify deployment and ensure consistent performance across different systems. All WMS data must be stored, so backend development with PHP will use object-oriented features for modular design and maintainability, utilizing MySQL for data storage and retrieval.

Team Details & Dynamics:

Everyone on this team plays a crucial role in developing this project. To ensure success, members selected the leadership role on the phases they felt most comfortable in, had experience with, and related to their major. These roles and members' majors will be displayed in the table

below. Allowing members to choose leadership ensures that our final project will be of the best quality. Each leader of a respective phase will ensure that all tasks are completed correctly and on time. Members can ask for help from other members about their phase if needed, which makes for a more cohesive work environment. project will be of the best quality. Each leader of a respective phase will be responsible for ensuring that all tasks are completed correctly and on time. Members can ask for help from other members about their phase if needed, which makes for a more cohesive work environment.

Team Member:	Major:	Leader of Phases:
Brian Colditz	CS	Implementation/Proposal
Shakear Wilson	CS	Requirements /Proposal
Ty Kress	CS & Mathematics	Analysis /Proposal
William Serowik	CS	Design /Proposal

To complete this project, the team will need an easy way to communicate and to keep things organized. An established Discord server will be our primary means of communication and meetings, and another application called GitHub Desktop will utilize Git to share and publish code in a private repository on GitHub. The team meetings on Discord are scheduled around members' schedules depending on the task at hand, but we are expected to be able to meet weekly.

Application Domain

Project content:

Our WMS project intends to be user-friendly and affordable for local and small businesses to help manage their inventory. The cost of a WMS varies based on the business size, as smaller businesses tend to spend between $500 - $1000 per facility every month or around $100 - $200 per user every month. For many businesses, a WMS is a significant investment but can lead to savings by improving efficiency and reducing errors in warehouse operations. With

this in mind, we will work towards a management system that provides supplier and purchase order management, inventory tracking and updates, and pinpoints for product returns.

Initial Business Model

Operational Environment:

This WMS is to create a more efficient, organized, and responsive warehouse operations using applications like bootstrap, php, MySQL and JavaScript. It achieves this by optimizing processes like inventory tracking, order fulfillment, space utilization and shipment management. By maintaining real-time records and automating repetitive tasks, reducing errors will enhance overall warehouse performance. This will ensure that companies can manage their supply chains more effectively and stay competitive in the market.

Description of Data Sources:

The Warehouse Management System will have its data created by users inputting inventory data such as incoming products, shipments that are coming in or out, and inventory that is being shuffled around in a warehouse. WMS relies heavily on user input to maintain accuracy in all warehouse activities. This input is crucial for tracking and managing inventory shipments and the movement of products within the warehouse. This enables the WMS to optimize operations. Improve efficiency, reduce errors and support better decision-making for managers and stakeholders.

Use Case UML Diagrams and Descriptions:

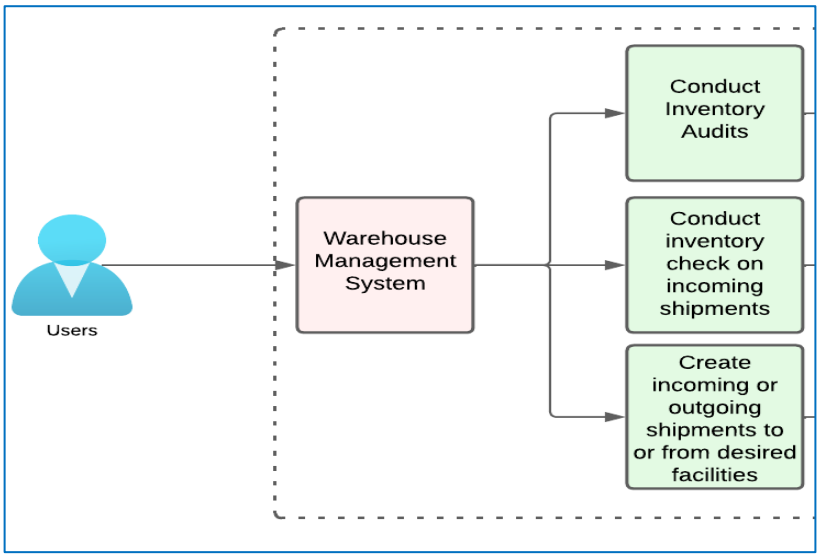

Figure 1: Use case UML for the user available features in the WMS

Description: When developing this project, the user base will need access to all key operations involved in managing a warehouse from audits to shipment handling and notifications. The use of bootstrap will be implemented here for our user interface.

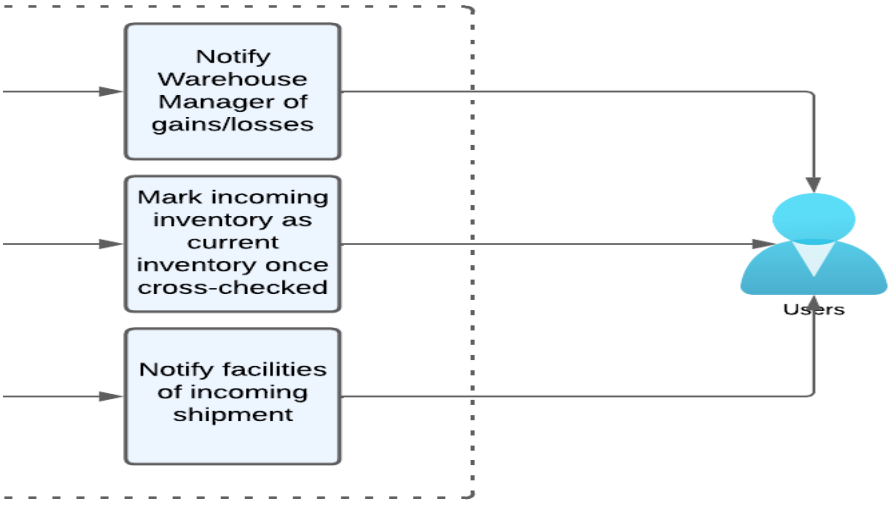

Figure 2: Use case UML diagram for representing system response to user

Description: The system now notifies other users that something has changed in the system which is represented by using arrows. This ensures the smooth operation of warehouse logistics. Information about changes in inventory helps maintain accurate records across the warehouse.

Initial Requirements

Functional:

A WMS requires certain key functional features to ensure smooth warehouse operations. These functionalities include tracking, control and management of inventory, and productivity. In this project inventory management will be a real-time inventory tracking which will have the ability to track each inventory level, its location and movements. It will also include cycle counting and SKU which support managing expiration dates and ensures accuracy without stopping operations. To take care of this data we'll implement MySQL and backend development with PHP, which will take care of all our data storage and retrieval. In addition to tracking, and managing warehouse zones, bins and other storage locations. In this project an interface is required to be able to display data and other features that are available in this project. We'll be using a template for our user-friendly interface whose features include an audit system to review stock levels, real-time stock level tracking and a display of supplier's details, purchase history and payment statues. Also, with the use of Bootstrap for our interface it's able to create, view, and update purchase orders. Since this project is going to be available on apps and websites the utilization of AJAX and JavaScript is needed to improve user interaction without full page reloads. Lastly, the use of Docker environment is required to simplify deployment and ensure consistent performance across different systems. These functional requirements ensure efficient warehouse operations from inventory management to shipping.

Nonfunctional:

This project has some non-functional requirements. This includes a template, scalability, and the overall performance of the system. A template with be used as the base of our interface to display data which is more professional. The WMS must respond quick to user actions in real-time so, low latency will be necessary for inventory updates and task assignments, even under heavy loads. In addition, this system will have to handle growing business so it must support expanding inventory levels, users, and operational workflows without performance degradation. Even though there aren't many nonfunctional, they are essential in ensuring the WMS operates efficiently while warehouses start to grow and handle more sensitive data.

Documentation:

I) Proposal Document

- Initial document used to propose and describe the project.

II) Requirement Document

- A document that communicates the vision of the project and establishes a development plan.

III) Specification Document

- A document that outlines design, testing and functionality of the system.

IV) Design Document

- A document that contains a detailed plan that will outline how the system will be built, this is based on the requirement and specification documents.

V) Project Log

- A document that provides information about the ongoing progress, activities, issues and lifecycle of the project

Testing / Revisions:

This document was created through the efforts of the group members. The document was being shared online allowing each member to be given access to edits to the document. The document was completed in sections with a rough draft being made in subsections. Revisions were saved to the cloud allowing for the most up to date document. For software we'll be using a docker environment to store all versions of our software and the updates made along with its creation. This ensures that if we have issues later in the development process, we can backtrack with the different versions saved. These saved versions will be versions of the program that implemented new features that execute properly to ensure that we have quality programming. As for hardware testing, this will be tested by individual members who have all components of the project.

List of References:

What is a warehouse management system (WMS)? Retrieved October 12, 2024, from https://www.sap.com/products/scm/extended-warehouse-management/what-is-a-wms.html

10 Warehouse Management System functional requirements. Retrieved October 12,2024 from https://hy-tek.com/resources/wms-functional-requirements/

Warehouse Management System Management System Requirements/WMS Requirements. Retrieved October 12,2024, from https://hy-tek.com/resources/wms-functional-requirements/

Essential WMS requirements for ecommerce Retrieved October 12, 2024, from https://www.shipbob.com/blog/wms-requirements/

Appendix: Technical Glossary

WMS (warehouse management system) - a software application that helps companies manage their warehouses operations.

AJAX - stands for Asynchronous JavaScript and XML. It is a set of web development techniques using various technologies to create dynamic web applications

JavaScript – an object-oriented computer programming language commonly used to create interactive effects within web browsers.

Bootstrap – a front-end framework for building responsive, mobile-first websites using pre-designed CSS, JavaScript, and HTML components.

PHP – a free, open-source scripting language used to create dynamic web pages and applications.

MySQL – an open-source relational database management system that uses Structured Query Language (SQL) to store, retrieve and manipulate data.

Docker environment – a lightweight containerized platform that allows developers to create, deploy, and run applications in isolated environments, ensuring consistency across different systems.

Template – a form, mold or pattern used as a guide to make something.

Interface – a physical connection between two computer systems, conversational syntax, a format for logical messages passed between the systems, and data – encoding structure understood by both systems.

SKU (stock keeping unit) - a unique alphanumeric code that retailers use to identify and track products.

QR (Quick response) - a two-dimensional barcode that can be scanned with a smartphone to access information.

Appendix: Team details

This document was written in cooperation with all team members over a course of a couple weeks, it was a team effort under the leadership of Shakear Wilson. The majority contributors for each section are the following:

- Brian Colditz – Contributed to the development of “Initial Business model”, “Use Case UML Diagrams and Descriptions”, and “Testing/Revisions “
- Ty Kress – Contributed to the development of “Testing/Revisions” and “Use Case UML Diagrams and Descriptions”
- William Serowik - Contributed to the development of “Testing/Revisions", "Initial Requirements” and “Use Case UML Diagrams and Descriptions”

Other contributions that were coordinated by group members consisted of:

- Discord meetings/discussion
- In person discussion
- Formatting and planning outline of document
- Meeting at Writing Center

- Feedback and Proofreading

Appendix: Workflow Authentication

I, Shakear ,agree with the details defined in this document that represent functional requirements of WMS. Also, I agree that the work that was done as stated by this document

Signature:  Date: 10/16/2024

I, Ty Kress, agree with the details defined in this document that represent functional requirements of WMS. Also, I agree that the work that was done as stated by this document

Signature:  Date: 10/16/2024

I, Brian Colditz, agree with the details defined in this document that represent functional requirements of WMS. Also, I agree that the work that was done as stated by this document

Signature:  Date: 10/16/2024

I, William , agree with the details defined in this document that represent functional requirements of WMS. Also, I agree that the work that was done as stated by this document

Signature:  Date: 10/16/2024

Appendix: Report from Writing Center

Formatting:

- add "WMS Project Requirements" as a header through Word instead of physically typing it on the type of each page. This ensures that everyone who opens this document will have the header at the top of the page instead of in the middle of paragraphs.
- add page numbers to the header
- keep the font consistent throughout the entire paper (I noticed that one paragraph switches between Aptos and Times New Roman)

Grammar:

- be sure to add a comma (Oxford comma) before the word "and" in your items in series. While this comma is usually optional, adding it makes the sentence clearer and avoids confusion when listing multiple items in one part of the series.
- similarly, add a comma at the end of introductory clauses. I added a comment on your paper where this happens. Look throughout the rest of the paper for this same mistake.

Misc:

- I switched around the wording "products incoming" to "incoming products" to fit the format of the rest of the sentence.
- check for spelling errors. I added a comment on the ones that I noticed, but it's always good to read over it yourself a few times.
- Overall, there were very few errors.