



An Undergraduate Internship on Falcon Apparel Warehouse Management System with POS system

By

Md. Farhan Soberin

Student ID: 1930403

Spring, 2024

Supervisor:

Md. Abu Sayed

Senior Lecturer

Department of Computer Science & Engineering

Independent University, Bangladesh

June 06, 2024

Dissertation submitted in partial fulfillment for the degree of Bachelor of
Science in Computer Science

Department of Computer Science & Engineering

Independent University, Bangladesh

Attestation

I, Md. Farhan Soberin, hereby affirm that the internship report titled "Falcon Apparel Warehouse Management System with POS System" is the result of my own work and has not been submitted elsewhere for any other purpose. Every source that was used to compile this study has been properly cited. During the course of my internship at Falcon Apparel, I had the opportunity to engage in various tasks and projects relevant to my field of study. The experiences gained and insights gathered during this internship period have significantly contributed to my understanding and development in the web development. I declare that the information presented in this report is accurate to the best of my knowledge. Any opinions or views expressed herein are solely my own and do not necessarily reflect those of Falcon Apparel. Furthermore, I acknowledge the support and guidance provided by my supervisor, Tofael Ahmed, throughout the duration of my internship. Their mentorship and feedback have been invaluable in shaping my learning experience. I affirm that all ethical standards and guidelines were adhered to during the course of this internship, and any ethical considerations relevant to the research conducted for this report have been duly addressed.



Signature

01.07.2024.

Date

Md. Farhan Soberin

Name

Acknowledgement

To everyone who helped make my internship at Falcon Apparel rewarding and enriching, I would like to sincerely thank you. Above all, I want to express my gratitude to my supervisor, Tofayel Ahmed, for his consistent support, direction, and mentorship during my internship. My professional development and learning journey have been greatly influenced by their knowledge, support, and helpful criticism. I express my gratitude to the entire Falcon Apparel team for their cordial greeting, cooperative spirit, and readiness to impart their knowledge and skills to me. Every team member has been essential in giving me insightful knowledge and practical experience in the field of web development. I want to express my gratitude to everyone who has taken the time to guide me, respond to my inquiries, and help me out when I've needed it. Your readiness to make an investment in my growth has been incredibly motivating and has enhanced my educational experience tremendously. In addition, I want to thank my family and friends for their inspiration and support during this internship. Their steadfast faith in my potential has been a continual source of inspiration and fortitude. Finally, I would want to express my gratitude to Independent University, Bangladesh for giving me the chance to complete my internship and for their unwavering support during the entire endeavour. I'd like to express my gratitude to everyone who helped make my internship possible. I sincerely appreciate the chance to have been a part of Falcon Apparel, and your help and advice have been essential.

Letter of Transmittal

06-06-2024

Md. Abu Sayed

Senior Lecturer

Department of Computer Science and Engineering

School of Engineering and Computer Science

Subject: Submission of Internship Report for Graduation Completion.

Dear Sir, It gives me great pleasure to turn in the "**Falcon Apparel Warehouse Management System with POS**" internship report, which is needed to finish my internship at Falcon Apparel. This report summarizes the insights, experiences, and learnings I had while working as an intern at Falcon Apparel. I have had the chance to immerse myself in many facets of web development throughout my internship, learning from my supervisor, Tofael Ahmed, and the helpful team of Falcon Apparel as I have gained real-world knowledge and experience. I think the information in this report fairly captures the range of my internship experiences and my contributions to the tasks that were given to me. With my report, I intend to shed light on the work I did during my internship and use it as a useful resource for my academic evaluation.

I would want to take this opportunity to thank my supervisor, Md. Abu Sayed, and the entire Falcon Apparel team for their unwavering support, guidance, and inspiration during my internship. I am sure that my report meets all the requirements needed to finish an internship program and get a partial completion for a bachelor's degree in computer science and engineering. I would be incredibly appreciative if you could review my report and provide your thoughtful comments. I would also be very grateful if you might find this report to be informative and helpful.

Yours Sincerely,

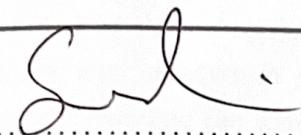
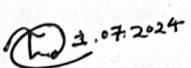
Md. Farhan Soberin

ID:1930403

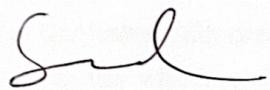
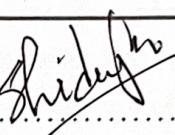
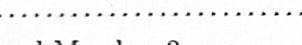
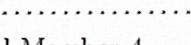
Department of Computer Science & Engineering
Independent University, Bangladesh.

Evaluation Committee

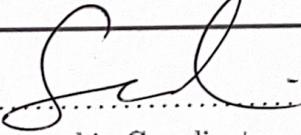
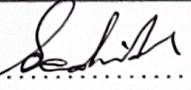
Supervision Panel

	
Academic Supervisor	Industry Supervisor

Panel Members

	
Panel Member 1	Panel Member 2
.....
	
Panel Member 3	Panel Member 4

Office Use

	
Internship Coordinator	Head of the Department
.....
Industry Coordinator of the Department	

Abstract

The creation and deployment of a warehouse management system built with Laravel is presented in this internship report. The system was designed and developed while the author was an intern at Falcon Apparel. This project's main goals were to improve efficiency, streamline the warehouse management procedure, and give real-time tracking and reporting capabilities for operations related to the warehouse. The report opens with a summary of the project's history and setting, highlighting the difficulties the company has had in efficiently running its warehouse. The process used to create the warehouse management system is then covered in detail, including requirement collecting, system design, and Laravel framework implementation. The paper also covers the difficulties that arose during the development process and the methods used to resolve them. It also summarizes the lessons discovered and offers suggestions for upcoming improvements and adjustments to the warehouse management system. Overall, Falcon Apparel has made great progress toward streamlining its warehouse management procedures with the help of the Laravel-based system. It offers a reliable and scalable way to enhance operational effectiveness, facilitate well-informed decision-making, and manage warehouse-related tasks efficiently.

Keywords— laravel, php, warehouse, pos, management, falcon, apparel

Contents

Attestation	i
Acknowledgement	ii
Letter of Transmittal	iii
Evaluation Committee	iv
Abstract	v
1 Introduction	1
1.1 Overview	1
1.2 Problem Statement	1
1.3 Objectives	2
1.4 Scopes	2
2 Literature Review	5
2.1 Relationship with Undergraduate Studies	5
2.2 Related works	6
3 Project Management & Financing	7
3.1 Work Breakdown Structure	7
3.2 Process & Activity wise Time Distribution	8
3.3 Gantt Chart	8
3.4 Process & Activity wise Resource Allocation	10
3.5 Estimated Costing	11
4 Methodology	13
5 Body of the Project	16
5.1 Work Description	16
5.2 Requirement Analysis	17
5.3 System Analysis	20
5.3.1 Six Element Analysis	21
5.3.2 Feasibility Analysis	21

CONTENTS

5.3.3	Problem Solution Analysis	25
5.3.4	Effect and Constraints Analysis	26
5.4	System Design	27
5.5	Implementation	29
5.6	Testing	31
6	Results & Analysis	33
7	Project as Engineering Problem Analysis	34
7.1	Sustainability of the Project	34
7.2	Social and Environmental Effects and Analysis	34
7.3	Addressing Ethics and Ethical Issues	35
8	Lesson Learned	36
8.1	Problems Faced During this Period	36
8.2	Solution of those Problems	36
9	Future Work & Conclusion	37
9.1	Future Works	37
9.2	Conclusion	37
	Bibliography	38

List of Figures

3.1	Work Breakdown Structure (WBS)	7
3.2	Gantt Chart for WMS Project	10
3.3	Process & Activity wise Resource Allocation	11
3.4	Estimated costing chart	11
5.1	Process & Activity wise Resource Allocation	18
5.2	Use Case diagram of Falcon Apparel Warehouse Management System	27
5.3	Activity diagram of Falcon Apparel Warehouse Management System	28
5.4	Architecture diagram of Falcon Apparel Warehouse Management System	29
5.5	Log in page of Falcon Apparel Warehouse Management System	29
5.6	Dashboard of Falcon Apparel Warehouse Management System	30
5.7	Order list of Falcon Apparel Warehouse Management System	30
5.8	Inventory of Falcon Apparel Warehouse Management System	31
5.9	Product listing page of Falcon Apparel Warehouse Management System	31

List of Tables

3.1	Process & Activity Wise Time Distribution	8
5.1	System Elements and Their Descriptions	21
5.2	Problem solution analysis	25
5.3	Effect and Constraint Analysis	26
5.4	Test Cases for System Validation	32

Chapter 1

Introduction

1.1 Overview

My internship was completed at Falcon Apparel. On February 5th, I began my internship at this organization. I began working on the warehouse management system project under the direction of my project manager, Tofayel Ahmed. Currently, Falcon Apparel is exploring the possibility of developing their own web-based warehouse management system. Falcon Apparel's existing warehouse management system is primarily based on manual operations, which causes tracking and warehouse management to be inefficient, inaccurate, and time-consuming.

1.2 Problem Statement

During my internship, I discovered a number of important problems with the warehouse management system's digitalization.

1. **Manual Data Input:** A significant portion of the current system is based on manual data input, which is prone to human error and can result in disparities in warehouse records as well as possible stockouts or overstock scenarios.
2. **Absence of Real-Time Updates:** Warehouse levels are not updated in real-time when automated processes are not present. Decision-making procedures are hampered by this lack of real-time visibility, which leads to less-than-ideal warehouse management techniques.
3. **Ineffective Tracking:** It takes a lot of time and labor to track warehouse traffic, including sales, purchases, and returns. This inefficiency can affect customer satisfaction and operating expenses by delaying the identification of supply shortages or surplus.
4. **Limited Accessibility:** Authorized staff find it difficult to remotely access warehouse information due to the current system's lack of accessibility. This restriction makes it more difficult to respond quickly to changing market needs and to make decisions.

1.3. OBJECTIVES

5. **Ineffective Reporting:** Because data collection and analysis are done manually, creating thorough reports on warehouse levels, product turnover, and trends is a laborious effort. This makes it more difficult to plan strategically and to maximize warehouse management techniques.
6. **Risk of Data Loss:** Relying too much on manual documentation raises the possibility of data destruction or loss, putting warehouse records' security and integrity at risk. The accuracy and dependability of the warehouse management system are seriously threatened by this.

A digitalized warehouse management system that automates procedures, offers real-time visibility, improves accessibility, increases tracking accuracy, and makes thorough reporting easier must be implemented in order to address these issues. [1] Falcon Apparel wants to improve customer happiness by cutting expenses, streamlining processes, and optimizing inventory levels through the digitalization of the warehouse management system.

1.3 Objectives

The objective of this project is to develop and implement a comprehensive warehouse management system using the Laravel framework. The system aims to streamline the process of inventory tracking, management, and reporting for Falcon Apparel. Key goals include:

1. Designing an intuitive and user-friendly interface for adding, editing, and deleting inventory items.
2. Integrating barcode to facilitate efficient inventory management and tracking.
3. Developing reporting and analytics functionalities to generate insights into inventory levels, usage patterns, and trends.
4. Ensuring scalability and flexibility to accommodate future growth and evolving business needs.
5. Providing secure user authentication and authorization mechanisms to safeguard sensitive inventory data.

By achieving these objectives, the project aims to improve the efficiency, accuracy, and transparency of warehouse management processes at Falcon Apparel, ultimately contributing to enhanced operational performance and informed decision-making.

1.4 Scopes

1. Inventory Management Scope:

- Implementing features for managing inventory within the warehouse, including receiving, storing, and distributing goods.

1.4. SCOPES

- Tracking stock levels, locations, and movement of inventory items.
- Providing functionalities for inventory reconciliation and cycle counting to ensure accuracy.

2. Warehouse Operations Scope:

- Managing warehouse operations such as picking, packing, and shipping orders.
- Optimizing warehouse layout and storage configurations for efficient space utilization.
- Implementing workflows and task management features to streamline warehouse operations.

3. Inventory Control Scope:

- Enforcing inventory control measures to prevent stockouts, overstocking, and inventory shrinkage.
- Implementing replenishment strategies to maintain optimal inventory levels.
- Integrating with supply chain systems for demand forecasting and inventory planning.

4. Warehouse Automation Scope:

- Automating repetitive tasks and processes within the warehouse, such as order picking and packing.
- Implementing barcode scanning and RFID technologies for inventory tracking and management.
- Integrating with material handling equipment and warehouse automation systems for improved efficiency.

5. Reporting and Analytics Scope:

- Developing reporting features to generate warehouse performance metrics and KPIs.
- Analyzing warehouse data to identify trends, bottlenecks, and areas for improvement.
- Providing dashboards and visualization tools for monitoring warehouse operations in real-time.

6. User Management Scope:

- Implementing user authentication and access control mechanisms to manage user permissions within the warehouse management system.
- Defining user roles and responsibilities for warehouse staff, supervisors, and administrators.

1.4. SCOPES

- Providing user management features for adding, editing, and deactivating user accounts.

7. Scalability and Performance Scope:

- Designing the system architecture for scalability to handle growing warehouse operations and increasing data volumes.
- Optimizing database queries and application performance for fast response times during peak periods.
- Putting in place load balancing and caching techniques to provide high availability and dependability.

Chapter 2

Literature Review

2.1 Relationship with Undergraduate Studies

My work in the company has been made easier by the theoretical information I have gained during my undergraduate education. I was able to complete the duties that were given to me while developing my problem-solving abilities and learning how a website functions, both frontend and backend, with the aid of programming classes. Additionally, because most of the courses required group projects, it helped me develop my communication and teamwork skills, which were useful because I had to work in teams at work. My curiosity in this profession was piqued by the project-based courses as well. These were the classes that I took that were pertinent to the project I worked on during my internship:

1. **CSE309 (Web Applications and Internet):** In this course, we were taught the frontend of a website and the backend. For frontend, HTML, CSS, Bootstrap and JavaScript were taught, and for backend, we were taught PHP. The projects in this course were helpful since I could build a website from scratch.
2. **CSE307 (System Analysis and Design):** The main objective of this course is to teach us how a system works and how to document a system for each of the software development processes. I could relate to the terms and the documentation used in the industry since those were taught in the course. Moreover, it helped me to understand the process of software development and how each phase is carried out.
3. **CSE303 (Database Management):** This course helped me to understand how the frontend and backend of a website is connected and how the database is managed for a website.
4. **CSE203+L (Data Structure):** By learning how to store data in different formats to optimize the code, this course has helped me to code more efficiently.
5. **CSE451 (Software Engineering):** This course taught me how to make work breakdown structure (WBS), Gantt chart, SDLC, and many crucial things about software documentation and working on projects.

2.2 Related works

Some of the works based on Warehouse Management System are discussed below:

- a) **Literature Reviewed: Performance Improvement of Inventory Management System Processes by an Automated Warehouse Management System.**

This paper underscores the necessity of transitioning from manual to automated Warehouse Management Systems (WMS) for improved efficiency and reliability. Focused on a telecommunications company's small warehouse handling scratch cards and SIM cards, the study stresses the need for thorough process analysis before automation. Business Process Modelling and Notation (BPMN) is proposed for workflow visualization. Selecting scalable software capable of managing large data volumes is crucial. An Enterprise Resource Planning (ERP) system is considered for integration. The implemented software successfully handles data, ensuring FIFO sorting of serial numbers and streamlining order processing. Additionally, the study discusses the implementation of an in-house labelling line to optimize space and reduce costs. Time and motion studies inform optimal processes for labelling, ensuring minimal errors and efficient performance. In conclusion, the transition to automated warehouse management enhances reliability and efficiency, simplifying operations for all stakeholders. [2]

- b) **Literature Reviewed: Smart Warehouse Management System Concept with Implementation, 2018, E. Žunić, S. Delalić, K. Hodžić, A. Beširević and H. Hindija.**

This study examines the idea and execution of an intelligent warehouse management system (WMS) in one of Bosnia and Herzegovina's biggest distribution firms. The system improves warehouse processes, including order picking, transport, tracking, product placement, stock transfer, and stock planning, by utilizing artificial intelligence and optimization algorithms. Anomaly detection is integrated into the process to further improve it. The necessity of an effective WMS in distribution is discussed in the introduction, along with the methodical arrangement of warehouses and their potential for optimization at every stage. The case study, results, conclusions, and literature analysis in this paper show how streamlined procedures can save a significant amount of time and money. Subsequent research endeavours to extend optimization methodologies to alternative distribution procedures and enhance client engagement and monitoring functionalities. [1]

Chapter 3

Project Management & Financing

3.1 Work Breakdown Structure

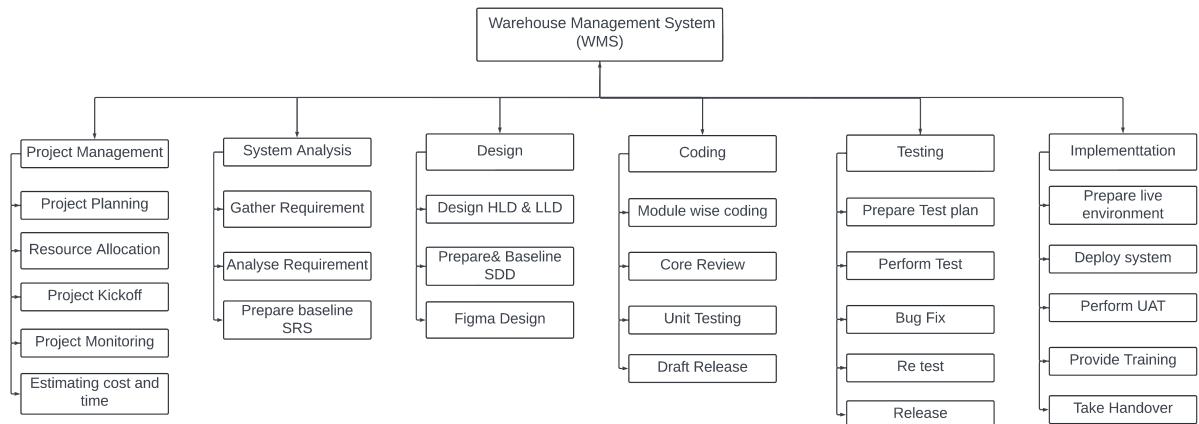


Figure 3.1: Work Breakdown Structure (WBS)

The Work Breakdown Structure (WBS) for the Warehouse Management System (WMS) project is a comprehensive framework that breaks down the project into six main phases, each comprising specific tasks to ensure successful completion:

- Project Management:** This phase focuses on planning, resource allocation, kick-off, monitoring, and estimating cost and time to ensure the project is executed efficiently and effectively.
- System Analysis:** In this phase, requirements are gathered from stakeholders, analysed, and documented in a baseline Software Requirements Specification (SRS) to guide the design and development of the system.
- Design:** This phase involves creating high-level and low-level designs, preparing a Software Design Document (SDD), and developing detailed UI/UX designs using tools like Figma to ensure a user-friendly interface.

3.2. PROCESS & ACTIVITY WISE TIME DISTRIBUTION

- d) **Coding:** The actual development of the system takes place in this phase, including module-wise coding, code reviews, unit testing, and preparing a draft release for integration and testing.
- e) **Testing:** Comprehensive testing is conducted in this phase, including preparing a test plan, performing various tests, fixing bugs, retesting, and finalizing the release to ensure the system functions as expected.
- f) **Implementation:** The final phase involves deploying the system in a live environment, conducting User Acceptance Testing (UAT), providing training to staff, and handing over the system to the client with all necessary documentation and support plans.

This WBS provides a structured approach to managing and executing the WMS project, ensuring that all necessary tasks are clearly defined and systematically addressed.[3]

3.2 Process & Activity wise Time Distribution

Activity Name	Work Percentage	Expected Days	Considered Days
Project Management	8%	7	8
System Analysis	8%	7	17
Design	17%	14	13
Coding	33%	28	25
Testing	17%	14	13
Implementation and Deployment	17%	14	8
Total	100%	84	84

Table 3.1: Process & Activity Wise Time Distribution

The project's duration, as determined by the critical path, is from February 5, 2024, to April 25, 2024. The tasks listed above are the most important ones, and they are the ones that will directly affect when the project will be finished. It is essential to oversee and manage these tasks to guarantee that the project is finished on schedule.

3.3 Gantt Chart

- **Project Management (Early February - Late April):**
 - **Project Planning:** Initiates the project with thorough planning, identifying key objectives, and defining the scope.
 - **Resource Allocation & Assign Work:** Assigns resources and tasks to team members to ensure efficient project execution.
 - **Project Kickoff:** Officially starts the project, aligning stakeholders and team members on the project's goals and expectations.

3.3. GANTT CHART

- **Project Monitoring (Ongoing):** Involves continuous monitoring, risk management, and regular updates to ensure the project stays on track.
- **System Analysis (Early to Mid-February):**
 - **Gather Requirement:** Collects detailed requirements from stakeholders to understand the necessary functionalities.
 - **Analyse Requirement:** Analyses the gathered requirements to ensure clarity and feasibility.
 - **Prepare and Baseline SRS:** Documents the requirements in a Software Requirements Specification (SRS) for reference during development.
- **Design (Mid-February to Early March):**
 - **Design HLD & LLD:** Develops high-level and low-level designs to outline the system architecture and detailed component designs.
 - **Prepare and Baseline SDD:** Creates a Software Design Document (SDD) to serve as a blueprint for development.
 - **Figma Design:** Develops UI/UX designs using Figma to ensure a user-friendly interface.
- **Coding (Late February to Mid-March):**
 - **Module Wise Coding:** Implements the system in modular components, each focusing on specific functionalities.
 - **Core Review:** Conducts code reviews to ensure quality and consistency.
 - **Unit Testing:** Performs unit tests to verify that individual modules function correctly.
 - **Draft Release:** Prepares an initial version of the system for further testing and integration.
- **Testing (Mid-March to Early April):**
 - **Prepare Test Plan and Test Cases:** Develops a comprehensive test plan and detailed test cases.
 - **Perform Test and Report Bug:** Executes the tests and reports any defects found.
 - **Bug Fix and Retest:** Fixes reported bugs and retests to ensure issues are resolved.
 - **Release:** Finalizes testing and prepares the system for deployment.
- **Implementation (Late March to Late April):**
 - **Prepare the Live Environment:** Sets up the necessary infrastructure for the live system.

3.4. PROCESS & ACTIVITY WISE RESOURCE ALLOCATION

- **Deploy the System:** Deploys the system to the live environment.
- **Perform UAT:** Conducts User Acceptance Testing (UAT) to validate the system with end-users.
- **Provide Training:** Trains staff on how to use the new system effectively.
- **Prepare & Provide Operational Manual:** Develops and distributes a manual for ongoing system operations.
- **Complete the System Handover Checklist:** Ensures all deliverables are met and documented before handover.
- **Take Handover & Sign Off:** Officially hands over the system to the client.
- **Send Project Closure Note and Collect Lesson Learned:** Concludes the project with a closure note and collects feedback for future improvements.

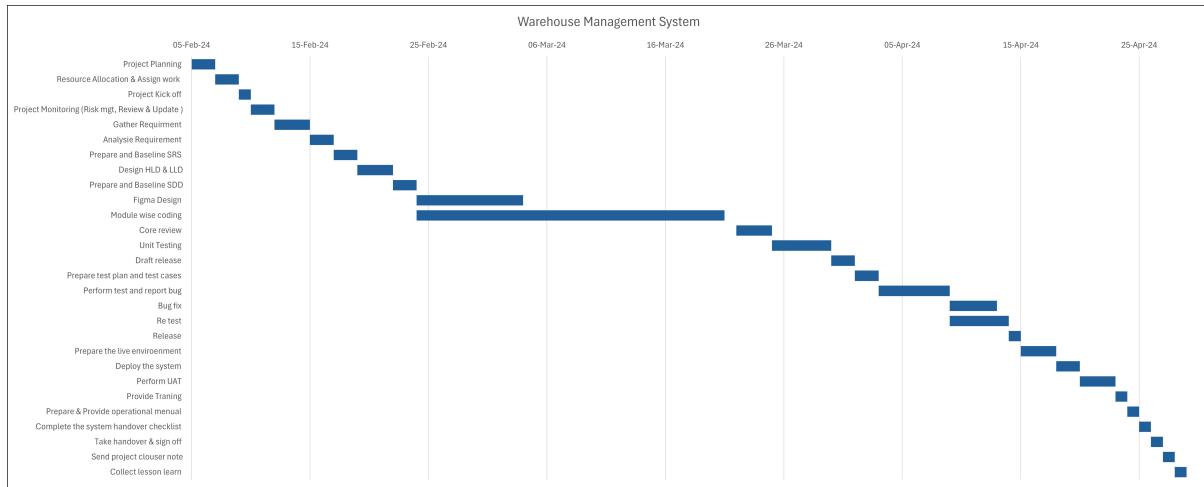


Figure 3.2: Gantt Chart for WMS Project

This Gantt chart provides a structured timeline that ensures each phase of the project is executed efficiently and effectively, culminating in the successful deployment and adoption of the WMS.

3.4 Process & Activity wise Resource Allocation

The practice of arranging and distributing available resources in the most practical and efficient way is known as resource allocation. Though they can occasionally be few, resources are always needed for undertakings. Therefore, scheduling and allocating these resources in accordance with the project timeframe falls under the purview of the project manager. Therefore, resource allocation is the process of managing the project and allocating resources to ensure that it moves forward as efficiently and smoothly as possible. The table below shows how the resources are distributed.[4]

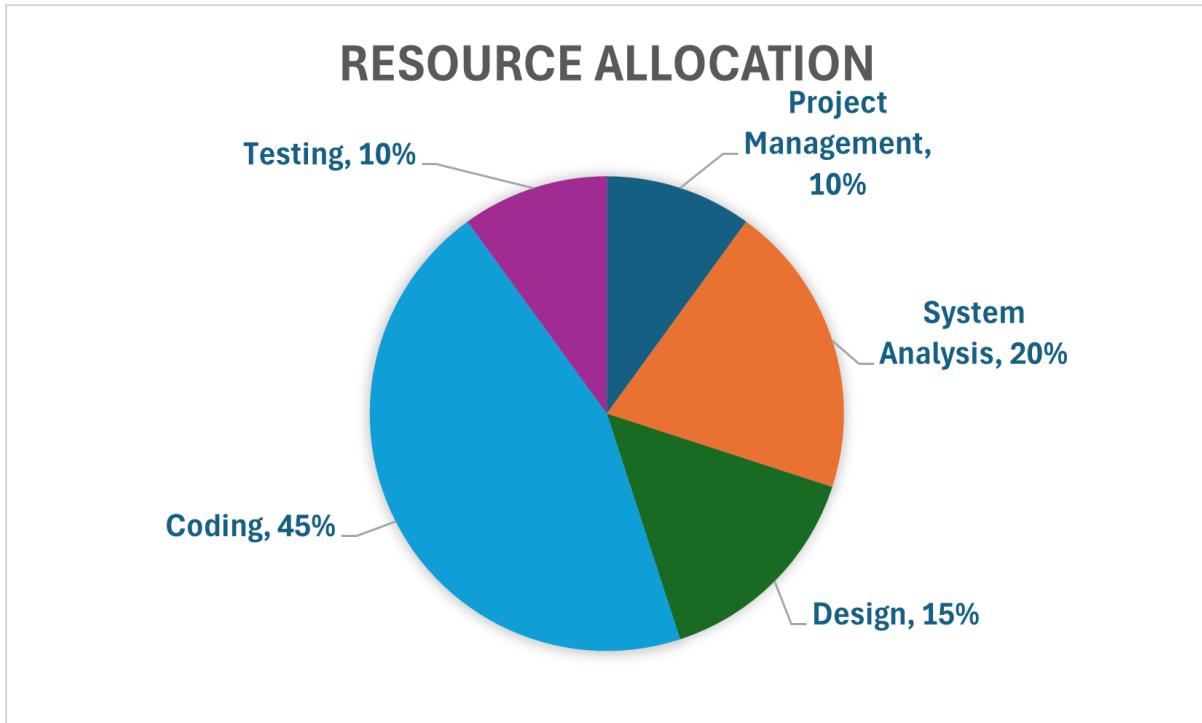


Figure 3.3: Process & Activity wise Resource Allocation

3.5 Estimated Costing

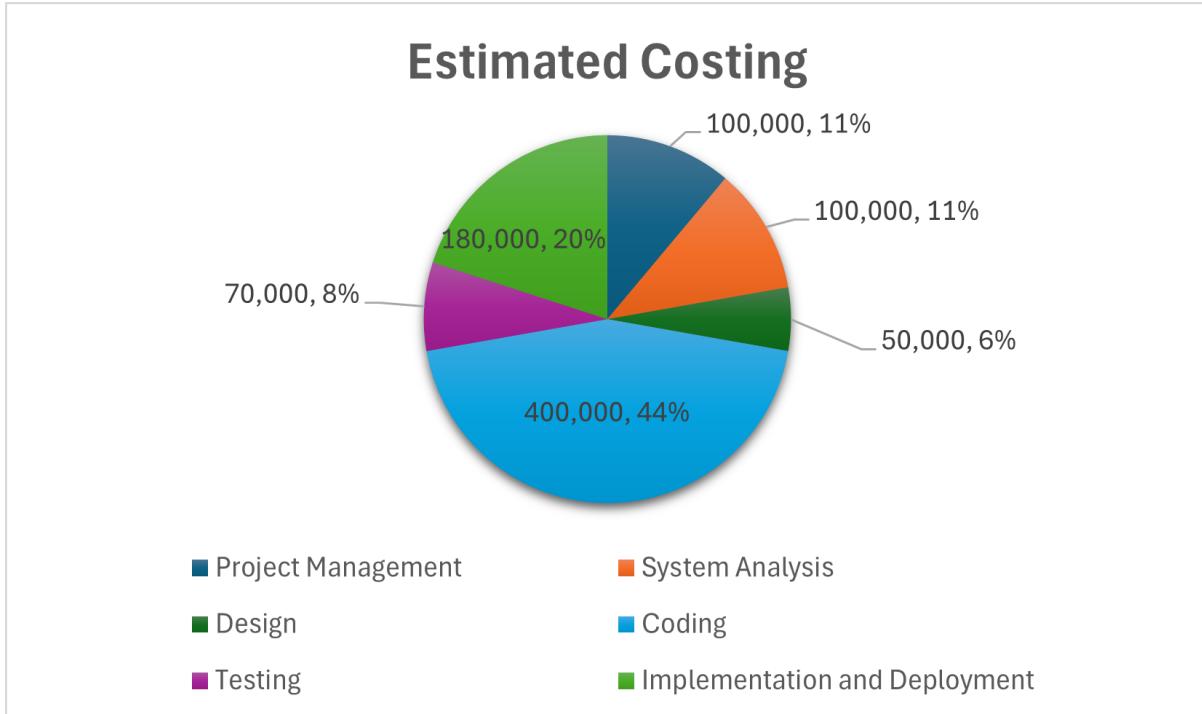


Figure 3.4: Estimated costing chart

The project is expected to cost BDT 9,000,000 in total. This budgetary allotment guarantees the effective development and implementation of the project's fundamental features and

3.5. ESTIMATED COSTING

components. Keeping a small development team, utilizing cost-effective solutions, and giving priority to core functionality would enable the project to provide a scalable and user-friendly system within budgetary restrictions. Future updates and new features can be budgeted for and planned for later stages, guaranteeing the system's scalability and ongoing progress.

Chapter 4

Methodology

The development of the Laravel-based warehouse management system followed an Agile Software Development Life Cycle (SDLC) approach. This methodology ensured flexibility, iterative progress, and close collaboration with stakeholders, ensuring the project was completed efficiently and met all specified requirements. The methodology can be broken down into several key phases:

a) **Requirements Gathering and Analysis**

- Conducted meetings and interviews with stakeholders, including warehouse managers, staff, and IT personnel, to gather detailed requirements.
- Documented functional and non-functional requirements, focusing on core functionalities such as inventory tracking, warehouse operations, and user management.
- Analysed existing warehouse processes and systems to identify gaps and areas for improvement.

b) **Planning**

- Divided the project into iterations (sprints), each lasting 2-4 weeks.
- Established sprint goals and tasks, ensuring each sprint delivered functional components of the system.

c) **System Design**

- Developed a high-level system architecture, outlining the major components and their interactions within the Laravel framework.
- Designed the database schema to efficiently store and manage warehouse data, ensuring data integrity and scalability.
- Created detailed wireframes and user interface (UI) mock-ups for key system modules, including inventory management, order processing, and reporting dashboards in each sprint.

-
- Defined user roles and permissions to ensure secure and role-based access to system functionalities.
 - Developed high-level system architecture and database schema designs to support scalability and efficiency.
 - Engaged stakeholders for feedback on design prototypes to ensure alignment with user expectations and requirements.

d) Development

- Set up the development environment using Laravel, MySQL, and other necessary tools and libraries.
- Implemented the core functionalities in iterative sprints, following agile development practices:
 - *Inventory Management*: Developed features for tracking stock levels, locations, and movements; implemented inventory reconciliation and cycle counting functionalities.
 - *Warehouse Operations*: Created modules for managing picking, packing, and shipping processes; optimized warehouse layout and storage configurations.
 - *Reporting and Analytics*: Developed reporting features for warehouse performance metrics and KPIs; implemented dashboards and visualization tools for real-time monitoring.
- Conducted regular code reviews and testing to ensure code quality and functionality.

e) Integration

- Integrated the warehouse management system with existing enterprise systems such as ERP, TMS, and OMS for seamless data exchange.
- Developed APIs to facilitate integration with third-party applications and services, including e-commerce platforms and CRM systems.
- Ensured compatibility with warehouse automation equipment and devices, such as barcode scanners and material handling systems.

f) Testing

- Performed unit testing, integration testing, and system testing to verify that all components functioned correctly and met the specified requirements.
- Ensured that all features developed in a sprint were thoroughly tested before the sprint review.
- Conducted user acceptance testing (UAT) with stakeholders at the end of each sprint to validate functionality and gather feedback.
- Addressed any issues or bugs identified during testing and refined the system accordingly.

g) Deployment and Training

- Deployed the warehouse management system to a staging environment for final validation before production release.
- Provided comprehensive training sessions for warehouse staff and administrators to ensure smooth adoption of the new system.
- Created user manuals and documentation to support ongoing use and maintenance of the system.

h) Evaluation and Continuous Improvement

- After implementation, tracked system performance and user input to find any areas that needed improvement.
- Implemented iterative updates and enhancements based on feedback and evolving business needs.
- Established a maintenance plan to ensure the system remains up-to-date and continues to meet operational requirements.

By following this Agile methodology, the project team was able to remain flexible and responsive to changes, ensuring the final product met the stakeholders' needs and provided a robust solution for managing warehouse operations. The iterative nature of Agile allowed for continuous delivery of functional components, early detection of issues, and incorporation of user feedback throughout the development process. The project's goal was to provide Falcon Apparel with a reliable, scalable, and easy-to-use warehouse management system that would improve visibility, accuracy, and operating efficiency.

Chapter 5

Body of the Project

5.1 Work Description

During my internship at Falcon Apparel, I was tasked with developing a comprehensive warehouse management system integrated with a point-of-sale (POS) system using the Laravel framework. The project aimed to enhance the efficiency and accuracy of warehouse operations and sales transactions. Below is a detailed description of the work I carried out:

- Conducted initial meetings with stakeholders, including warehouse managers, sales staff, and IT personnel, to gather detailed requirements and understand the existing workflow.
- Documented functional and non-functional requirements, creating a product backlog with prioritized features and user stories.
- Participated in sprint planning meetings to define the scope and objectives of each iteration.
- Developed high-level system architecture, outlining the major components and their interactions within the Laravel framework.
- Designed the database schema to efficiently store and manage warehouse and sales data, ensuring data integrity and scalability.
- Created detailed wireframes and UI/UX mock-ups for key system modules, including inventory management, order processing, and POS functionalities.
- Set up the development environment using Laravel, MySQL, and other necessary tools and libraries.
- Implemented core functionalities in iterative sprints, following agile development practices.
- Developed features for adding, editing, and deleting inventory items, tracking stock levels and locations, and integrating barcode scanning for efficient inventory tracking.

5.2. REQUIREMENT ANALYSIS

- Created modules for creating, editing, and managing customer orders, assigning orders to warehouse staff, and tracking order status.
- Implemented POS functionalities for processing sales transactions, synchronizing sales data, and generating sales receipts.
- Implemented user authentication and authorization mechanisms, defining role-based access controls.
- Developed reporting features to generate real-time reports on inventory levels, sales performance, and order status, including graphical dashboards for KPIs and trends analysis.
- Conducted regular code reviews and testing to ensure code quality and functionality.
- Performed unit testing, integration testing, and system testing continuously throughout the development process.
- Conducted user acceptance testing (UAT) with stakeholders to validate the system's usability and effectiveness in real-world scenarios.
- Deployed the system incrementally to a staging environment for final validation before production release.
- Provided comprehensive training sessions for warehouse and sales staff to ensure smooth adoption of the new system.

I learned a lot about full-stack web development, agile processes, and working with cross-functional teams during this project. Falcon Apparel's operational efficiency and data accuracy were greatly enhanced by the effective integration of a POS system with a Laravel-based warehouse management system, which helped to improve decision-making and overall business performance.

5.2 Requirement Analysis

Stakeholders

- Supplier
- Warehouse Staff
- Admin
- POS System
- Sale Processing
- Delivery Team
- Customers

5.2. REQUIREMENT ANALYSIS

Rich Picture

A rich picture is a diagrammatic technique used primarily in systems thinking and soft systems methodology to visually capture and represent complex situations, problems, or systems. It includes various elements such as people, processes, structures, and relationships, often drawn informally with symbols, sketches, and annotations to illustrate interactions, concerns, and viewpoints.

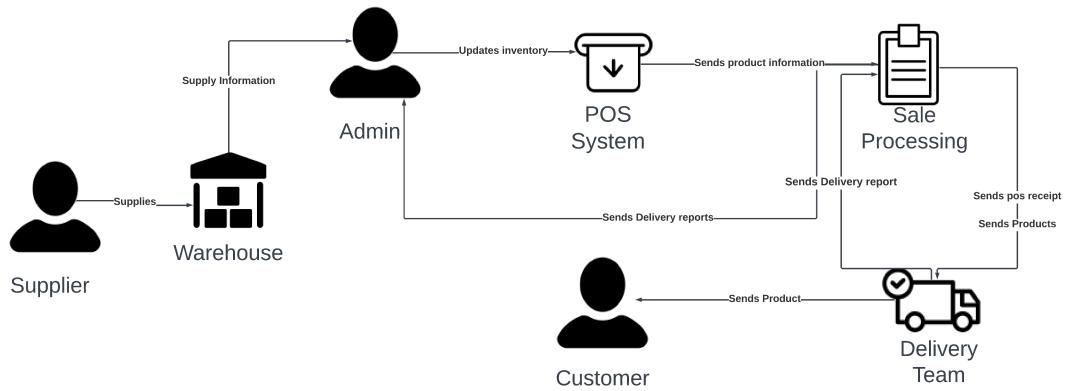


Figure 5.1: Process & Activity wise Resource Allocation

Functional Requirements

a) Inventory Management:

- Ability to add, edit, and delete inventory items.
- Track stock levels, locations, and movements within the warehouse.
- Support for barcode scanning for inventory tracking and management.

b) Order Management:

- Create, edit, and manage customer orders.
- Assign orders to warehouse staff for picking and packing.
- Track order status from creation to delivery.

c) POS Integration:

- Process sales transactions through the POS system.
- Synchronize sales data between the POS and warehouse management system.
- Generate sales receipts and manage returns and exchanges.

d) Reporting and Analytics:

5.2. REQUIREMENT ANALYSIS

- Generate real-time reports on inventory levels, sales performance, and order status.
- Provide graphical dashboards for key performance indicators (KPIs) and trends analysis.

e) Supplier Management:

- Maintain a database of suppliers and their contact information.
- Track purchase orders and inventory received from suppliers.

f) Customer Management:

- Maintain a database of customer information and order history.
- Provide customer insights and sales trends analysis.

g) Notifications and Alerts:

- Send notifications and alerts for low stock levels, order status updates, and system errors.

Non-Functional Requirements

a) Performance

- Inventory search and retrieval should complete within 2 seconds.

b) Scalability

- The system should be scalable to accommodate future growth in data volume and user base.
- It should support horizontal scaling by adding more servers.

c) Usability

- Users with different degrees of technical competence should find it easy to use and intuitive to navigate the user interface.
- Provide comprehensive user documentation and help guides.

d) Reliability

- A 99.9% uptime rate for the system would guarantee high availability for consumers.
- To avoid data loss, have backup and failover systems in place.

e) Maintainability

- The codebase should follow best practices for readability and modularity, facilitating easy maintenance and updates.
- Maintain comprehensive documentation for developers.

5.3. SYSTEM ANALYSIS

f) Compatibility

- The system ought to work with the majority of online browsers, including Chrome, Firefox, Edge, and Safari.

g) Compliance

- The system should comply with relevant industry standards.
- Implement audit trails for tracking system changes and user activities.

5.3 System Analysis

The requirements, architecture, functionality, and performance of the Laravel-based Warehouse Management System (WMS) with POS integration are the main subjects of the system analysis. This guarantees that it satisfies the requirements of the business for smooth sales procedures and effective warehouse operations.[5]

5.3. SYSTEM ANALYSIS

5.3.1 Six Element Analysis

Element	Description	Purpose	Stakeholders	Inputs	Outputs	Constraints
Inventory Control	Tracks and manages inventory levels, updates inventory on receiving and dispatching goods.	To ensure accurate inventory levels and prevent stockouts or overstocking.	Warehouse Staff, Suppliers	Stock arrival data, dispatch data, sales data	Updated inventory levels, inventory reports	Assumes real-time data entry and integration with all sales and procurement processes.
Order Management	Processes incoming orders and coordinates with inventory control for stock availability.	To efficiently process customer orders and ensure timely fulfillment.	Customers, Warehouse Staff	Customer orders, stock levels	Order status updates, pick/pack instructions	Assumes accurate and up-to-date stock level information.
Supplier Management	Maintains supplier information and manages orders to suppliers.	To streamline procurement processes and maintain good supplier relationships.	Suppliers, Procurement Team	Supplier data, procurement requirements	Purchase orders, supplier performance reports	Assumes reliable supplier data and predefined procurement processes.
Sales Processing (POS)	Handles sales transactions, updates inventory in WMS, and generates sales reports.	To process sales efficiently and keep inventory levels accurate.	Sales Staff, Customers	Sales transaction data, customer payment details	Updated inventory levels, sales receipts, sales reports	Assumes POS system integration with WMS and real-time inventory updates.
Warehouse Operations	Manages picking, packing, shipping processes, and tracks goods within different warehouse zones.	To ensure smooth warehouse operations and accurate tracking of goods.	Warehouse Staff	Order details, packing lists, inventory data	Pick lists, packing lists, shipping labels, tracking information	Assumes efficient layout and management of warehouse zones.
Customer Management	Stores customer data, processes order and returns, and manages customer interactions through POS.	To enhance customer service and streamline sales processes.	Customers, Sales Staff	Customer data, order and return requests	Customer profiles, order confirmations, return labels	Assumes that client data is handled and stored securely.

Table 5.1: System Elements and Their Descriptions

5.3.2 Feasibility Analysis

- a) **Technical Feasibility** Evaluating whether the technology and infrastructure required to implement the Laravel-based WMS with POS integration are available and capable of meeting the project's requirements.

Technology Stack:

- Laravel Framework: Suitable for web applications with robust MVC architecture.
- Database: MySQL.
- Frontend: HTML, Bootstrap, CSS, Jquery, Jscript.

5.3. SYSTEM ANALYSIS

- Integration: UIpackage for integrating auth features, Yajra data table package for integrating data table, image intervention package for integrating image upload.
- Network: Reliable internet connectivity for real-time data synchronization.

Risks:

- Complexity of integrating POS with WMS.
- Ensuring data security and compliance with regulations.

Mitigation:

- Utilize middleware or API gateways for seamless integration.
- Implement robust security protocols and data encryption.

Conclusion:

- Technically feasible given the current availability of technologies and infrastructure.
- b) **Economic Feasibility** Assessing the cost-effectiveness of the project and its financial viability.

Initial Costs:

- Development: Salaries for developers, designers, and testers.
- Infrastructure: Cost of servers, cloud services, and hardware.
- Licensing: Software licenses for development tools and libraries.

Ongoing Costs:

- Maintenance: Regular updates, bug fixes, and security patches.
- Support: Customer and technical support staff.
- Hosting: Monthly fees for cloud services and data storage.

Benefits:

- Improved operational efficiency.
- Reduced manual errors and operational costs.
- Increased customer satisfaction and sales through efficient POS.

ROI:

- Enhanced inventory management reduces overstock and stockouts.
- Streamlined processes lead to faster order fulfilment and customer service.

Conclusion:

5.3. SYSTEM ANALYSIS

- Economically feasible with a positive return on investment anticipated within the first year of implementation.
- c) **Operational Feasibility** Determining if the current operational setup can support the new system and if it aligns with the organization's goals.

Current Operations:

- Existing workflows and processes in the warehouse and sales.
- Employee readiness and adaptability to new technology.

System Integration:

- Compatibility with existing systems and processes.
- Training requirements for staff.

Scalability:

- Ability to handle future growth and increased transaction volumes.

Risks:

- Resistance to change from employees.
- Downtime during system transition.

Mitigation:

- Conduct comprehensive training sessions.
- Implement in phases to minimize disruption.

Conclusion:

- Operationally feasible with proper change management and training.
- d) **Legal Feasibility** Evaluating compliance with relevant laws and regulations.

Data Protection:

- Compliance with GDPR, CCPA, and other data protection regulations.
- Ensuring secure handling of customer and supplier data.

E-commerce Regulations:

- Adhering to laws governing online sales and transactions.
- Compliance with tax regulations for sales and inventory.

Risks:

- Non-compliance with data protection laws leading to legal penalties.

5.3. SYSTEM ANALYSIS

- Mismanagement of tax and financial regulations.

Mitigation:

- Regular audits and legal consultations.
- Implementing strict data protection measures and protocols.

Conclusion:

- Legally feasible with adherence to relevant laws and regulations.

e) **Schedule Feasibility**

Overview: Assessing if the project can be completed within the proposed timeline.

Project Plan:

- Detailed timeline based on the work breakdown structure (WBS).
- Milestones and deliverables for each phase.

Resource Allocation:

- Availability of skilled personnel.
- Allocation of necessary resources and tools.

Time Constraints:

- Realistic deadlines for each phase of development, testing, and implementation.

Risks:

- Delays due to unforeseen technical challenges.
- Resource bottlenecks affecting the timeline.

Mitigation:

- Buffer time for critical phases.
- Regular progress reviews and adjustments.

Conclusion:

- Schedule feasible with a well-structured project plan and resource management.

Conclusion: The project of developing a Laravel-based Warehouse Management System with POS integration is feasible across all dimensions. With adequate planning, resource allocation, and risk management, the project can be successfully implemented to enhance operational efficiency, improve customer satisfaction, and provide a significant return on investment.

5.3.3 Problem Solution Analysis

Problem	Solution	Benefit
Inefficient Inventory Management	Automated Inventory Control	Enhanced operational efficiency and accuracy.
Slow Order Processing and Fulfilment	Integrated Order Management	Improved customer satisfaction through faster and more reliable service.
Poor Supplier Coordination	Efficient Supplier Management	Better supplier relationships and procurement processes.
Inconsistent Sales Processing and Inventory Updates	Synchronized POS and WMS	Secure and compliant data handling.
Lack of Data Security and Compliance	Robust Data Security	Successful adoption and minimal disruption during transition.

Table 5.2: Problem solution analysis

5.3. SYSTEM ANALYSIS

5.3.4 Effect and Constraints Analysis

Aspect	Effects	Constraints
Inventory Management	<p>Accurate real-time tracking of inventory levels.</p> <p>Reduces overstock and stockouts.</p> <p>Generates comprehensive inventory reports.</p>	<p>Requires continuous data entry and synchronization.</p> <p>Dependence on reliable internet connectivity for real-time updates.</p> <p>Potential integration issues with existing systems.</p>
Order Processing	<p>Faster order fulfilment and reduced processing times.</p> <p>Enhanced customer satisfaction due to timely deliveries.</p> <p>Automated workflows for picking, packing, and shipping.</p>	<p>Need for seamless integration between order management and inventory control modules.</p> <p>Complex workflows require thorough testing and validation.</p> <p>High initial setup and configuration efforts.</p>
Supplier Management	<p>Streamlined procurement processes and improved supplier relationships.</p> <p>Automated purchase orders based on inventory levels.</p> <p>Enhanced efficiency in restocking and supplier coordination.</p>	<p>Dependence on accurate supplier data and timely updates.</p> <p>Requires consistent performance monitoring and feedback mechanisms.</p> <p>Potential resistance from suppliers used to traditional methods.</p>
Sales Processing (POS)	<p>Real-time inventory updates with each transaction.</p> <p>Improved accuracy in sales data and reporting.</p> <p>Enhanced customer experience with efficient transactions.</p>	<p>Integration complexity between POS and WMS systems.</p> <p>Ensuring system robustness to handle peak transaction loads.</p> <p>Dependence on stable POS hardware and network infrastructure.</p>
Warehouse Operations	<p>Smooth and efficient warehouse operations.</p> <p>Accurate tracking of goods within different warehouse zones.</p> <p>Reduced labour costs through automation.</p>	<p>Requires detailed planning and layout optimization.</p> <p>Potential need for staff training and adjustment to new workflows.</p> <p>Initial investment in automation tools and equipment.</p>
Financial Impact	<p>Long-term cost savings from reduced operational inefficiencies.</p> <p>Positive return on investment through improved business processes.</p> <p>Potential for increased revenue through enhanced customer satisfaction and sales efficiency.</p>	<p>Significant initial investment in system development and implementation.</p> <p>Ongoing maintenance and support costs.</p> <p>Budget constraints and justification of high upfront costs.</p>
Scalability and Growth	<p>System can scale with business growth, handling increased transaction volumes and complexity.</p> <p>Supports future expansion into new markets or addition of new features.</p> <p>Flexibility to adapt to changing business needs and technology advancements.</p>	<p>Ensuring system scalability requires careful planning and robust architecture design.</p> <p>Potential performance issues if scalability is not properly managed.</p> <p>Continuous monitoring and optimization needed to maintain performance and scalability.</p>

Table 5.3: Effect and Constraint Analysis

5.4 System Design

UML Diagrams

A UML diagram is a system picture created using the Unified Modelling Language (UML). Software engineers use the Unified Modelling Language, a developmental modelling language that provides a standardized way to represent the system architecture. The use case and activity diagram for our project is one of many types of UML diagrams that are shown in this section.

Use-case Diagram: A type of behaviour diagram called a use case diagram makes apparent the way users interact with the system. This use-case diagram links and depicts the entire restaurant management system with the help of important use cases and actors.

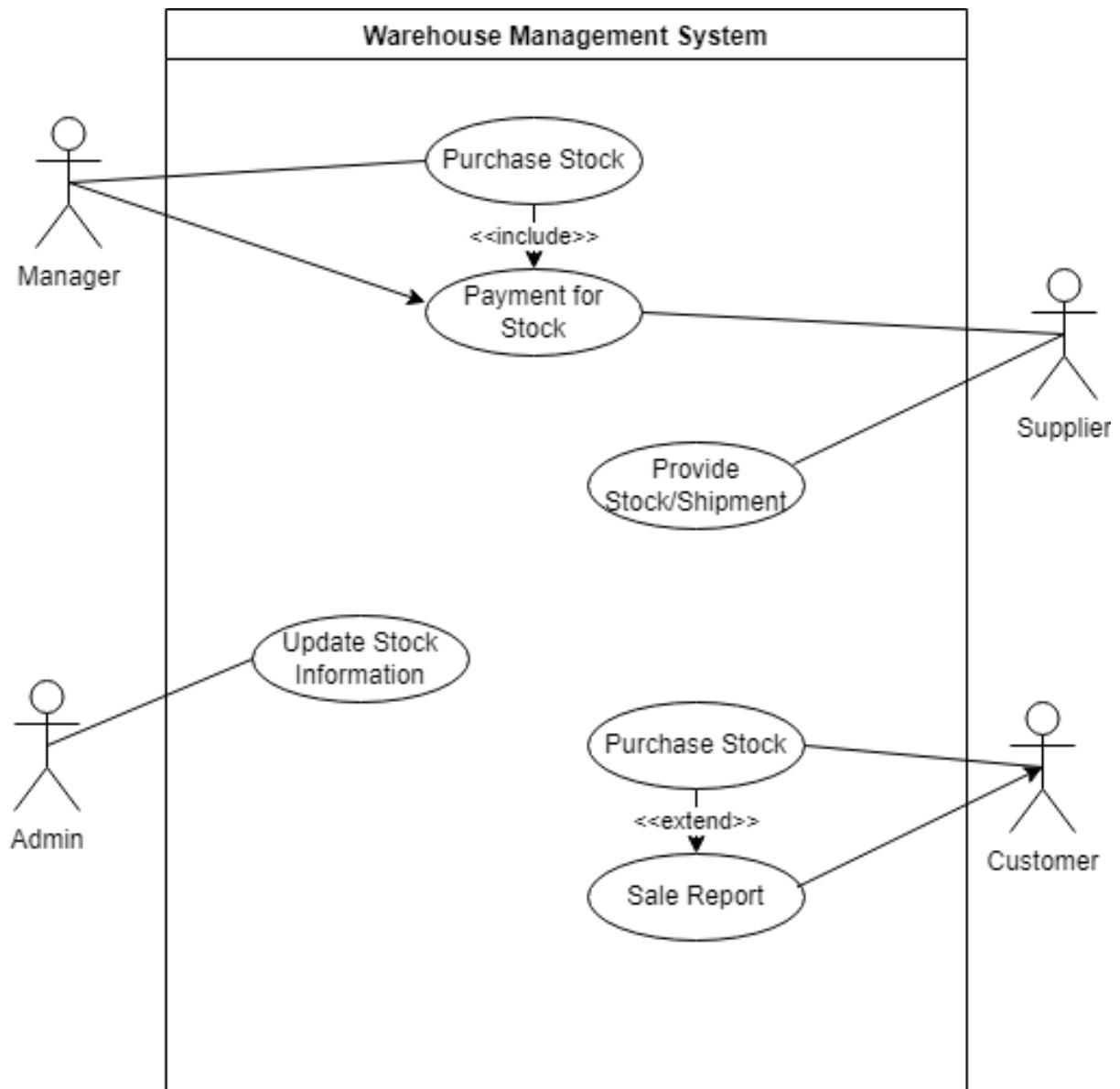


Figure 5.2: Use Case diagram of Falcon Apparel Warehouse Management System

Activity Diagram: Activity diagrams show the several decisions that can be made as the activity's events develop, illuminating the process from start to finish.

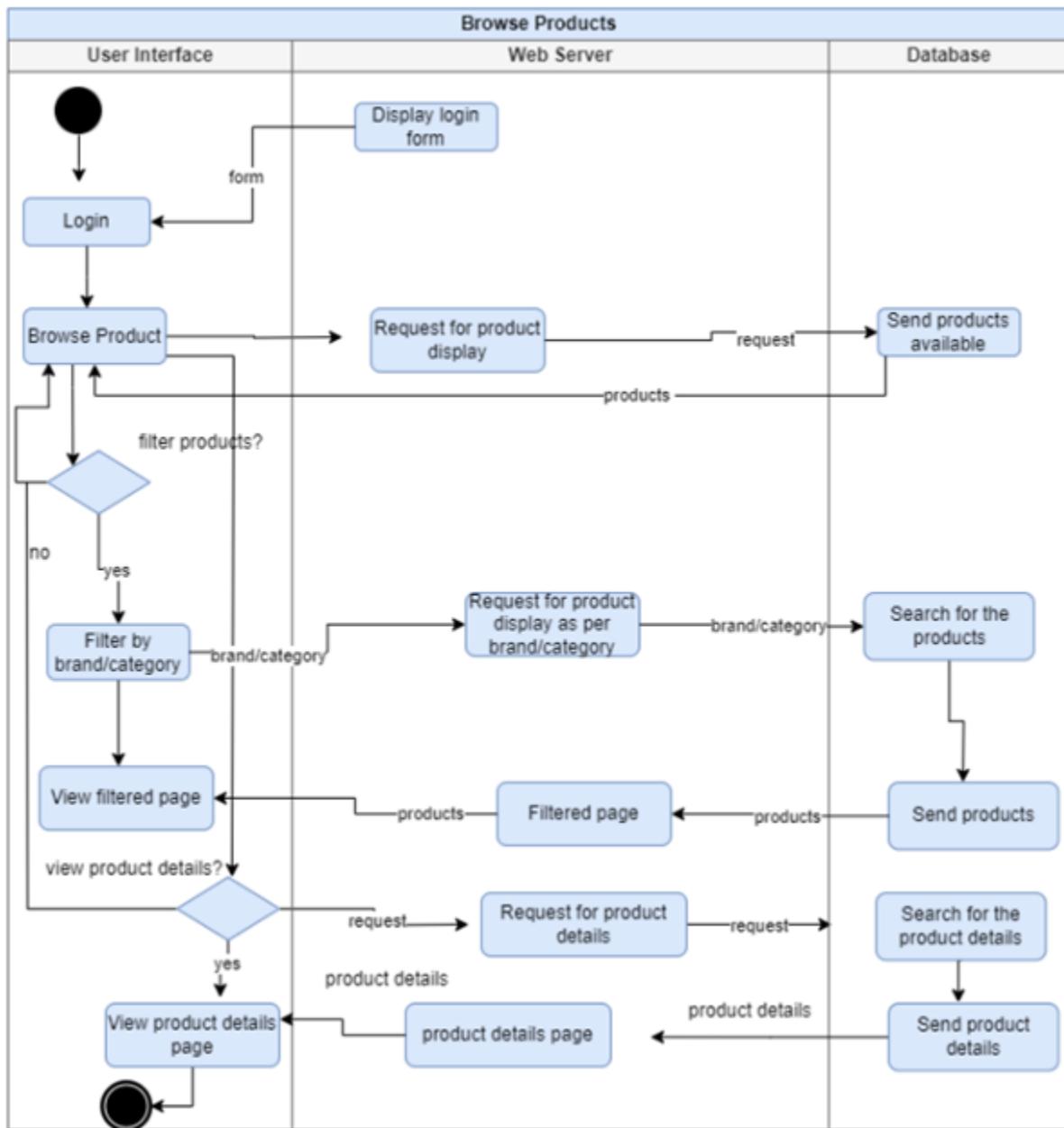


Figure 5.3: Activity diagram of Falcon Apparel Warehouse Management System

Architecture

System architecture diagrams offer a visual representation of the many parts of a system and demonstrate how they interact and communicate with one another

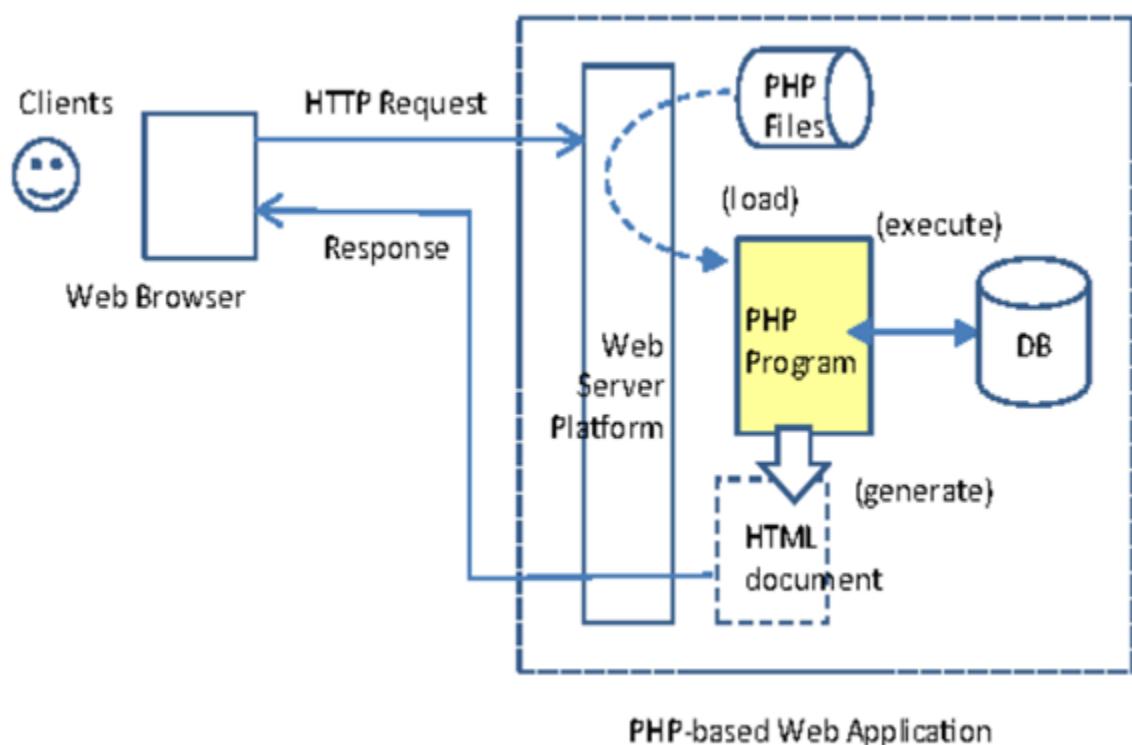


Figure 5.4: Architecture diagram of Falcon Apparel Warehouse Management System

5.5 Implementation

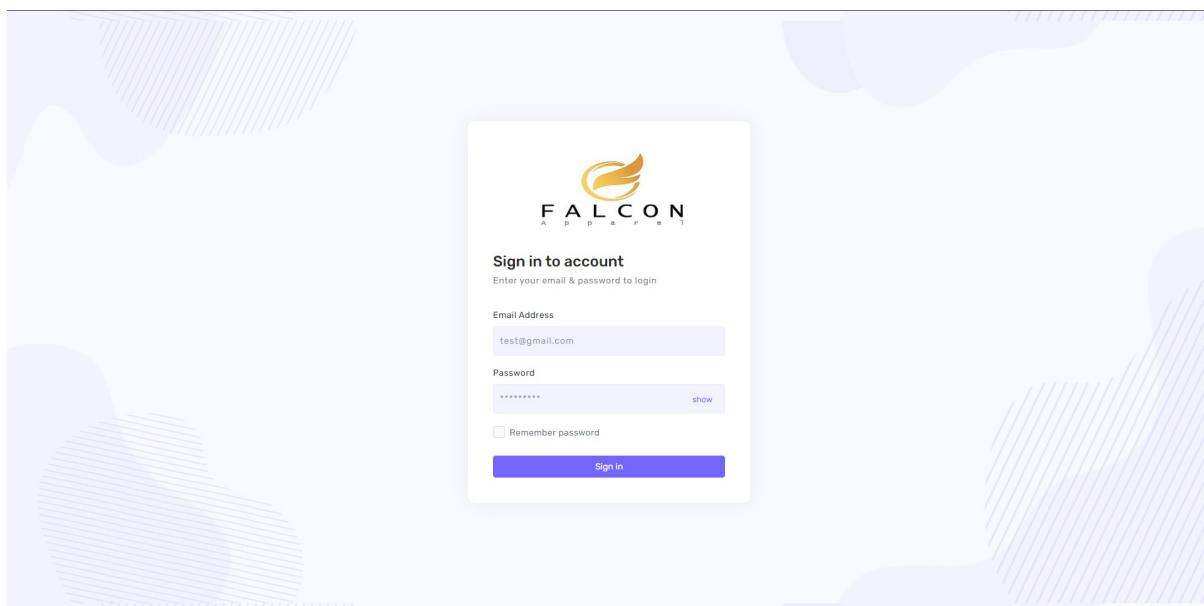


Figure 5.5: Log in page of Falcon Apparel Warehouse Management System

5.5. IMPLEMENTATION

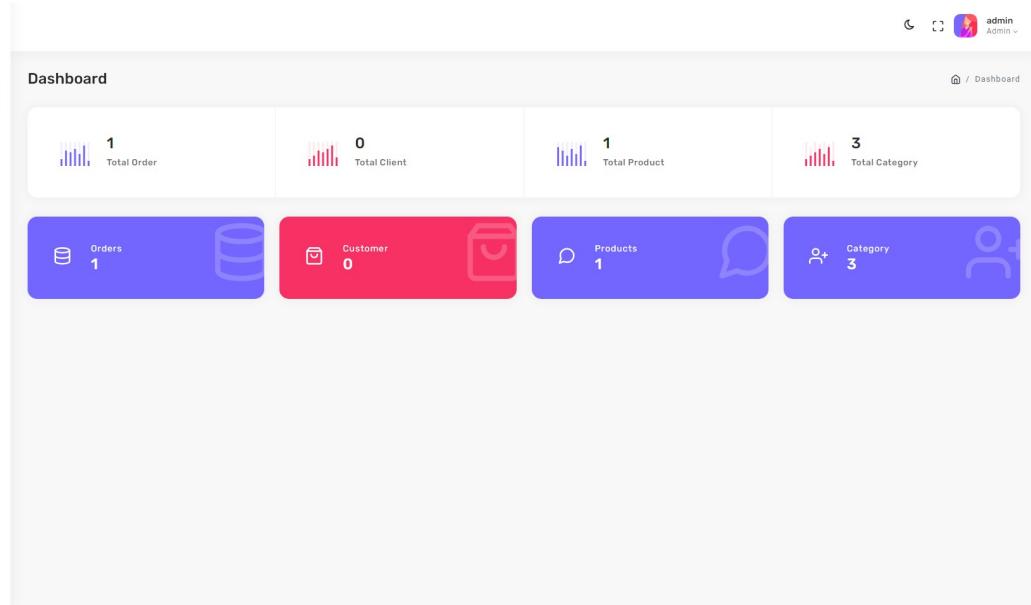


Figure 5.6: Dashboard of Falcon Apparel Warehouse Management System

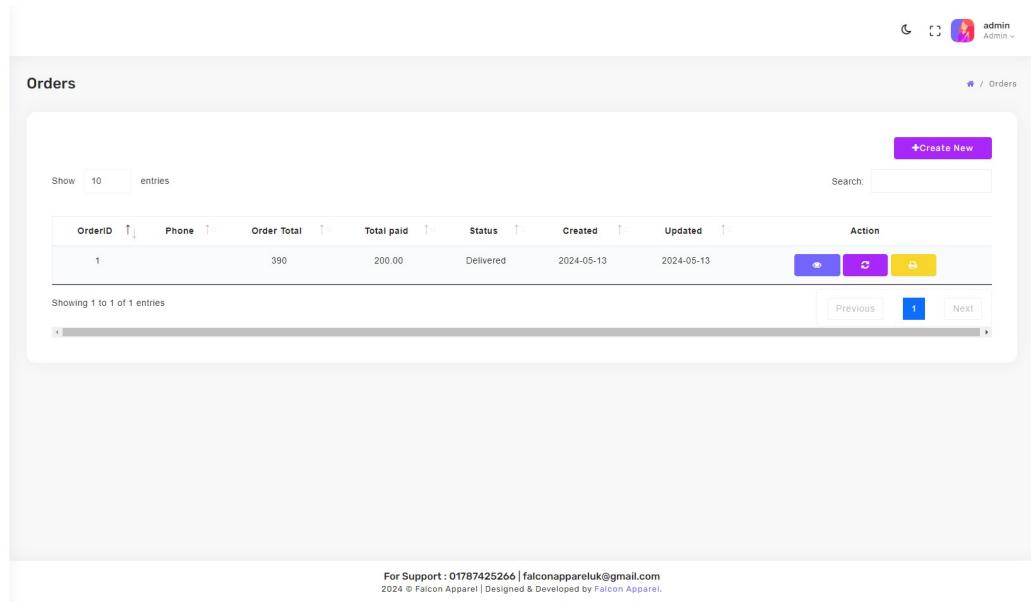


Figure 5.7: Order list of Falcon Apparel Warehouse Management System

5.6. TESTING

Figure 5.8: Inventory of Falcon Apparel Warehouse Management System

Figure 5.9: Product listing page of Falcon Apparel Warehouse Management System

5.6 Testing

The testing phase of the Warehouse Management System (WMS) with POS integration involved rigorous validation to ensure the system met all functional and non-functional requirements. The following table outlines the key test cases, including inputs, expected outputs, actual outputs, and test results.

5.6. TESTING

Test Case ID	Test Case Description	Input	Expected Output	Actual Output	Test Result (Pass/Fail)
TC001	Create Inventory Item	Item details (name, SKU, quantity, price)	Inventory item created successfully with correct details	Inventory item created successfully with correct details	Pass
TC002	Update Inventory Item	Updated item details (name, SKU, quantity, price)	Inventory item updated successfully with new details	Inventory item updated successfully with new details	Pass
TC003	Delete Inventory Item	Item ID to be deleted	Inventory item deleted successfully	Inventory item deleted successfully	Pass
TC004	Process New Order	Order details (customer info, item list, quantities)	Order created successfully and inventory updated accordingly	Order created successfully and inventory updated accordingly	Pass
TC005	Update Order Status	Order ID, new status (e.g., shipped, delivered)	Order status updated successfully	Order status updated successfully	Pass
TC006	Track Order	Order ID	Display order details and current status	Display order details and current status	Pass
TC007	POS Transaction	POS transaction details (items scanned, quantities, total)	Transaction processed successfully; inventory updated	Transaction processed successfully; inventory updated	Pass
TC008	Generate Sales Report	Date range (start date, end date)	Sales report generated with correct data for the specified range	Sales report generated with correct data for the specified range	Pass
TC009	User Authentication	Login credentials (username, password)	User logged in successfully with correct permissions	User logged in successfully with correct permissions	Pass
TC010	Manage Supplier Information	Supplier details (name, contact info, items supplied)	Supplier information saved/updated successfully	Supplier information saved/updated successfully	Pass

Table 5.4: Test Cases for System Validation

Chapter 6

Results & Analysis

A thorough testing process was used to validate the system's functionality. The implementation of the system's fundamental functions—inventory management, order processing, supplier management, and point-of-sale transactions—went well. All of these features worked as they should have in a variety of situations, enabling users to properly and efficiently oversee warehouse operations. Order creation, updating, and tracking functions operated without a hitch, displaying real-time changes and preserving data integrity across the system.

User Interface: The user interface and experience were also closely scrutinized during the testing phase. Feedback from user acceptability tests indicated that the system is easy to use and intuitive. The combination of Laravel with JQuery and JavaScript enabled a responsive and dynamic user interface, increasing user satisfaction overall. This was particularly true for the point-of-sale system, where it was crucial to process sales transactions fast and simply.

Performance: The system performed well and showed strong capabilities. Load testing demonstrated that the WMS could manage numerous users at once and big data volumes without experiencing appreciable performance reduction. This is critical in warehouse settings where speed and efficiency are critical. The system's excellent performance and scalability were made possible by its architecture, which made use of Eloquent ORM's effective database interactions and Laravel's robust framework.

Integration of POS System: Real-time synchronization of sales data with inventory levels was made possible by the smooth integration of the POS system with the WMS. Maintaining correct stock information and expediting the sales process depend on this integration. Testing revealed that the POS system's communication with hardware, such as receipt printers and barcode scanners, was dependable and effective.

Deployment and Integration: The system's deployment and maintainability were taken into account lastly. Utilizing cloud infrastructure for hosting guarantees low downtime and scalability of the system.

In conclusion, the result analysis of the Laravel-based Warehouse Management System with POS integration project indicates a successful implementation that meets all project requirements. The system is functional, user-friendly, performant, secure, and maintainable, making it a valuable tool for efficient warehouse management and sales operations.

Chapter 7

Project as Engineering Problem Analysis

7.1 Sustainability of the Project

The Laravel-based Warehouse Management System (WMS) with POS integration's scalability, maintainability, and future-development potential all contribute to its sustainable design. The system utilizes the Model-View-Controller (MVC) paradigm, which is based on the durable Laravel framework. This style promotes a clear separation of concerns and makes management and flexibility for future feature integration easier. Cloud architecture facilitates scalability by enabling the system to manage growing data volumes and user loads effectively while preserving peak performance. Maintainability is attained by following code guidelines, creating thorough documentation, and utilizing Laravel's integrated testing and debugging resources. By utilizing cloud-based infrastructure, the system minimizes its environmental impact by optimizing server utilization and decreasing the requirement for physical hardware. From an economic standpoint, the system provides a sales and warehouse management solution that lowers operating expenses while increasing efficiency. By offering an intuitive interface that lowers errors and boosts productivity, it improves the working circumstances for employees on a social level. The WMS with POS integration is a sustainable solution that will effectively assist businesses and communities for years to come because of its strong architecture, maintainability, low environmental impact, and future enhancement possibilities.

7.2 Social and Environmental Effects and Analysis

The system's ecological footprint is reduced to the least by virtue of its implementation on cloud infrastructure. Cloud services generally use renewable energy sources and optimize their energy use, which lowers the total carbon footprint related to data processing and storage. The decrease in the requirement for tangible hardware also results in a decrease in electronic waste, so promoting environmental sustainability. In addition, the system's enhanced inventory management, which guarantees ideal stock levels and minimizes overstocking or stockouts, con-

tributes to waste reduction. This efficiency not only reduces operating costs but also lessens the environmental effect of producing and disposing of excess inventory, which in turn supports sustainable business practices.

7.3 Addressing Ethics and Ethical Issues

Preventing illegal usage and preserving confidentiality can be achieved by limiting access to particular data to authorized personnel only. Furthermore, the system conforms to pertinent data protection laws, such the GDPR, to protect users' rights and guarantee legal compliance.

Accountability and transparency constitute yet another ethical consideration. By offering precise and lucid records of all transactions and inventory movements, which are auditable to guarantee honesty and integrity in business practices, the system fosters transparency. This openness aids in the prevention of fraud and other immoral activities in the sales and warehouse operations. Additionally, the system's user-friendly layout guarantees that staff members may utilize it efficiently without experiencing unnecessary stress or mistake, supporting an equitable and encouraging work atmosphere.

In terms of social responsibility, by boosting productivity and working conditions, the system supports morale among employees and promotes ethical labour practices. Additionally, it promotes sustainability by minimizing waste through inventory management optimization, which is consistent with moral principles for environmental care. The WMS with POS integration satisfies operational objectives and is consistent with wider social ideals of privacy, equity, and sustainability by tackling these ethical challenges in their whole.

Chapter 8

Lesson Learned

8.1 Problems Faced During this Period

Understanding the Project Requirement: I had trouble understanding the first prerequisite. I gradually developed expert knowledge of folder organization and coding.

Adapting to new technologies: I could not begin working until I had studied Laravel.

Meeting deadline: I took a while to get used to learning new technology because it was my first experience.

Identifying and Fixing Bugs: It was my first time studying new technology; therefore it took me some time to get used to it.

8.2 Solution of those Problems

Understanding the Project Requirement: I had trouble understanding the first prerequisite. I gradually developed expert knowledge of folder organization and coding.

Adapting to new technologies: I could not begin working until I had studied Laravel.

Meeting deadline: I took a while to get used to learning new technology because it was my first experience.

Identifying and Fixing Bugs: It was my first time studying new technology; therefore it took me some time to get used to it.

Chapter 9

Future Work & Conclusion

9.1 Future Works

Enhanced Security Features

More sophisticated security measures, including multi-factor authentication, biometric authentication, and frequent security audits, can be added to the system as cyber threats change. This would uphold user confidence and further protect sensitive data.

Customizable Modules

Allowing for more customizable modules within the system can cater to the specific needs of different businesses. This modularity would enable organizations to tailor the system according to their unique operational requirements. [6]

Sustainability Tracking

Adding features to track and report on sustainability metrics, such as carbon footprint and waste reduction, can help businesses adhere to environmental goals and regulations. [6]

9.2 Conclusion

To put it briefly, the culmination of my internship project involves successfully integrating a Point of Sale (POS) system and implementing a warehouse management system that is based on the Laravel framework. My technical skills have improved because of this project's meticulous design, rigorous development, and iterative improvement. It has also brought attention to how important problem-solving, and adaptation are in real-world situations. As the project draws to a close, I reflect on the invaluable lessons learned, the challenges surmounted, and the delight derived from creating a solution that satisfies real business needs. I'm ready to take on new tasks with passion and skill as I look ahead to the future with a deeper understanding of the difficulties involved in software development and a renewed sense of confidence in my abilities.

Bibliography

- [1] E. Žunić, S. Delalić, K. Hodžić, A. Beširević, and H. Hindija, “Smart warehouse management system concept with implementation,” in *2018 14th Symposium on Neural Networks and Applications (NEUREL)*, pp. 1–5, IEEE, 2018.
- [2] A. M. Atieh, H. Kaylani, A. Al-abdallat Yousef, Qaderi, L. Ghoul, L. Jaradat, and I. Hdairis, “Performance improvement of inventory management system processes by an automated warehouse management system,” *Procedia CIRP*, vol. 41, pp. 568–572, 2016.
- [3] E. Siami-Irdemoosa, S. R. Dindarloo, and M. Sharifzadeh, “Work breakdown structure (wbs) development for underground construction,” *Automation in Construction*, vol. 58, pp. 85–94, 2015.
- [4] W. b. J. Elliott, “Resource allocation: 5 step guides for project managers.” <https://example.com/resource-allocation-5-step-guides-for-project-managers>, 2020.
- [5] K. E. Kendall and J. E. Kendall, *Systems analysis and design*, vol. 2013. Pearson Prentice Hall Upper Saddle River, NJ, 2011.
- [6] A. M. Atieh, H. Kaylani, A. Al-abdallat Yousef, Qaderi, L. Ghoul, L. Jaradat, and I. Hdairis, “Performance improvement of inventory management system processes by an automated warehouse management system,” *Procedia CIRP*, 2016.



An Undergraduate Internship/Project on Falcon Apparel Warehouse Management System with POS System

By

Md. Farhan Soberin

Student ID: 1930403

Spring, 2024

The student modified the internship final report as per the recommendation made by his or her academic supervisor and/or panel members during final viva, and the department can use this version for achieving.

Signature of the Supervisor

Md. Abu Sayed

Senior Lecturer

Department of Computer Science & Engineering

School of Engineering, Technology & Sciences

Independent University, Bangladesh

1930403_Internship_Report_Spring24

ORIGINALITY REPORT



PRIMARY SOURCES

1	ar.iub.edu.bd Internet Source	3%
2	Submitted to Independent University Bangladesh Student Paper	2%
3	www.ijritcc.org Internet Source	<1 %
4	Submitted to University of Keele Student Paper	<1 %
5	Submitted to Westminster International University in Tashkent Student Paper	<1 %
6	Submitted to Asia Pacific Institute of Information Technology Student Paper	<1 %
7	Submitted to University of Greenwich Student Paper	<1 %
8	Submitted to University of Lancaster Student Paper	<1 %