A close-up of a logo

Description automatically generated

**School of Computing and Digital Technologies**

**Introduction to Software Engineering**

**(55-508876-AF-20245)**

**Descriptive Report**

**Task 1**

**Project:** (Business) Inventory Management System

**Group #: 16**

**Group Members: David, Odi, Igor, Abdirizak**

**Date: 05/01/2025**

|  |  |
| --- | --- |
| Name | ID |
| David Adegeloye | c2020232 |
| Odi Ehimuan | c3018521 |
| Igor Jankowski | c2053730 |
| Abdirizak Mumin | c1053702 |

# Introduction

This report provides a detailed account of our achievements and developments during the creation of an Inventory Management System for QuickBuy Ltd. We emphasize the processes, methodologies, and tools utilized to meet the specified requirements, highlighting the functional and architectural components.

We developed an Inventory Management System to solve operational challenges faced by QuickBuy Ltd., a multi-branch business. Our system was designed to streamline inventory management, generate actionable reports, and facilitate efficient over-the-counter sales. Our objectives focused on usability, data accuracy, accessibility, and responsiveness. To achieve these goals, we used the C4 architectural framework and derived our database schema and functionalities from well-defined user stories and domain requirements.

The project was motivated by the need to enable seamless coordination across various managerial roles within the organization, including store managers, warehouse managers, category managers, and sales associates. We created a unified system to streamline operations, minimize errors, and improve decision-making.

## Scope of work

**Development Methodology**

Our development process was guided by an iterative and incremental approach, aligning closely with Agile principles. This methodology allowed us to incorporate user feedback continuously and adapt to evolving requirements. The project began with a detailed requirement-gathering phase, followed by systematic design, implementation, testing, and deployment. Prototyping was conducted using tools like Figma, while GitHub was employed to ensure efficient version control and collaboration among team members.

During each iteration, deliverables were reviewed and refined to ensure alignment with the project’s objectives. Regular team meetings facilitated effective communication and coordination, enabling us to address challenges promptly.

**System Architecture**

We structured the system’s architecture using the C4 model, which provided a clear, structured view of the software components. The context diagram showed how users interacted with the system, highlighting roles such as store managers, warehouse managers, category managers, and sales associates. Each role was linked to specific system features to ensure a focused implementation.

The container diagram outlined the main systems, which included the user interface, back-end logic, and the database. The user interface was built using HTML, CSS, and JavaScript to ensure responsiveness and accessibility. The back-end logic was implemented in PHP, handling core functionalities such as authentication and data processing. Data persistence was achieved through a MySQL database, which was configured and managed using phpMyAdmin.

Component diagrams showed system elements clearly, emphasizing simplicity and reuse to support maintenance and future improvements.

**Functional Features**

The Inventory Management System was equipped with several core functionalities to meet the needs of QuickBuy Ltd. These functionalities were implemented as follows:

1. **User Authentication and Role Management**
   * A secure role-based authentication mechanism was developed. Users were assigned predefined roles: store managers, warehouse managers, category managers, and sales associates.
   * Password encryption was implemented to ensure data security, and validation mechanisms were established to prevent unauthorized access.
2. **Inventory Monitoring**
   * Real-time inventory updates were enabled using AJAX requests, allowing users to track stock levels and identify low-stock items efficiently.
3. **Sales and Reporting**
   * Sales data were visualized using interactive charts generated with Chart.js. Key performance indicators such as revenue and profit trends were displayed to support decision-making.
   * Reports could be generated for specific time ranges, providing detailed insights into sales performance and inventory turnover.
4. **Product Management**
   * Features to add, edit, and categorize products were integrated. Additionally, a CSV upload functionality was included to streamline bulk updates of product information.
5. **Invoicing and Notifications**
   * Dynamic invoice generation was implemented, with each invoice including itemized details, tax calculations, and total amounts.
   * A robust notification system ensured that critical updates, such as low-stock alerts and order statuses, were promptly communicated to users.

**Challenges and Solutions**

Several challenges were encountered during the development process. These challenges and our solutions are outlined below:

1. **Database Optimization**
   * **Challenge**: Slow query performance was observed as the dataset grew.
   * **Solution**: Indices were added to frequently queried columns, and query execution plans were analyzed and optimized.
2. **Cross-Browser Compatibility**
   * **Challenge**: Styling inconsistencies were noted across different web browsers.
   * **Solution**: A CSS reset was applied, and thorough testing on multiple browsers ensured consistent rendering and functionality.
3. **Real-Time Updates**
   * **Challenge**: Real-time inventory updates needed to be streamlined to ensure up-to-date stock tracking.
   * **Solution**: AJAX was used to dynamically refresh stock levels and notifications, removing the need for page reloads and enhancing the user experience.
4. **Scalability**
   * **Challenge**: Ensuring the system could handle increasing data volumes and user demands.
   * **Solution**: The architecture used simple, modular parts, allowing future features and increased workloads to be integrated without major redesigns.

**Accessibility and Usability**

Accessibility and usability were prioritized throughout the system’s design and development. High-contrast modes and keyboard navigation features were implemented to enhance the experience for users with visual impairments. Customizable color schemes were introduced to support colorblind users, improving the clarity of data visualizations.

Usability testing was conducted with representative end-users to gather feedback on navigation, workflow efficiency, and overall user experience. Based on the feedback, enhancements were made to ensure intuitive interfaces and seamless functionality.

**Ethical and Legal Considerations**

Data privacy and security were fundamental to the system’s design. Measures were taken to comply with GDPR, including encryption of sensitive user data, secure password storage, and strict access controls.

Ethical considerations included providing equal access to all users, regardless of disabilities or technical proficiency. Accessibility features were implemented to ensure inclusivity, reflecting our commitment to ethical software development.

**Conclusion**

Our Inventory Management System successfully addresses the operational challenges faced by QuickBuy Ltd., providing a reliable and effective solution for inventory tracking, sales reporting, and over-the-counter transactions. The system’s modular architecture, accessibility features, and comprehensive functionalities ensure scalability and usability.

While we faced challenges during the development process, we resolved them through teamwork and iterative refinement. We believe future enhancements, such as AI-driven analytics and advanced reporting capabilities, could make the system even more effective.

## Group Work Diary (Double Click on icon)



# User Stories, Personas, & Roles (Double Click on Link)



# Non-functional Requirements

***Performance:***

* The system must be accessible to all store locations
* Inventory updates and dashboard refreshes should happen in real-time to ensure accurate data representation.
* The system must ensure minimal latency, with page load times under 2 seconds under typical loads.
* Handle up to 4 concurrent users (Sales Associate, Store / Category / Warehouse Managers) without degradation of service.

***Security:***

* The system complies with GDPR and other relevant data protection regulations.
* Role-based access control (RBAC) ensures users only access data pertinent to their roles.
* Implement automated monitoring and alerts for unauthorized access attempts.

***Usability:***

* Ensure consistent design patterns across all modules to minimize the learning curve.
* Interfaces must support both mouse and keyboard navigation seamlessly.
* Support customizable dashboards to meet diverse user needs, such as visual impairments or workflow preferences.
* The system must be responsive across all devices → phones, tablets, PCs, laptops

# Design Artifacts:

## C4 Model Diagrams (L1,L2,L3)

A diagram of a system

Description automatically generated A diagram of a diagram

Description automatically generatedA diagram of a flowchart

Description automatically generated

## Use Case Diagram

A diagram of a diagram

Description automatically generated with medium confidence

## Sequence Diagram

A screenshot of a computer screen

Description automatically generated

## Data Design Diagrams (Class, ERD)

A diagram of inventory management system

Description automatically generatedA robust database schema was developed based on our Entity-Relationship (ER) diagram, which defined relationships among key entities such as users, roles, inventory, products, orders, and invoices. The database schema was designed to ensure data integrity and efficient performance through normalization, which minimized redundancy and improved consistency.

Indices and constraints were applied strategically to optimize query performance and enforce referential integrity. The schema included tables like Users, Products, Orders, and Invoices, with foreign key constraints ensuring accurate relationships between data entities. Sample data were added to support thorough testing and verification.

## User Interface Design (Wire Frames)

Prototype User Interface: <https://www.figma.com/proto/uZCnMOVmSZjizlBHcQRIzz/Inventory-Management-System-UI?page-id=1%3A2&node-id=303-75&node-type=canvas&viewport=71%2C297%2C0.19&t=R4DUeA5CbSkJH8Ij-1&scaling=min-zoom&content-scaling=fixed&starting-point-node-id=303%3A75>  
  
Inventory Management System SCRUM Sprint: <https://www.figma.com/board/0fzCTC7MLsCNzrEEQ3VQP4/Inventory-Management-System-SCRUM?t=oIeRxRPUd3MsiNOW-6>

# Testing Plan and Results (Double Click on Link)



Thorough testing was conducted to confirm the system’s functionality, reliability, and performance. Both manual and automated testing methods were utilized to ensure the application met its objectives. Functional testing involved verifying the correctness of features through test cases designed to cover various scenarios.

Integration testing ensured that modules interacted seamlessly, while system testing validated the application’s overall performance under real-world conditions. Performance testing was conducted using automated tools to simulate concurrent user activity and assess response times.

Some notable test cases include:

* **TC001**: Verified the registration process with valid user inputs. The system successfully stored the data in the database.
* **TC002**: Ensured the login functionality redirected users to their respective dashboards upon entering valid credentials.
* **TC008**: Confirmed the dashboard displayed accurate real-time updates for inventory and sales data.

Defects found during testing were recorded and addressed in order of priority. Regular testing ensured ongoing improvements and reduced the risk of errors reoccurring.

# Team reflection

The **team** can **briefly** reflect here on pros, cons, challenges faced, and lesson learned in relation to working as a team on the project. Avoid repeating points mentioned in the individual reports.

# Formative Feedback and Sprint Retrospective

**Feedback:**

1. **Expand Details on Features in the Group Diary**
   * Our teacher pointed out that we needed to provide more detailed descriptions of the features we are developing in our group diary. This would help ensure clarity and align our work with the project’s objectives.
2. **Make the Login Process Smoother and More Secure**
   * We were advised to improve the user experience of the login functionality and ensure it meets higher security standards.

**Actions Taken:**

1. **Expanding Feature Details in the Group Diary**
   * We reviewed our group diary and added more comprehensive descriptions for each feature, focusing on their purpose, expected outcomes, and technical implementation.
2. **Improving the Login Functionality**
   * We worked on optimizing the login process to make it faster and more user-friendly.

**Retrospective Insights:**

* Expanding the details in our group diary has made our work more transparent and helped the team align better with the sprint’s goals.
* The changes to the login process have not only improved the user experience but also strengthened security, which is critical for the success of our system.

# Peer Review Form (individual)

This is an individual work part. Use the template provided on Blackboard to evaluate your team members.

# References

Chart.js. (n.d.). *Simple, clean and engaging charts for designers and developers*. Retrieved from <https://www.chartjs.org>

European Parliament. (2016). *Regulation (EU) 2016/679 of the European Parliament and of the Council*. General Data Protection Regulation. Retrieved from <https://eur-lex.europa.eu>

phpMyAdmin. (n.d.). *phpMyAdmin: A free software for managing MySQL*. Retrieved from <https://www.phpmyadmin.net>

# Appendix 1: GitHub Repository

Link to Github: <https://github.com/iAmHxllow/Inventory-Management-System>

A screenshot of a computer program

Description automatically generated A screenshot of a computer

Description automatically generated

Table 4 - GitHub Usernames

|  |  |
| --- | --- |
| **Student name** | **GitHub name** |
| David | iAmHxllow |
| Odi | isharpz007 |
| Igor | ijank1111 |
| Abdirizak | c2036954 |