Project Proposal of Group 2: Small Business Inventory and Sales System

Dimas Stevano Ethanuel (24/532938/PA/22555)
Ryan Ethan Halim (24/536718/PA/22765)
Hamzah Rajwaa Abdillah (24/546622/PA/23207)
Department of Computer Science and Electronics, Faculty of Mathematics and Natural Sciences
Universitas Gadjah Mada
Yogyakarta, Indonesia

I. Introduction

This project aims to develop a comprehensive database management system for a small business online shop focused on efficient inventory tracking and sales management. The system will be designed using Python and MySQL, implemented over a 5-week development cycle by a team of three members. The Small Business Inventory and Sales System will provide automated solutions for managing product catalogs, tracking stock levels, recording sales transactions, and generating business insights through reports and analytics.

II. Problem Statement

Small online shop businesses often struggle with manual inventory management, leading to inefficiencies such as stockouts, overstocking, inaccurate sales records, and difficulty tracking business performance. Without a centralized system, business owners face challenges in maintaining real-time inventory visibility, managing customer orders efficiently, and making data-driven decisions. This project addresses these challenges by creating an integrated database system that automates inventory and sales processes, providing real-time data management and analytical capabilities.

III. Objective

The primary objectives of this project are: label=c.

1) To design and implement a robust relational database for managing inventory and sales operations.
2) To develop a web-based application that provides user-friendly interfaces for inventory management, sales recording, and order processing, with a backend utilizing Python and MySQL.
3) To enable real-time tracking of stock levels and automatic alerts for low inventory.
4) To generate meaningful reports and analytics for business decision-making.
5) To ensure data integrity, security, and efficient retrieval of business information.

IV. System and Requirements

A. Functional Requirements

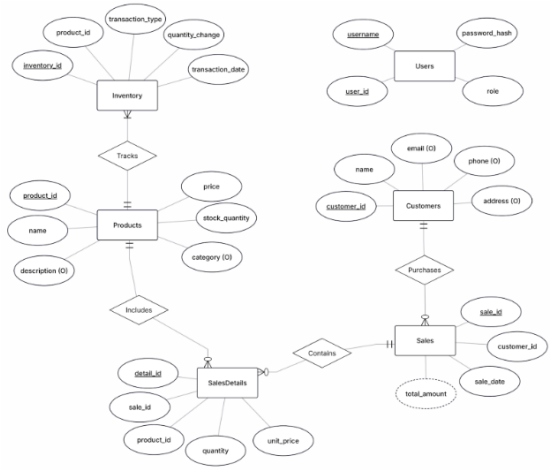
1) Product Management: Add, update, delete, and retrieve product information including name, description, price, and stock quantity.
2) Inventory Tracking: Real-time monitoring of stock levels with automatic updates upon sales.
3) Sales Management: Record sales transactions, process orders, and maintain sales history.
4) Customer Management: Store and manage customer information and purchase history.
5) Stock Alerts: Generate notifications when inventory levels fall below predefined thresholds.
6) Search and Filtering: Ability to search products by name, category, price range, or availability.
7) Reporting: Generate sales reports, inventory reports, and revenue analytics.
8) User Authentication: Secure login system for authorized access.

B. Non-Functional Requirements

1) Performance: System should respond to queries within 2 seconds.
2) Security: Encryption of sensitive data, role-based access control, and secure password storage.
3) Scalability: Design should accommodate growth to handle increased transactions and product inventory.
4) Data Integrity: Maintain referential integrity and prevent data inconsistencies.
5) Usability: Intuitive user interface requiring minimal training.

V. Database Design

A. Entity-Relationship Diagram



Additional note: *user_id* and *username* are both marked as unique; *user_id* is the primary key while *username* must be distinct.

B. Database Implementation Plan

The database will consist of the following main entities:

1) **Products:** Stores product details (*product_id, name, description, price, stock_quantity, category*).
2) **Customers:** Maintains customer information (*customer_id, name, email, phone, address*).
3) **Sales:** Records sales transactions (*sale_id, customer_id, sale_date, total_amount*).
4) **SalesDetails:** Links products to sales (*detail_id, sale_id, product_id, quantity, unit_price*).
5) **Inventory:** Tracks inventory movements (*inventory_id, product_id, transaction_type, quantity_change, transaction_date*).
6) **Users:** Manages system users for authentication (*user_id, username, password_hash, role*).

VI. Tools and Technologies

1) Database Management System: MySQL
2) Programming Language: Python
3) Framework: Flask
4) Development Environment: VS Code
5) Version Control: GitHub
6) Database Design Tool: ERDPlus

VII. Project Timeline & Deliverables

A. Milestones

1) Week 1, Define the problem domain, list system users, outline system objectives, and design the Entity Relationship Diagram (ERD).
2) Week 2, Convert ERD into a relational schema, normalize up to 3NF, and implement with SQL(CREATE TABLE, keys, constraints).
3) Week 3, Develop core CRUD features for the main entities (add, view, update, delete).
4) Week 4, Connect the database with a web frontend; implement forms and simple search or report functions.
5) Week 5, Complete documentation, finalize the system, and present to the class.

VIII. Expected Results

Upon successful completion, the project will deliver:

1) A fully functional relational database optimized for inventory and sales management.
2) A Python backend application providing server-side logic.
3) A website frontend allowing users to manage inventory, record sales, and access business reports.
4) Comprehensive database documentation including schema, relationships, and normalization details.
5) Test reports demonstrating system functionality and reliability.
6) User manual and technical documentation for system maintenance.
7) Presentation showcasing system capabilities and implementation details.

IX. Conclusion

The Small Business Inventory and Sales System represents a practical application of database management principles to solve real-world business challenges. This project will demonstrate the team's ability to design, implement, and deploy a functional database system while applying best practices in database normalization, security, and application development. The system will provide a foundation that can be extended with additional features and deployed for actual business use.