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. *Background*

The project's goal is to design and implement a database management system for small online businesses, focused on inventory tracking and sales management. The project uses ERDplus for the database's design and is implemented using Python and MySQL. The system aims to provide assistance for managing product catalogue and information, customer records, sales transactions, and activity logs. Additionally, the system is also designed to support core business processes such as maintaining user access information, monitoring stock levels, recording sales, and generating business insights.

II. *Problem Statement*

The lack of a centralized database system makes it difficult for small online businesses to manage their inventory and sales records. Many businesses might use spreadsheets or manual record-keeping, both of which are vulnerable to human error, oversights, redundancy, and inconsistent data. As businesses grow, so do the number of transactions they make, keeping these methods of record-keeping would be inefficient and difficult to manage, ultimately becoming unreliable.

Without an integrated database system, businesses would struggle to maintain accurate and up-to-date records. They might face issues such as false inventory counts, delayed transactions, missing records, and a risk of creating weak analytic reports. Expanding their operations will be a challenge if businesses continue working with manual record-keeping methods.

A reliable and structured database system that supports inventory tracking and sales management would ease the difficulties that may come. Such systems must be capable of centralizing business records, maintaining consistency, automating transaction processes, and providing analytical reports. By implementing an easy to use and well designed database system, small online businesses can improve their operational efficiency and support their long term growth.

III. *Objectives*

The objective of this project is to create a database management system that could support small online businesses in inventory tracking and sales management. Specific objectives for this project are:

1) Design and implement a relational database that does inventory and sales operations.
2) Develop a web-based application that can provide a user-friendly interface for inventory management, sales recording, and order processing, with a backend that uses Python and MySQL.
3) Activate real-time tracking of stock levels and automatic alerts for low inventory.
4) Generate accurate reports and analytics for business decision-making.
5) Ensures data integrity, security, and efficient retrieval of business information.

IV. *System Overview*

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 level with automatic updates upon sales.
3) Sales Management: Record sales transactions, process orders, and maintain sales history.
4) Customer Management: Stores and manages customer information and purchase history.
5) Stock Alerts: Generate notifications when inventory levels fall below predefined thresholds.
6) Search and Filtering: Add the ability to search products by name, category, price range, or availability.
7) Reporting: Generate the sales reports, inventory reports, and revenue analytics.
8) User Authentication: Secure a login system for authorized access.

B. Non-Functional Requirements

1) Performance: The system should respond to queries within2 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. *Methodology*

A. *System Analysis*

This phase identifies the functional and non-functional requirements of an inventory and sales management system. Business processes were analyzed to form the base of the database's design and application.

B. *Conceptual Database Design*

A conceptual model of the system was made using an Entity Relationship Diagram (ERD) which was created in ERDPlus. It defined the main entities of the system such as Users, Products, Categories, Customers, Sales, SalesDetails, and InventoryLogs. Entity relationships, cardinalities, and key attributes were specified to accurately represent real-world business interactions.

C. *Logical Database Design*

The ERD is then turned into a relational database schema, defining the tables along with their primary keys and foreign key constraints. Normalization principles were applied to reduce data redundancy and maintain data consistency. This schema served as the blueprint for the physical database implementation.

D. *Database and Application Implementation*

MySQL was used to implement the relational schema while the application layer was developed using Python. CRUD (Create, Read, Update, Delete) functionalities were implemented into the system, and an HTML web-based user interface was developed to make user interaction with the system easier.

E. *System Integration and Testing*

The frontend is integrated with the backend to guarantee user inputs and database operations are connected seamlessly. Tests were performed by using some operational scenarios to ensure accurate implementation database constraints, accurate transaction processing, and overall data integrity.

VI. *System Design*

The system's design is illustrated with an Entity Relationship Diagram (ERD) and a logical database schema. They form the base of the system and give a clear understanding of the database's structure and relationships.

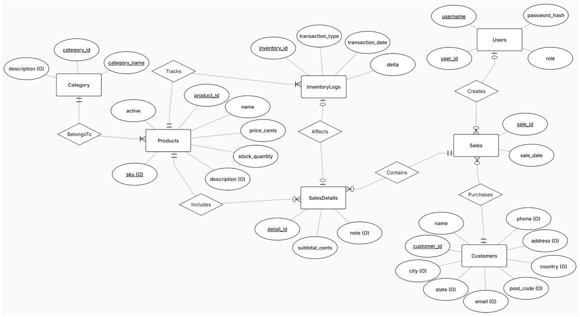

Figure 1. Entity Relationship Diagram of The System

The conceptual design of the system is illustrated by the ERD which identifies the main entities involved in inventory and sales management, along with their attributes and relationships. It shows how entities such as Users, Products, Categories, Customers, Sales, SalesDetails, and InventoryLogs interact with one another.

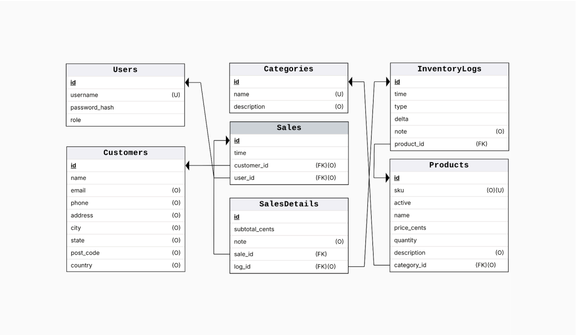

Figure 2. Logical Database Schema of the System

The ERD is then derived to a relational structure which is represented by the logical database schema. Tables, primary keys, and foreign key relationships that guarantee data consistency and integrity are defined. The schema serves as the basis for the physical database and makes sure that relationships between entities are appropriately maintained by converting the conceptual design into an implementable form.

VII. *Implementation*

A. *Database Implementation*

The database based on the logical schema, which was derived from the ERD, was implemented by using MySQL. Primary and foreign keys were clearly defined in the tables to maintain data consistency, data types were also made sure to match each attribute's nature. Constraints such as NOT NULL and foreign key references were used to prevent invalid data insertion.

B. *Application Layer Implementation*

User interactions were handled by using Python along with core application logic, including establishing database connections, executing SQL queries, and managing transactions. CRUD (Create, Read, Update, Delete) operations were implemented into the application, which was designed to handle real business processes such as automatically lowering stock numbers when a sales transaction is recorded.

C. *User Interface Implementation*

To enable users to interact with the system, a straightforward web-based user interface was created using HTML, it focuses on usability and clarity to make sure that basic operations could be performed efficiently. The interface provided means of entering product data, customer information, and sales transactions. User inputs from the interface were passed to the backend for processing and database storage.

D. *Transaction Handling and Data Integrity*

Transaction handling was implemented with caution in order for data integrity to be maintained. Inventory changes and sales records could be performed regularly since sales operations were handled as atomic transactions. Database rules and foreign key restrictions prevented orphan records and maintained valid relationships between tables.

E. *Deployment and Execution*

The system was executed in a local development environment where the MySQL database server and Python application ran together. The final implementation demonstrated end-to-end functionality, from user input through the web interface to database storage and retrieval. This confirmed that the system components were successfully integrated into a functional prototype.

VIII. *Conclusion*

This project successfully designed and implemented an inventory and sales management database system that can help small online businesses. The system's integrated application layer and well-structured relational schema allowed it to simulate actual business processes.

The Entity Relationship Diagram (ERD) and logical database schema ensured data consistency, integrity, and minimal redundancy. Implementing the database by using MySQL and Python made handling core operations easier. CRUD functionalities were implemented correctly along with database constraints, which enforced proper data relationships.

Frontend and backend integration allowed for immediate inventory updates and accurate transaction records. Several tests were done to make sure the system worked as intended, it maintained data integrity throughout transactions when performing under usual operational settings.

In summary, this project shows the real-life use of database design, implementation, and integration. The developed system is helpful for small online businesses, with useful features such as user authentication, reporting dashboards, and analytics that can support their decision-making as a business.