ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue VI, June 2026

Pos²: A Gui-Based Mini Point-of-Sale and Inventory Management System for a Mini Grocery Store

Devera, Freddie G. Ii., Balboa, Marky R., Banol, Air Humphrey B., Tablizo, Kier Franz D., Tenedero, Angelo D., Rafal, Ruzzel M., Engr. Fabro, Meshelle N.

Department of Computer Engineering, Eulogio "Amang" Rodriguez Institute of Science and Technology (EARIST) Nagtahan, Manila, Philippines

DOI: https://doi.org/10.51583/IJLTEMAS.2026.150600058

Received: 08 June 2026; Accepted: 13 June 2026; Published: 04 July 2026

ABSTRACT

Manual sales and inventory management in mini grocery stores often results in inaccurate calculations, delayed transactions, inconsistent stock monitoring, missing records, and difficulty in generating sales reports. To address these challenges, this study developed **POS²: A GUI-Based Mini Point-of-Sale and Inventory Management System for a Mini Grocery Store**. The system was designed to automate sales transactions, inventory monitoring, refund processing, product record management, activity logging, and sales report generation. It was developed using Python as the main programming language, Tkinter for the graphical user interface, and SQLite as the database management system. The system includes essential features such as PIN authentication, sales and checkout processing, product adding, viewing, updating, and deleting, stock adjustment, refund handling, activity log monitoring, and report generation. The SQLite database stores important information such as administrator accounts, inventory records, sales history, and activity logs, allowing data to remain available even after the application is closed. Through its graphical interface, users can manage store operations more easily without relying on handwritten records or manual computations. Testing showed that the system successfully performed its major functions with proper database integration. Overall, POS² provides a simple, organized, and efficient system that can help mini grocery stores improve transaction accuracy, inventory reliability, and business record management.

Keywords: POS², Point-of-Sale System, Inventory Management, Mini Grocery Store, Python, Tkinter, SQLite, Sales Management

INTRODUCTION

Background of the Study

Managing a mini grocery store requires proper handling of daily sales transactions, product inventory, customer purchases, refunds, and business records. Many small stores still depend on manual methods such as handwritten notebooks, calculators, and paper-based inventory lists. Although these methods may be useful for very small operations, they can become inefficient as the number of products and transactions increases.

Manual sales and inventory management may result in several problems, such as incorrect computations, missing transaction records, inaccurate stock counts, delayed sales processing, and difficulty in preparing reports. Store owners may also find it challenging to determine which products are still available, which items need restocking, and how much sales were earned during a specific period. These problems can affect business efficiency and may lead to poor decision-making.

With the help of computerized systems, sales and inventory management can become faster, more accurate, and more organized. A Point-of-Sale system can automatically compute totals, process sales, update

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue VI, June 2026

inventory, record transactions, and generate reports. It reduces the need for manual computation and helps prevent common human errors. Inventory management also becomes more reliable because product stock levels can be updated whenever items are sold or refunded.

This study focuses on the development of **POS²: A GUI-Based Mini Point-of-Sale and Inventory Management System for a Mini Grocery Store**. The system was designed to help mini grocery stores automate essential business operations such as product management, sales processing, refund handling, report generation, and inventory monitoring. It uses Python for the system logic, Tkinter for the graphical user interface, and SQLite for database storage.

The system also includes PIN authentication to protect important functions such as inventory management, refunds, and reports. This helps ensure that only authorized users can access sensitive system features. The database stores administrator credentials, inventory records, sales history, and activity logs. Through this, the system can keep organized and permanent records even after the program is closed.

Overall, POS² was developed to provide a simple and efficient digital solution for mini grocery store operations. It aims to reduce manual work, improve transaction accuracy, monitor stock levels, organize records, and support better business management through a user-friendly graphical interface.

Statement of the Problem

This study aims to develop **POS²**, a GUI-Based Mini Point-of-Sale and Inventory Management System that can automate sales and inventory management processes for a mini grocery store.

Specifically, this study seeks to answer the following questions:

1. How can POS² automate the sales and inventory management process of a mini grocery store?
2. How can the system store and retrieve product, sales, refund, and activity records efficiently?
3. How can the system reduce errors commonly encountered in manual sales and inventory management?
4. How can a graphical user interface improve user interaction and usability of the system?
5. How can the system help store owners monitor product availability, sales records, refunds, and activity logs effectively?

Objectives of the Study General Objective

The general objective of this study is to develop **POS²: A GUI-Based Mini Point-of-Sale and Inventory Management System for a Mini Grocery Store** using Python, Tkinter, and SQLite to automate sales transactions, inventory monitoring, refund processing, and sales report generation.

Specific Objectives

Specifically, this study aims to:

1. Design the system flow using a flowchart and structured process.
2. Develop the system using Python programming language.
3. Create a user-friendly graphical user interface using Tkinter.
4. Integrate an SQLite database for storing administrator accounts, inventory records, sales history, and activity logs.
5. Implement PIN authentication for secure access to sensitive system functions.
6. Automate sales transactions, checkout processing, and total computation.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue VI, June 2026

7. Provide product management functions such as adding, viewing, updating, and deleting product records.
8. Enable inventory monitoring by updating stock levels after sales and refunds.
9. Include refund processing and activity logging for better transaction tracking.
10. Generate sales and inventory reports to support business monitoring.
11. Test the functionality and usability of the system.

Significance of the Study

This study is significant because it provides a computerized solution for improving sales and inventory management in mini grocery stores. POS² can help reduce manual errors, speed up transactions, and improve record organization.

Store Owners. The system can help store owners monitor sales, inventory levels, refunds, and activity logs more efficiently. It can support better decision-making by providing organized records and sales reports.

Cashiers and Store Employees. The system can help employees process sales faster and more accurately. Instead of manually computing totals and updating stock records, they can use the system to perform transactions through a simple graphical interface.

Customers. Customers may benefit from faster transaction processing and reduced billing errors. A more organized checkout process can improve customer service and reduce waiting time.

Administrators and Managers. The system provides PIN authentication and activity logging, allowing administrators to control access to important functions and monitor system activities.

Students and Future Researchers. This study can serve as a reference for students and future developers who want to create similar systems related to point-of-sale applications, inventory management, database integration, and GUI-based software development.

SCOPE AND LIMITATIONS

This study focuses on the development of a desktop-based mini point-of-sale and inventory management system for a mini grocery store. The system is developed using Python programming language, Tkinter graphical user interface, and SQLite database integration.

The scope of the system includes PIN authentication login, sales transaction processing, inventory management, product record management, stock monitoring, refund processing, activity logging, report generation, and SQLite database storage. The system allows users to add, view, update, and delete products, process sales and refunds, update inventory stock, and generate sales-related records.

The system uses an SQLite database file to store important data such as administrator accounts, inventory information, sales history, and activity logs. This allows the records to remain saved even after the application is closed.

However, the system is limited to basic point-of-sale and inventory management functions only. It is intended for mini grocery store use and educational purposes. It does not yet support online transactions, cloud storage, barcode scanner integration, receipt printing, multi-user network access, or online payment systems. Future improvements may include barcode scanning, printable receipts, user role management, improved GUI design, and online payment integration.

REVIEW OF RELEVANT THEORY, STUDIES, AND LITERATURE

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue VI, June 2026

This chapter presents the relevant theories, studies, and literature that support the development of **POS²: A GUI-Based Mini Point-of-Sale and Inventory Management System for a Mini Grocery Store**. It discusses concepts related to point-of-sale systems, inventory management, computerized sales processing, database management, graphical user interface design, Python programming, SQLite database integration, PIN authentication, refund processing, activity logging, and related systems. These concepts provide the foundation for understanding the importance of developing a computerized system for managing sales and inventory in a mini grocery store.

Point-of-Sale System

A Point-of-Sale system, commonly known as POS, is a computerized system used to process sales transactions, compute totals, record purchases, and manage business records. In retail stores and grocery stores, POS systems are important because they help cashiers process customer purchases faster and more accurately. Instead of manually writing sales in notebooks and using calculators, a POS system automatically computes the total amount of selected items and records the transaction. In mini grocery stores, a POS system is useful because it supports the daily operation of selling products, monitoring sales, and updating inventory. When a product is sold, the system can automatically deduct the quantity from the available stock. This helps store owners determine which products are still available and which items need restocking.

POS² was developed to provide a simple point-of-sale solution for a mini grocery store. It includes sales transaction processing, checkout, refund handling, report generation, and inventory monitoring. These features help improve store operations by reducing manual work and minimizing errors in sales computation.

Inventory Management System

Inventory management is the process of monitoring, organizing, and controlling product stocks. It helps store owners know the quantity of available items, the products that need to be restocked, and the items that are frequently sold. Proper inventory management is important in grocery stores because products must always be available for customers.

Manual inventory tracking can lead to inaccurate stock counts, missing records, overstocking, or running out of important items. In small stores, owners often rely on notebooks or handwritten lists to track products. However, this method becomes difficult when there are many items and daily transactions.

A computerized inventory management system helps solve these problems by storing product details in a database. In POS², the inventory table stores the product name, price, and stock quantity. The system allows users to add, view, update, and delete products. It also updates stock levels whenever a sale or refund is processed. This makes inventory monitoring faster, more organized, and more accurate.

Computerized Sales and Inventory Management

Computerized sales and inventory management systems provide a better alternative to manual business records. Manual systems often depend on paper records, calculators, and handwritten notes. These methods are prone to human error, missing entries, inaccurate computations, and delayed reporting.

A computerized system improves store operations by automating important tasks. It can calculate transaction totals, update stock, save sales history, process refunds, and generate reports. This helps store owners and employees save time and reduce mistakes.

In POS², computerized processing is applied through the sales, inventory, refund, and report modules. The Sales module handles customer purchases and updates stock. The Inventory module manages product records. The Refund module processes returned items and restores stock. The Report module displays sales,

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue VI, June 2026

refunds, net sales, and activity records. These modules work together to make store management more efficient.

Database Management System

A database management system is used to store, organize, retrieve, update, and delete data. In system development, a database is important because it allows information to remain saved even after the program is closed. Without a database, records may be lost once the system stops running.

POS² uses SQLite as its database management system. SQLite is a lightweight and file-based database that stores data in a single database file. It is suitable for small to medium systems because it does not require a separate database server. In this system, the database file is named **grocery_system.db**.

The database contains important tables, including **admins**, **inventory**, **sales_history**, and **activity_log**. The admins table stores administrator login credentials. The inventory table stores product information such as product name, price, and stock. The sales_history table records sales and refund transactions. The activity_log table records administrative actions such as adding, updating, deleting products, and processing refunds.

Through SQLite, POS² can keep business records organized and permanent. It also improves reliability because product records, sales history, and activity logs remain available even after the application is closed.

Python Programming in System Development

Python is a programming language commonly used in system development because it is simple, readable, and flexible. It is widely used for developing desktop applications, database-connected programs, automation tools, and educational software projects.

In POS², Python serves as the main programming language. It handles the logic of the system, including user input, sales computation, inventory updates, refund processing, database connection, report generation, and activity logging. Python also allows the system to connect to SQLite through the built-in sqlite3 library.

Python supports structured programming by allowing the use of functions, loops, conditional statements, lists, dictionaries, and database commands. These programming concepts help organize the system and make the code easier to manage and improve. Through Python, POS² can perform different operations such as adding products, updating stock, deleting records, saving sales, and generating reports.

Graphical User Interface

A graphical user interface, or GUI, allows users to interact with a system through visual elements such as windows, buttons, labels, entry fields, menus, and dialog boxes. GUI-based systems are easier to use than command-line programs because users do not need to memorize commands. Instead, they can click buttons and input information through organized forms.

POS² uses Tkinter to create its graphical user interface. Tkinter is a Python library used for building desktop applications. Through Tkinter, the system provides a user-friendly interface for sales processing, inventory management, reports, refunds, and login authentication.

The GUI improves usability because cashiers and store owners can navigate the system more easily. The main menu provides access to different modules, while buttons and input fields guide users in performing tasks. This makes the system suitable for mini grocery stores where users may prefer a simple and easy-to-understand interface.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue VI, June 2026

PIN Authentication and System Security

System security is important in POS and inventory systems because not all users should have access to sensitive functions. Important operations such as inventory modification, refund processing, and report viewing should be limited to authorized users only.

POS² includes PIN authentication to protect sensitive system functions. When the user opens the application, the system asks for a PIN before allowing access. Some functions, such as Inventory, Reports, and Refunds, require an administrator PIN before they can be accessed. If the wrong PIN is entered, the system prevents access and returns the user to the main menu.

This feature helps protect business records and prevents unauthorized users from making changes to inventory or sales data. It also supports accountability because only authorized users can perform administrative actions.

Sales Transaction Processing

Sales transaction processing refers to the process of selecting products, entering quantities, computing totals, and saving the completed transaction. In manual systems, cashiers usually compute totals using calculators and write transactions in notebooks. This can result in incorrect totals, missing records, or slow checkout service.

In POS², sales transactions are automated. The user selects products, enters the quantity, and the system computes the total amount. After checkout, the system saves the sale in the database and updates the inventory stock. This ensures that product quantities remain accurate after each transaction.

Automated sales processing improves transaction speed and reduces billing errors. It also helps store owners maintain a complete record of sales transactions for future review and report generation.

Refund Processing

Refund processing is important in retail operations because customers may return products or request transaction adjustments. In manual systems, refunds may be difficult to track because they require changes in both sales records and inventory stock.

POS² includes a refund feature that records negative transaction amounts in the sales history table. Positive amounts represent completed sales, while negative amounts represent refunds. This allows the system to compute total sales, refunds, and net sales more accurately.

The refund feature also updates stock levels when products are returned. This helps maintain accurate inventory records and prevents confusion in product availability.

Activity Logging

Activity logging is the process of recording actions performed within the system. It is useful for monitoring administrative activities and maintaining accountability. In business systems, activity logs help identify what changes were made and when they were performed.

POS² includes an activity log table that records administrative actions such as adding products, updating products, deleting products, and processing refunds. This feature helps store owners monitor important changes in the system.

Activity logging improves transparency and control because administrators can review system activities. It also helps in checking errors or tracking changes made to inventory records.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue VI, June 2026

Related Studies

Manual product tracking and inaccurate record keeping are common issues in small grocery stores. Suya et al. (2026), in their study on smart inventory management for mini grocery stores, developed a PHP/MySQL system that automated inventory tracking and reporting. Their study showed that computerized inventory systems can improve efficiency and reduce stock errors.

Garados et al. (2026) found that POS systems improve transaction speed, inventory management, employee productivity, and customer satisfaction. Their study concluded that computerized systems perform better than manual systems in business operations.

Dimentieva et al. (2026) developed a web-based POS and inventory management system using Rapid Application Development and Design Thinking. Their system achieved a high user acceptance rating, showing that computerized POS systems can improve usability and operational efficiency.

Madamidola et al. (2024) emphasized that technologies such as barcode scanning, RFID, and IoT can improve inventory accuracy and reduce stock management errors. Although POS² does not yet include these advanced technologies, their study supports the importance of improving inventory systems through automation.

Ismail et al. (2024) developed a GUI-based inventory teaching application and found that graphical interfaces improve usability and understanding of inventory systems. This supports the use of a GUI in POS² because the system relies on visual interaction to help users manage sales and inventory tasks.

Trappey, Trappey, and Hwang (1996) developed a computerized retail system and found that computerized systems improve decision-making, service efficiency, and data organization. Their findings support the development of POS² as a digital solution for improving mini grocery store operations.

Related Systems

Several existing systems are related to POS². One example is a standard Point-of-Sale system used in supermarkets, convenience stores, and retail shops. These systems process sales, compute totals, update inventory, and generate receipts. POS² follows a similar concept but is designed for a smaller mini grocery store setting.

Another related system is an inventory management system. This type of system focuses on tracking stock levels, adding products, updating quantities, and preventing shortages or overstocking. POS² includes inventory management as one of its major features.

There are also mobile-based POS and inventory applications that allow store owners to manage products and sales using phones or tablets. These applications show how digital tools can help small businesses manage operations more efficiently. However, POS² is a desktop-based system developed mainly for educational purposes and local store use.

Synthesis of Reviewed Theory, Studies, and Literature

The reviewed theories, studies, and literature show that computerized POS and inventory management systems are useful in improving small business operations. Manual methods can cause errors, delays, missing records, inaccurate stock counts, and difficulty in generating reports. A computerized system helps solve these problems by automating sales processing, inventory monitoring, refund handling, and report generation

The literature also supports the use of Python, Tkinter, and SQLite in developing a small-scale business management system. Python provides the programming logic, Tkinter supports the graphical interface, and

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue VI, June 2026

SQLite provides permanent data storage. These tools work together to make POS² functional, organized, and easy to use.

Overall, the reviewed materials support the development of POS² as a practical solution for mini grocery stores. The system improves transaction accuracy, inventory reliability, record organization, and user convenience through a GUI-based and database-connected application.

CONCEPTUAL FRAMEWORK

The conceptual framework of this study follows the Input-Process-Output model. The input includes product information, product price, stock quantity, sales quantity, refund details, administrator username, and PIN. These data are entered by the user through the graphical user interface.

The process includes PIN authentication, sales computation, product management, inventory updating, refund processing, database storage, activity logging, and report generation. The system checks user input, performs calculations, updates records, and saves data in the SQLite database.

The output includes updated inventory records, completed sales transactions, refund records, sales reports, activity logs, and organized business records. These outputs help store owners and employees monitor store operations more effectively. Through this framework, POS² transforms raw sales and product data into organized and useful business information for mini grocery store management.

METHODOLOGY

This chapter presents the research design, system development process, system design, database design, tools and technologies used, system features, testing procedure, and evaluation method used in the development of **POS²: A GUI-Based Mini Point-of-Sale and Inventory Management System for a Mini Grocery Store**. It explains how the system was planned, designed, developed, tested, and implemented to provide a more organized and efficient way of managing sales transactions, inventory records, refunds, activity logs, and sales reports.

Research Design

This study used a system development approach. This design was appropriate because the main objective of the study was to develop a functional desktop-based point-of-sale and inventory management system for a mini grocery store. The system was developed to address common problems in manual sales and inventory management, such as inaccurate computations, slow transactions, missing sales records, incorrect stock counts, and difficulty in generating reports. Through the development of POS², the researchers aimed to provide a computerized system that can help store owners and employees manage products, sales, refunds, inventory, and reports in a faster and more organized way. POS² was developed using **Python** as the main programming language, **Tkinter** for the graphical user interface, and **SQLite** as the database management system. These tools were selected because they support desktop application development, user-friendly interface design, and local database storage.

System Development Process

The development of POS² followed several stages to ensure that the system would meet its intended purpose and function properly.

Planning

During the planning stage, the researchers identified the problems commonly encountered in manual grocery store operations. These problems included manual computation errors, delayed transaction processing, inaccurate product stock records, and difficulty in checking sales and refund history. The

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue VI, June 2026

researchers also identified the necessary system features, including PIN authentication, sales transaction processing, inventory management, product record management, refund processing, activity logging, and report generation. These features were selected to make the system useful for basic mini grocery store operations.

System Design

The system design stage involved creating the system flowchart, database structure, and graphical user interface layout. The flowchart served as the guide for the overall process of the system, from login authentication to the main menu and module selection. The system was designed to begin with a PIN login authentication. After successful login, the user is directed to the main menu, where the available modules can be accessed. These modules include Sales, Inventory, Reports, Refund, and Exit.

Development

The development stage involved coding and implementing the system using Python. Tkinter was used to create the graphical user interface, including windows, labels, entry fields, buttons, tables, and dialog boxes. The system functions were developed to support product management, sales transactions, inventory updates, refund processing, report generation, and activity logging. Python functions were used to organize the program into smaller and manageable parts. Conditional statements were used to validate user actions, while loops and database commands were used to process records and update stored data.

Database Integration

SQLite was integrated into the system to provide permanent storage for important records. The database file used in the system is named **grocery_system.db**. SQLite was chosen because it is lightweight, file-based, and suitable for small to medium-sized systems such as a grocery cashier application. The database stores administrator accounts, inventory records, sales history, and activity logs. This allows the system to preserve information even after the program is closed. Through database integration, the system can insert, retrieve, update, and delete records using SQL commands.

Testing

After the development stage, the system underwent functional testing. The researchers tested whether each feature worked according to its intended function. The testing process included valid and invalid login attempts, adding products, updating products, deleting products, processing checkout, handling refunds, and checking report generation. The purpose of the testing was to identify possible errors and verify that the system could perform its basic POS and inventory management functions correctly.

Implementation

After testing and debugging, POS² was implemented as a desktop-based mini point-of-sale and inventory management system. The final system allows users to process sales, manage product records, monitor inventory, process refunds, view reports, and maintain activity logs through a graphical interface connected to an SQLite database.

System Design

POS² starts with a PIN authentication screen. The user must enter the correct PIN before accessing the main menu. This login process helps secure the system and prevents unauthorized access to important business functions. After successful login, the system displays the main menu. The main menu serves as the central control panel of the system. From this menu, the user can access the Sales, Inventory, Reports, Refund, and

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue VI, June 2026

Exit modules. The **Sales Module** allows users to select products, enter quantities, add items to the cart, compute totals, and complete checkout. Once a sale is completed, the system updates the inventory stock and saves the transaction record.

The **Inventory Module** allows authorized users to add, view, update, and delete product records. Product information includes product name, price, and stock quantity. This module helps store owners monitor product availability.

The **Reports Module** displays sales information, refunds, net sales, and activity logs. This helps store owners review business transactions and monitor system activities.

The **Refund Module** allows users to process returned products. When a refund is processed, the system records the refund and restores the product stock. The **Exit Module** allows the user to close the system safely.

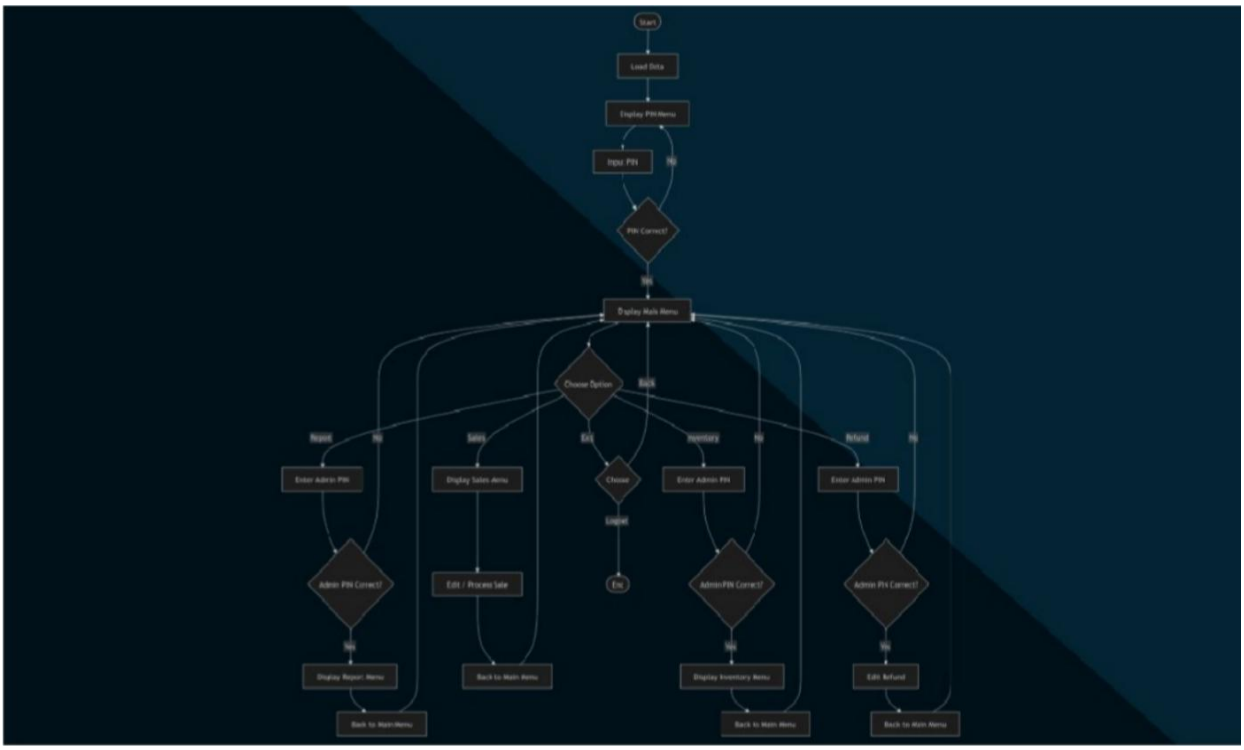


Figure 1. Flowchart of Mini Point-of-Sale and Inventory Management System

This figure shows the overall process flow of the POS² system. It begins with data loading and PIN authentication before allowing access to the main menu. From the main menu, users can access the Sales, Inventory, Reports, and Refund modules. The flowchart also shows that some modules require administrator PIN verification before proceeding, ensuring that sensitive functions are protected from unauthorized access.

Database Design

The system uses SQLite as its database management system. The database file is named **grocery_system.db**. It contains several tables that organize data according to their specific functions.

Table Name **Description**

admins Stores administrator login credentials such as username and PIN.

Inventory Stores product information such as product name, price, and stock quantity.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue VI, June 2026

sales_history Stores sales and refund transactions. Positive amounts represent sales, while negative amounts represent refunds.

Activity_log Stores descriptions of administrative actions such as adding, updating, deleting products, and processing refunds.

The **admins** table is used for authentication. It helps verify whether a user has permission to access sensitive functions such as inventory, reports, and refunds.

The **inventory** table stores product records. The product name serves as the main identifier to avoid duplicate product entries. The price field stores the product price, while the stock field stores the available quantity.

The **sales_history** table records financial transactions. Sales are stored as positive values, while refunds are stored as negative values. This allows the system to compute total sales, total refunds, and net sales.

The **activity_log** table records important system actions. This helps store owners monitor changes made in the system and improves accountability.

Python connects to the SQLite database using the built-in sqlite3 library. The system uses SQL commands such as INSERT, SELECT, UPDATE, and DELETE to manage data. The commit() function is used to save changes permanently, while the connection is closed after completing each database operation.

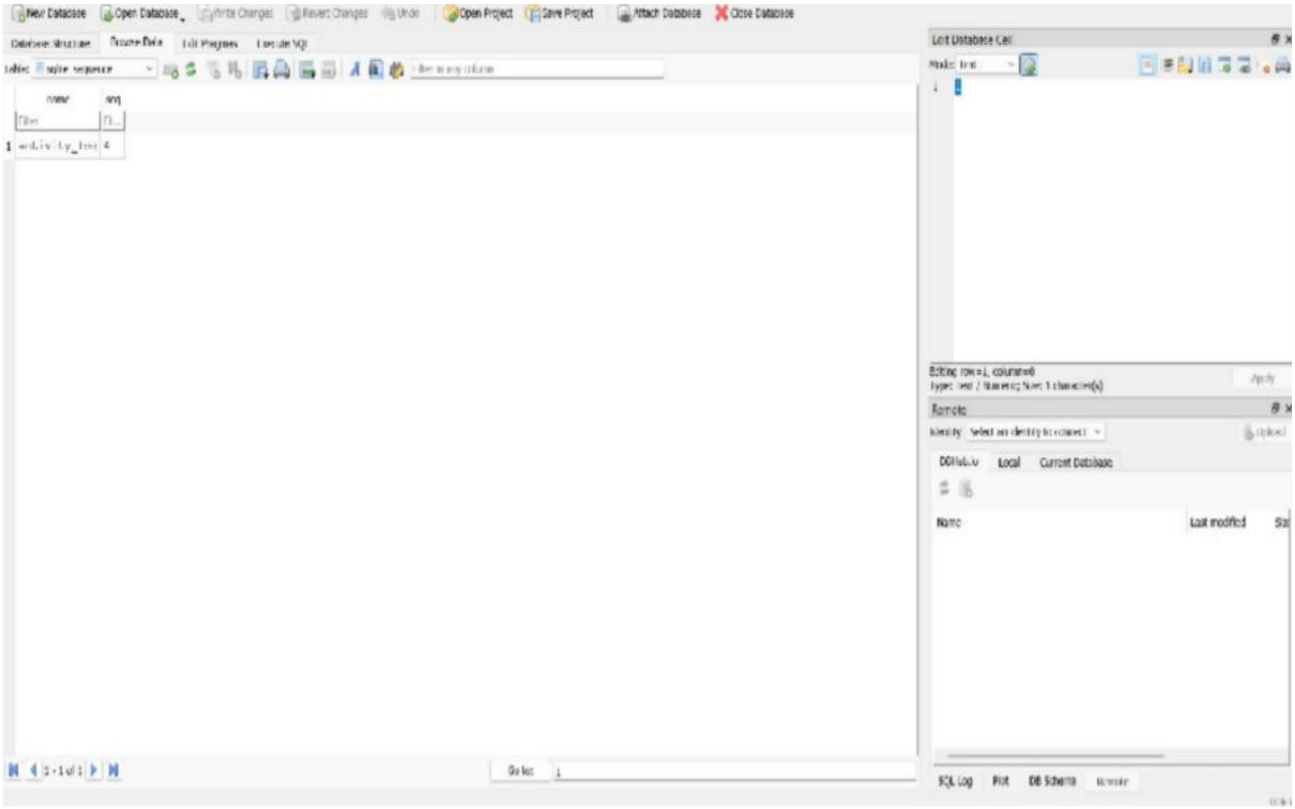
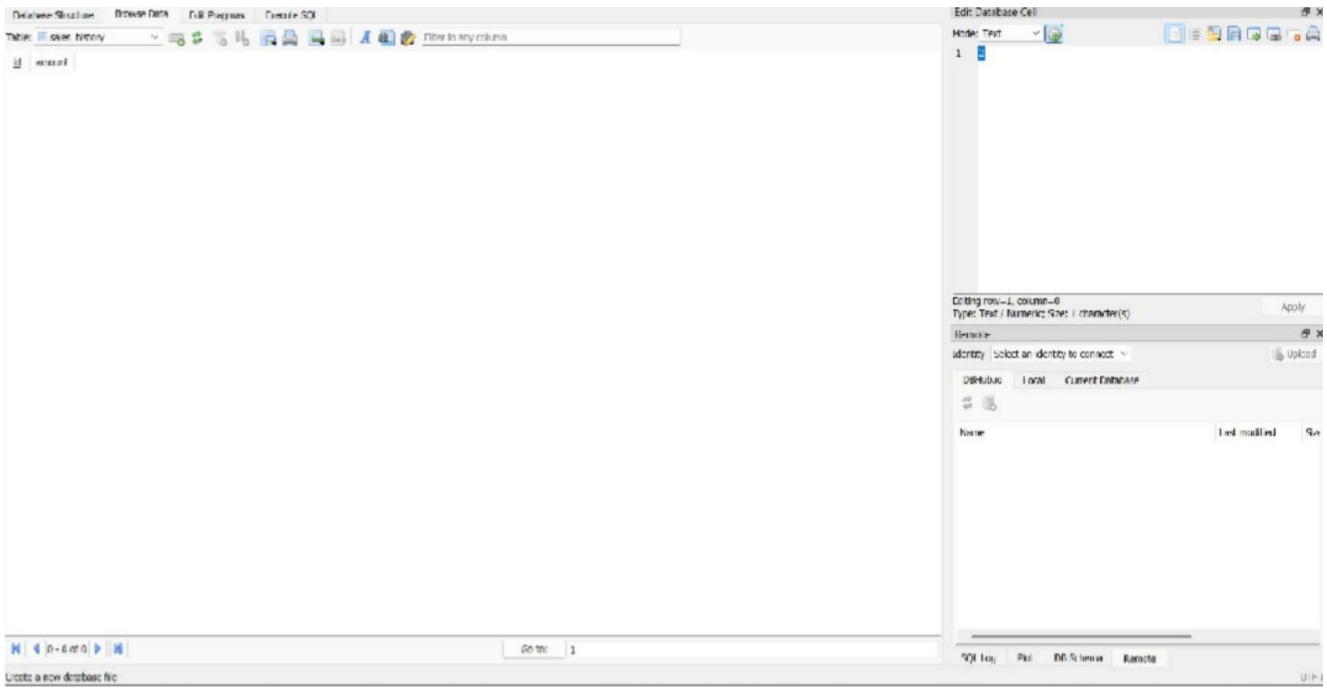


Figure 2. SQLite Auto-Increment Record of the POS² Database

This figure shows the `sqlite_sequence` table in the POS² SQLite database. It records the latest automatic sequence number used by the system, particularly for the `activity_log` table, ensuring that each saved activity has a unique record ID.

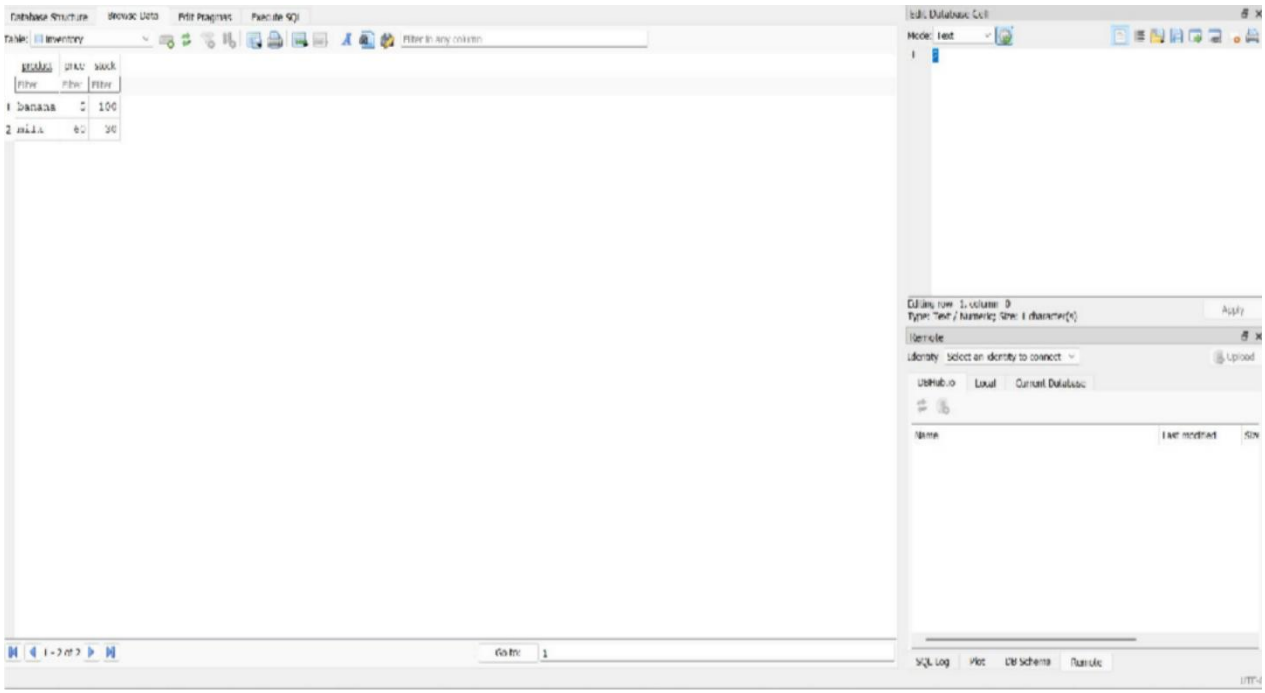
Figure 3. SQLite Sales History Table of the POS² Database

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue VI, June 2026



This figure shows the `sales_history` table in the POS² SQLite database. The table is designed to store sales and refund transaction records, including the transaction ID and amount. This table supports the system's report generation by keeping track of sales, refunds, and net sales.

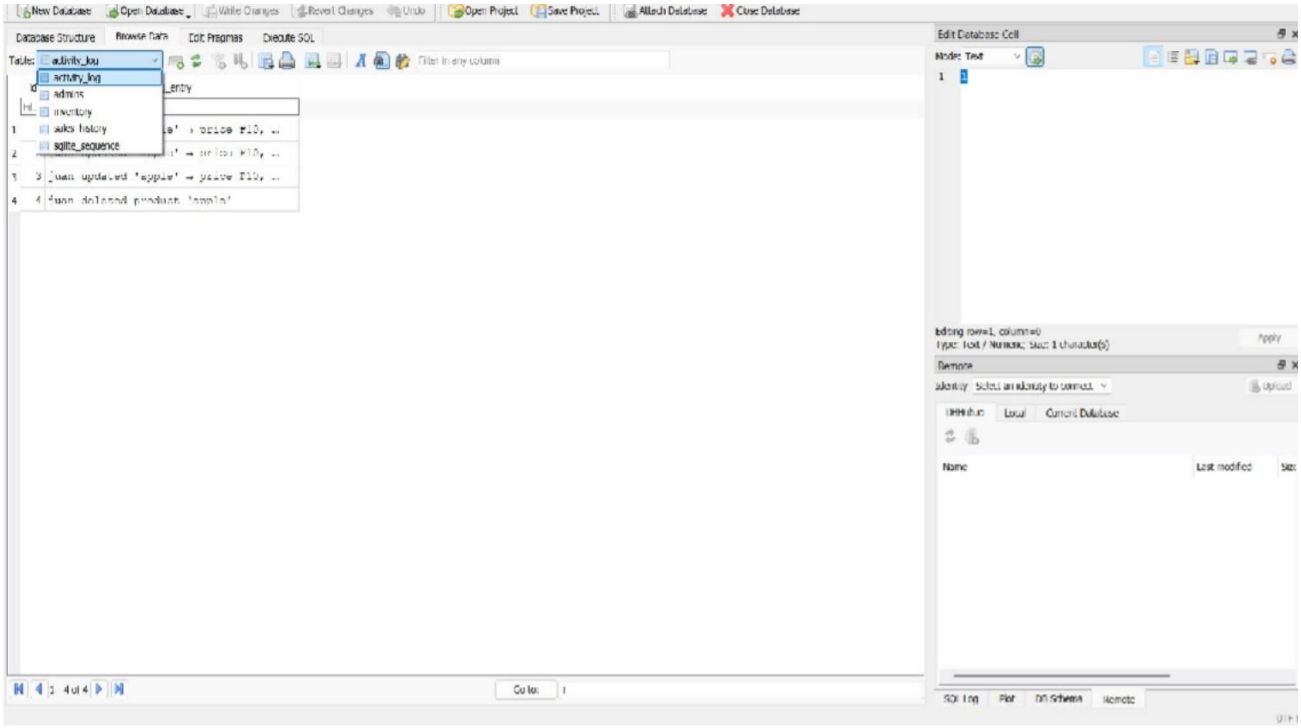
Figure 4. SQLite Inventory Table of the POS² Database



This figure shows the `inventory` table in the POS² SQLite database. The table stores product records such as product name, price, and stock quantity. It supports the inventory management function by allowing the system to save, update, and monitor available grocery items.

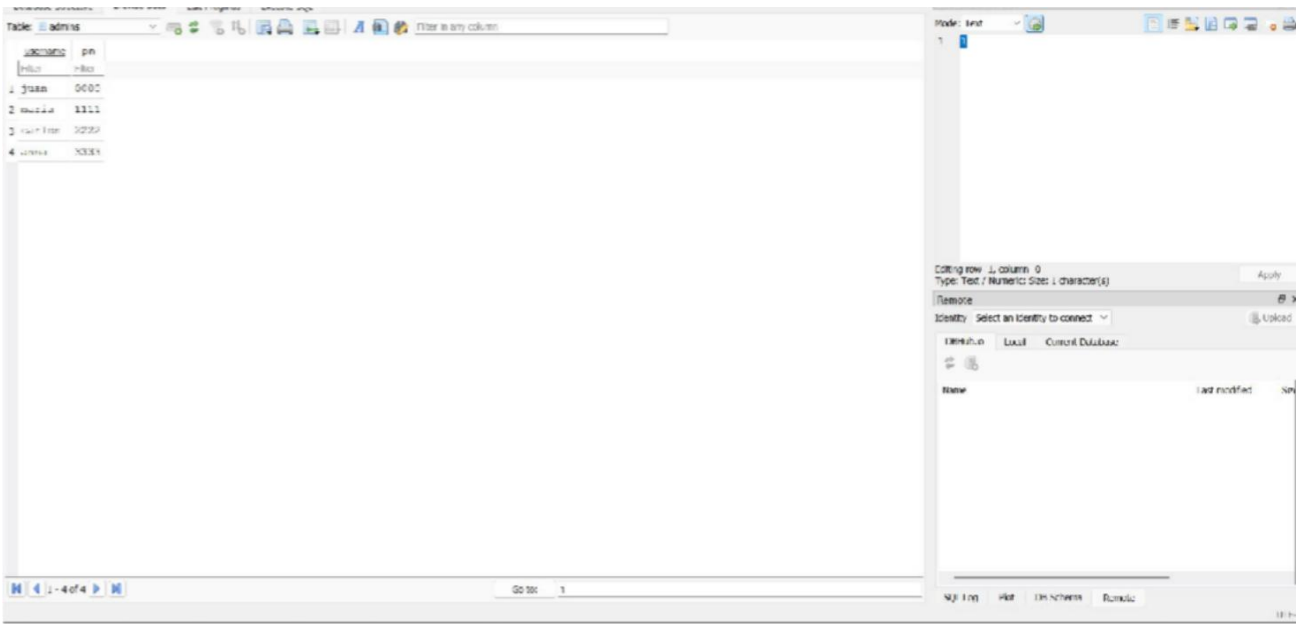
Figure 5. SQLite Activity Log Table and Database Tables of the POS² System

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue VI, June 2026



This figure shows the `activity_log` table and the list of database tables used in the POS² system, including `admins`, `inventory`, `sales_history`, and `sqlite_sequence`. The activity log records important system actions such as adding, updating, and deleting products, which helps monitor user activity and maintain accountability in the system.

Figure 6. SQLite Admin Accounts Table of the POS² Database



This figure shows the `admins` table in the POS² SQLite database. The table stores administrator usernames and PINs used for system authentication. This supports the security feature of the system by allowing only authorized users to access protected modules such as inventory, reports, and refunds.

Tools and Technologies Used

The following tools and technologies were used in developing POS²:

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue VI, June 2026

Python was used as the main programming language. It handled the system logic, sales computation, inventory updates, refund processing, database operations, and report generation.

Tkinter was used to create the graphical user interface. It provided the buttons, windows, input fields, tables, menus, and dialog boxes used in the system.

SQLite was used as the database management system. It stored administrator accounts, product records, sales history, and activity logs in a local database file.

VS Code / Python IDE was used as the development environment for writing, editing, testing, and debugging the program code.

System Features

POS² includes the following features:

PIN Authentication Login allows the user to access the system after entering the correct PIN.

Sales and Checkout System allows users to select products, enter quantities, add items to the cart, compute totals, and complete sales transactions.

Inventory Management allows authorized users to add, view, update, and delete product records.

Stock Management automatically updates product stock after sales and refunds.

Refund Processing allows users to process returned items and restore stock quantity.

Sales Report Generation displays sales, refunds, net sales, and other transaction summaries.

Activity Logging records administrative actions such as adding, updating, deleting products, and processing refunds.

SQLite Database Storage saves administrator accounts, inventory records, sales history, and activity logs permanently.

Testing Procedure

The system was tested using functional testing. Each major feature was tested to determine whether it produced the expected result. Both valid and invalid actions were tested to check whether the system could handle user input properly.

The following test areas were included:

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue VI, June 2026

Test Area	Purpose
Valid Login	To check if the system allows access when the correct PIN is entered.
Invalid Login	To check if the system blocks access when an incorrect PIN is entered.
Add Product	To check if a new product can be added to the inventory.
Update	To check if product details such as price and stock can be modified.
Product Delete	To check if a selected product can be removed from the inventory.
Product Sales	To check if the system can process customer purchases and update stock. To check if the system can record refunds and restore stock.
Checkout	To check if the system can display sales, refunds, net sales, and logs.
Refund Processing	
Report Generation	
Database Connectivity	To check if the system can save and retrieve data from SQLite.

The testing process helped confirm whether POS² could perform its intended functions and maintain proper database integration.

Evaluation of the System

The system was evaluated based on functionality, accuracy, usability, reliability, and data management. Functionality was evaluated by checking whether the system could successfully perform login authentication, product management, sales processing, refund handling, report generation, and activity logging.

Accuracy was evaluated by checking whether the system computed sales totals correctly and updated product stock after sales and refunds.

Usability was evaluated based on the clarity and simplicity of the graphical user interface. The use of buttons, tables, menus, and input fields made the system easier to operate.

Reliability was evaluated by checking whether data remained saved in the SQLite database after the system was closed. Data management was evaluated by checking whether the system properly stored and organized administrator accounts, inventory records, sales history, and activity logs.

Summary

The methodology of this study followed a system development process consisting of planning, system design, development, database integration, testing, and implementation. POS² was developed using Python, Tkinter, SQLite, and VS Code or Python IDE. Through this methodology, the researchers were able to create a functional GUI-based Mini Point-of-Sale and Inventory Management System for a mini grocery store. The system supports PIN authentication, sales processing, inventory management, stock monitoring, refund processing, report generation, activity logging, and SQLite database storage. It demonstrates the practical application of programming, graphical user interface design, database management, and system testing in a small business setting.

RESULTS AND DISCUSSION

This chapter presents the results and discussion of the developed **POS²: A GUI-Based Mini Point-of-Sale and Inventory Management System for a Mini Grocery Store**. It discusses the system overview,

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue VI, June 2026

completed system features, graphical user interface, testing results, and overall performance of the system. The results show how POS² was able to automate sales transactions, inventory management, refund processing, report generation, activity logging, and database storage using Python, Tkinter, and SQLite.

System Overview

POS² was developed as a desktop-based point-of-sale and inventory management system for a mini grocery store. The system was designed to help store owners and employees process sales, monitor inventory, manage products, handle refunds, and generate reports in a faster and more organized way.

The system uses **Python** as the main programming language, **Tkinter** for the graphical user interface, and **SQLite** as the database management system. The database file, named **grocery_system.db**, stores important records such as administrator accounts, inventory products, sales history, and activity logs. This allows records to remain saved even after the application is closed.

The system begins with PIN authentication before the user can access important functions. After successful login, the user can proceed to the main menu, where the system modules are available. These modules include Sales, Inventory, Reports, Refund, and Exit. Through these modules, users can complete daily store operations without relying on handwritten records and manual calculations.

Overall, POS² provides a simple but functional system that improves transaction accuracy, stock monitoring, and record organization for a mini grocery store.

System Features and Results PIN Authentication

The PIN authentication feature serves as the first layer of system security. The user must enter the correct PIN before accessing the system. This feature helps prevent unauthorized access to important functions such as inventory management, refunds, and reports. The result showed that the system successfully allowed access when the correct PIN was entered. If an incorrect PIN was entered, the system denied access and prevented the user from proceeding to restricted modules. This helped protect sensitive business records and administrative functions.

Sales and Checkout Processing

The Sales module allows users to select products, enter quantities, add items to the cart, compute totals, and complete checkout. The system automatically calculates the total amount of the transaction based on product price and quantity. The result showed that the system successfully processed customer purchases. After checkout, the transaction was saved in the sales history table and the inventory stock was automatically updated. This reduced the need for manual computation and helped prevent stock count errors.

Inventory Management

The Inventory module allows authorized users to add, view, update, and delete product records. Product details include product name, price, and available stock quantity. The result showed that the system successfully managed inventory records. New products were added to the SQLite database, existing product details were updated correctly, and deleted products were removed from the inventory list. This feature helped make stock monitoring more organized and efficient.

Stock Management

The system automatically updates product stock after sales and refunds. When a product is sold, the quantity is deducted from the inventory. When a refund is processed, the returned quantity is added back to the inventory.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue VI, June 2026

The result showed that the system successfully maintained updated stock levels. This helped ensure that the inventory records reflected the actual movement of products in the store.

Refund Processing

The Refund module allows users to process returned items. When a refund is made, the system records the refund in the sales history and restores the product quantity to the inventory.

The result showed that refunds were successfully recorded as negative transactions in the database. This allowed the system to compute total sales, total refunds, and net sales more accurately.

Sales Report Generation

The Reports module displays transaction summaries, including sales, refunds, net sales, and activity records. This feature helps store owners monitor business performance and review store activities.

The result showed that the system successfully generated reports based on stored records. The reports helped users view important business information without manually checking each transaction.

Activity Logging

The Activity Log feature records important administrative actions, such as adding products, updating inventory, deleting products, and processing refunds.

The result showed that the system successfully recorded activities in the activity log table. This improves accountability because store owners can monitor what changes were made in the system.

System Interface

POS² uses a graphical user interface developed using Tkinter. The interface includes windows, buttons, input fields, tables, menus, and message boxes that allow users to interact with the system easily.

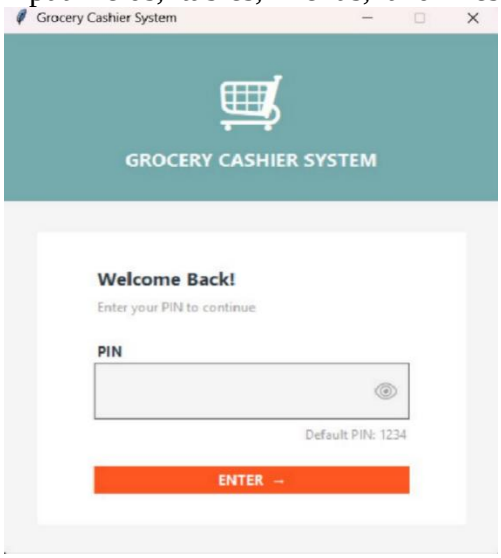


Figure 4.1. PIN Authentication Window

This window allows the user to enter the correct PIN before accessing the system. It helps protect important system modules from unauthorized access.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue VI, June 2026

Main Menu Window

The Main Menu serves as the central navigation area of the system. It provides access to the Sales, Inventory, Reports, Refund, and Exit modules.

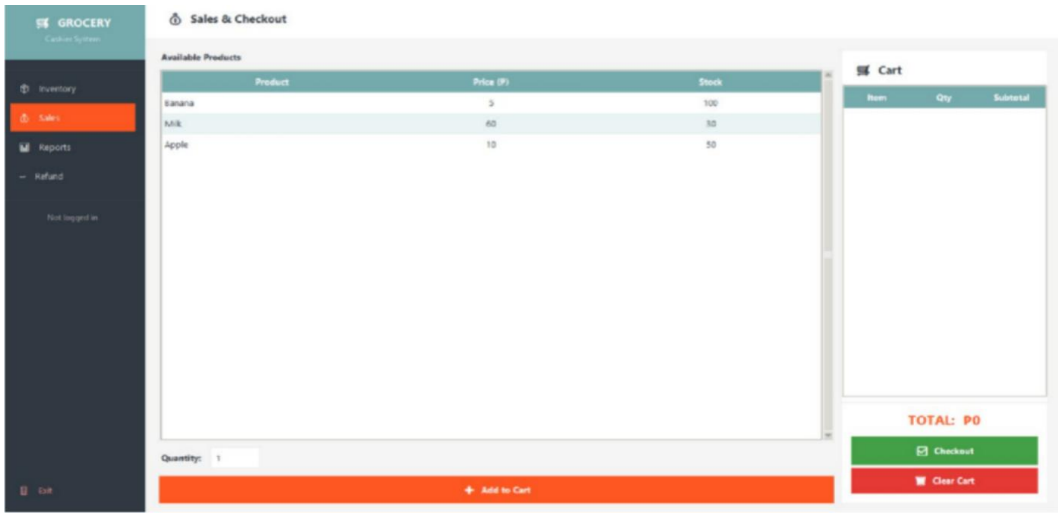


Figure 4.3. Sales Module

The Sales module allows users to select products, enter quantities, add items to the cart, compute totals, and complete checkout.

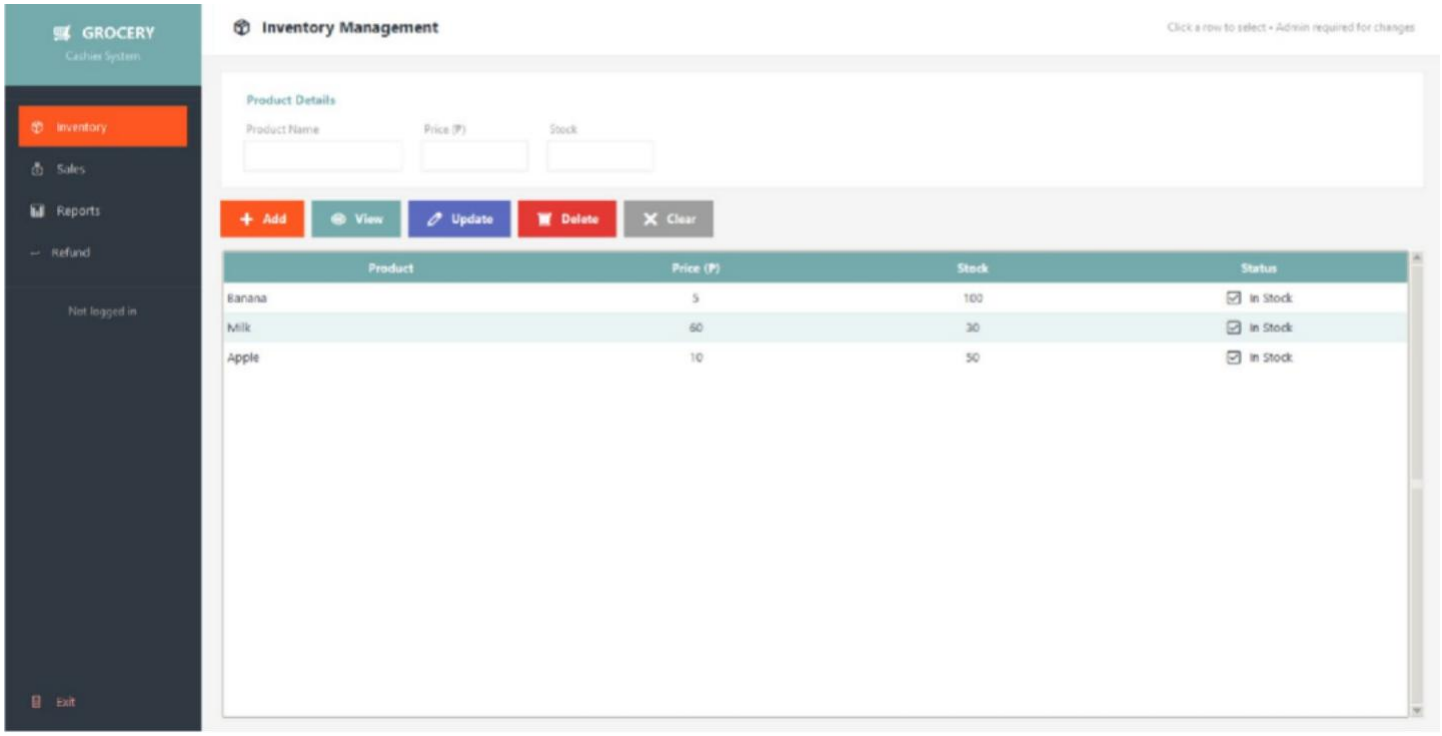


Figure 4.4. Inventory Management Window

The Inventory window allows users to add, view, update, and delete product records. It displays product information such as product name, price, and stock quantity.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue VI, June 2026

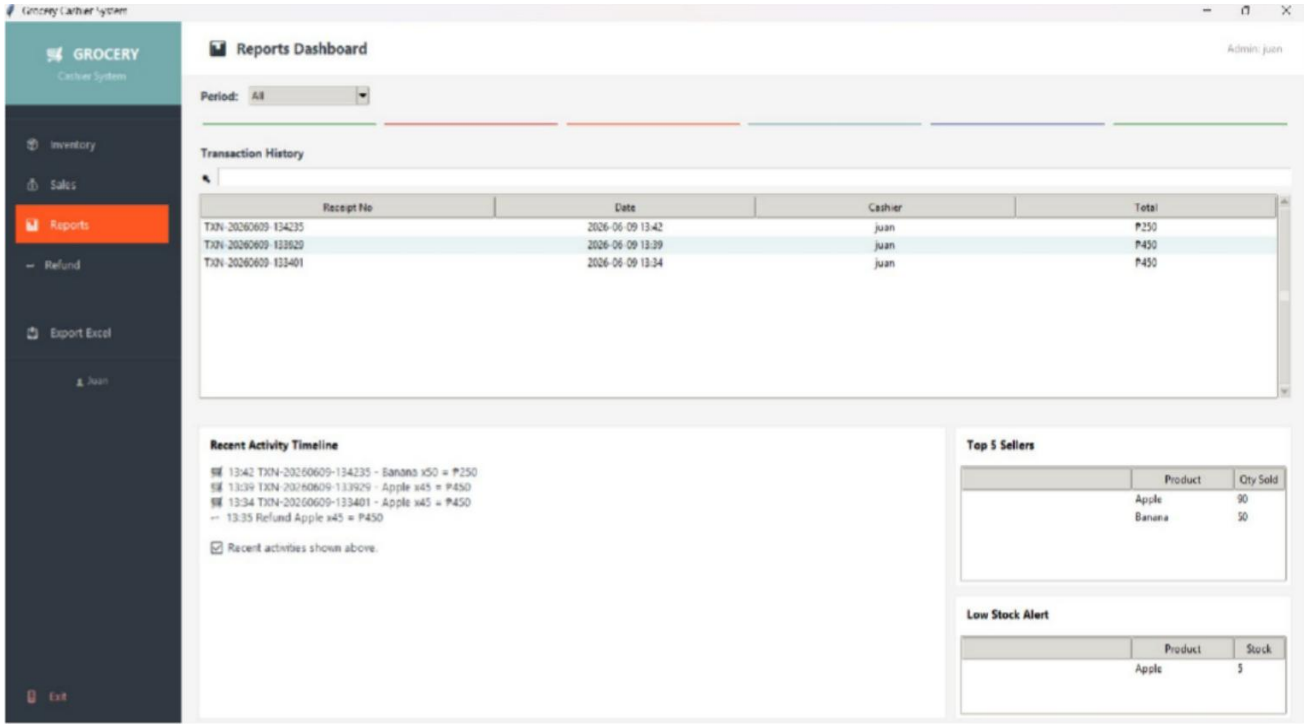


Figure 4.5. Reports Window

The Reports window displays sales records, refunds, net sales, and activity logs. This helps store owners monitor transactions and system activities.

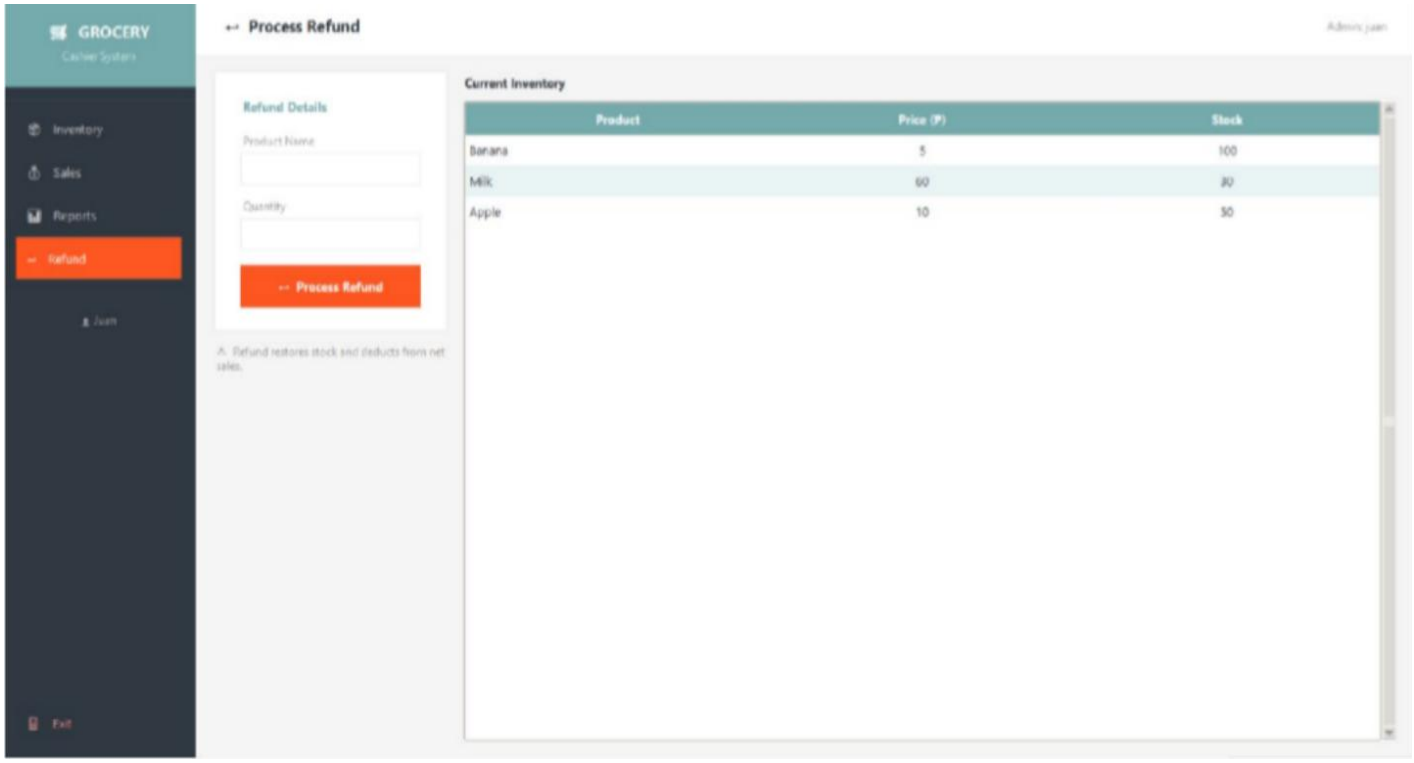


Figure 7. Refund Interface of the POS² System

This figure shows the Refund Interface of the POS² system, where authorized users can process returned items. The interface allows the user to select or enter the product to be refunded, record the refund transaction, and update the inventory stock. This feature helps maintain accurate sales history, refund records, and product quantity in the database.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue VI, June 2026

The graphical interface improved system usability because users can perform tasks through buttons and forms instead of typing command-line instructions. This makes the system more convenient for mini grocery store users.

System Testing Results

The system was tested using functional testing. Each major feature was tested to determine whether it produced the expected result. Both valid and invalid actions were tested to check system accuracy, reliability, and database integration.

Test Case	What Was Done	Expected Result	Actual Result	Status
Valid PIN Login	Entered the correct PIN	The system should allow access to the main menu.	The system successfully opened the main menu.	PASS
Invalid PIN Login	Entered an incorrect PIN	The system should deny access.	The system displayed an error message and blocked access.	PASS
Add Product	Entered a new product with price and stock	The product should be saved in the inventory database.	The product was saved successfully.	PASS
View Inventory	Opened the Inventory module	The system should display all product records.	All products were displayed in the inventory list.	PASS
Update Product	Changed product price or stock	The system should update the selected product.	The product record was updated successfully.	PASS
Delete Product	Selected and deleted a product	The system should remove the product from inventory.	The selected product was deleted successfully.	PASS
Sales Checkout	Selected product and entered quantity	The system should compute total and save transaction.	The sale was completed and saved in sales history.	PASS
Stock Deduction	Completed a sales transaction	The system should deduct sold quantity from inventory.	Stock was updated correctly.	PASS
Refund Processing	Processed a returned item	The system should record the refund and restore stock.	Refund was recorded and stock was restored.	PASS
Report Generation	Opened the Reports module	The system should display sales, refunds, and net sales.	Reports were displayed successfully.	PASS
Activity Logging	Performed admin actions	The system should record the actions in the activity log.	Actions were saved in the activity log table.	PASS
Exit System	Clicked the Exit button	The system should close safely.	The system closed without data loss.	PASS

The testing results showed that POS² successfully performed its intended functions. The system was able to authenticate users, manage products, process sales, update inventory, handle refunds, generate reports, and record system activities.

Discussion of Findings

The results show that POS² successfully addressed the common problems encountered in manual sales and inventory management. Manual methods often lead to inaccurate calculations, missing records, delayed transactions, and incorrect stock counts. Through POS², these problems were reduced because the system automated important store operations.

The use of SQLite improved the reliability of the system because all records were stored permanently in the local database. Product records, administrator accounts, sales history, and activity logs remained saved even after the program was closed. This made the system more dependable than manual notebooks or temporary data storage.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue VI, June 2026

The sales processing feature improved transaction accuracy by automatically computing totals and recording sales. This reduced the possibility of billing errors and helped speed up checkout transactions. The automatic stock deduction also improved inventory monitoring because product quantities were updated every time a sale was completed.

The refund feature also contributed to accurate business records. By recording refunds as negative transactions and restoring stock, the system was able to maintain more accurate sales and inventory data.

The activity log helped improve accountability by recording important administrative actions. This allows store owners to review system changes and monitor how the system is being used.

The graphical user interface also improved usability. Since the system uses buttons, input fields, tables, and message boxes, users can operate the system more easily compared to command-line programs or manual records.

Overall, the findings show that POS² is functional, organized, and useful for basic mini grocery store operations.

Strengths of the System

POS² has several strengths based on the results of development and testing. First, it provides PIN authentication to protect important functions. Second, it automates sales transactions and total computation. Third, it updates inventory stock after sales and refunds. Fourth, it stores data permanently using SQLite. Fifth, it generates reports for sales, refunds, and net sales. Sixth, it records administrative actions through activity logging. Lastly, it uses a graphical interface that makes the system easier to operate.

These strengths show that POS² can help mini grocery stores improve transaction processing, inventory monitoring, and record management.

Limitations Observed

Although POS² successfully performed its intended functions, some limitations were observed. The system is intended for local desktop use only and does not yet support cloud storage or online access. It also does not include barcode scanner integration, receipt printing, online payment options, or multi-user network access. The system is also designed mainly for basic mini grocery store operations. It may need further improvement if it will be used by larger stores with more complex inventory and sales requirements.

Summary

The results showed that POS² successfully performed its major functions, including PIN authentication, sales checkout, product management, stock monitoring, refund processing, report generation, activity logging, and SQLite database storage. The discussion showed that the system improved the accuracy, speed, and organization of sales and inventory management. Through its graphical interface and database integration, POS² provides a practical and user-friendly solution for managing the basic operations of a mini grocery store.

CONCLUSIONS AND RECOMMENDATIONS

Conclusions

Based on the development, testing, and evaluation of POS²: A GUI-Based Mini Point-of-Sale and Inventory Management System for a Mini Grocery Store, the researchers concluded that the system successfully achieved its main objective of creating a functional desktop-based point-of-sale and inventory management system for small grocery store operations.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue VI, June 2026

The system effectively addressed common problems in manual sales and inventory management, such as inaccurate computations, slow transaction processing, missing records, difficulty in monitoring product stocks, and delayed report generation. Through POS², users can process sales, manage products, monitor inventory, handle refunds, generate reports, and record activities in a faster and more organized manner.

The integration of SQLite provided reliable local database storage for important records, including administrator accounts, inventory items, sales history, and activity logs. This allowed the system to keep data saved even after the program was closed. As a result, the system became more dependable compared to manual notebooks or temporary data storage.

The system also improved transaction accuracy through automated sales computation and stock updating. When a sale is completed, the system automatically computes the total amount, records the transaction, and deducts the sold quantity from the inventory. When a refund is processed, the system records the refund and restores the returned stock. These functions help maintain accurate sales and inventory records.

The use of PIN authentication also helped protect important system functions from unauthorized access. This feature adds a basic level of security, especially for inventory management, reports, and refund processing. In addition, the activity log feature improved accountability by recording important administrative actions performed in the system.

The graphical user interface developed using Tkinter made the system easier to use. Through buttons, input fields, windows, tables, and message boxes, users can operate the system without relying on command-line instructions. This makes POS² suitable for mini grocery store users who need a simple and user-friendly system.

Overall, POS² can be considered a functional, organized, and useful system for basic mini grocery store operations. It demonstrates the practical application of Python programming, GUI development, database integration, sales automation, inventory management, and system testing.

Recommendations

Based on the findings and limitations of the study, the following recommendations are proposed for the improvement of POS²:

Future developers should add barcode scanner integration to make product selection faster and more accurate during checkout.

The system should include receipt printing so that customers can receive printed proof of their transactions.

A user role management feature may be added to separate access levels for administrators, cashiers, and store owners. The system may be improved by adding multi-user access so that different computers or users can access the same system in a store environment.

Future versions should include cloud backup or online database support to prevent data loss and allow access from different devices.

The report module may be enhanced by adding daily, weekly, monthly, and yearly sales summaries with charts or visual analytics.

The system should include low-stock alerts to notify users when a product needs to be restocked.

The refund feature may be improved by adding receipt-based refund verification to avoid incorrect or unauthorized refunds.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue VI, June 2026

The graphical user interface may be improved by making it more modern, responsive, and visually appealing.

Future researchers may conduct usability testing with actual store owners or cashiers to gather feedback and improve the system based on real store operations.

In conclusion, POS² already performs its intended functions effectively, but it can still be enhanced through barcode scanning, receipt printing, stronger user access control, cloud backup, improved reports, low-stock alerts, and better interface design. These improvements would make the system more efficient, secure, and suitable for broader mini grocery store use.

ACKNOWLEDGEMENT

The researchers would like to express their sincere and heartfelt gratitude to Engr. Meshelle N. Fabro, PCpE, for her invaluable guidance, professional expertise, constructive suggestions, and continuous support throughout the completion of this study. Her encouragement and supervision greatly contributed to the successful development and improvement of this research.

The researchers also extend their sincere appreciation to the Eulogio "Amang" Rodriguez Institute of Science and Technology (EARIST), particularly the Department of Computer Engineering and its faculty members, for providing the knowledge, technical skills, academic foundation, and institutional support necessary for the completion of this project. Their dedication to quality education and academic excellence served as an inspiration in accomplishing this study.

Special thanks are given to the families and friends of the researchers for their unwavering support, patience, understanding, encouragement, and motivation throughout the challenges encountered during the research process. Their support served as a source of strength and inspiration in completing this endeavor.

Above all, the researchers express their deepest gratitude to Almighty God for the wisdom, guidance, strength, and perseverance bestowed upon them throughout the conduct of this study. His blessings and grace made this achievement possible.

The researchers sincerely acknowledge everyone who, in one way or another, contributed to the successful completion of this study.

REFERENCES

1. Dimentieva et al. (2026). Web-based POS and inventory management system using RAD and design thinking. Garados et al. (2026). POS impact on operational performance and financial success.
2. Ismail, M. F., Nasir, N., Rahaman, W. E. W. A., Rashid, A. A., & Rusli, M. H. M. (2024). InventsimQD: GUI-based teaching application for quantity discounts inventory models. MYJIEAS, 4(01), 332–339.
3. Madamidola, A., Daramola, O. A., Akintola, K. G., & Adeboje, O. T. (2024). A review of existing inventory management systems. IJRES, 12(9), 40–50.
4. Python Software Foundation. (n.d.). Python documentation.
5. Python Software Foundation. (n.d.). Tkinter — Python interface to Tcl/Tk. SQLite. (n.d.). SQLite documentation.
6. Suya et al. (2026). Smart inventory management for mini grocery stores.
7. Trappey, C. V., Trappey, A. J. C., & Hwang, S.-J. (1996). A computerized quality function deployment approach for retail services. Computers & Industrial Engineering, 30(4), 611–622.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue VI, June 2026

ABOUT THE AUTHORS

Marky R. Balboa is a Computer Engineering student at Eulogio "Amang" Rodriguez Institute of Science and Technology. He enjoys learning about computers, programming, and modern technology. He is dedicated to his studies and works hard to improve his technical and problem-solving skills. He hopes to build a successful career in the technology industry in the future.

Air Humphrey B. Banol is currently pursuing a degree in Computer Engineering at Eulogio "Amang" Rodriguez Institute of Science and Technology. He is interested in learning new technologies and expanding his understanding of computer systems. He is known for being determined, responsible, and willing to learn. Through continuous effort and dedication, he aims to become a skilled professional in the field of computer engineering.

Freddie G. Devera II is taking up Bachelor of Science in Computer Engineering at Eulogio "Amang" Rodriguez Institute of Science and Technology. He is passionate about technology and is committed to improving his knowledge and skills in the field. He is a responsible and hardworking student who always strives to do his best in both academic and personal activities. His goal is to become a successful computer engineer and contribute to the advancement of technology.

Ruzzel M. Rafal is a Computer Engineering student who is passionate about technology, programming, and mathematics. He continuously works to improve his knowledge and skills in software development, problem-solving, and analytical thinking. Ruzzel is dedicated, hardworking, and eager to learn new concepts that help him grow both academically and professionally. His strong interest in programming and mathematics allows him to develop logical solutions to complex problems and strengthen his technical abilities. Through continuous learning and determination, he aspires to become a successful computer engineer and contribute to the advancement of technology.

Kier Franz D. Tablizo is a Computer Engineering student who is focused on developing his knowledge in programming, computer hardware, and system design. He is a cooperative and dependable individual who enjoys working with others on academic projects and activities. He believes that continuous learning and hard work are important for success. In the future, he hopes to use his skills to create useful technological solutions.

Angelo D. Tenedero is studying Computer Engineering at Eulogio "Amang" Rodriguez Institute of Science and Technology. He is eager to learn and improve his abilities in technology and engineering. He is a dedicated student who values discipline, teamwork, and self-improvement. His ambition is to become a competent computer engineer and make a positive impact in the field of technology.

Engr. Meshelle N. Fabro is a licensed Professional Computer Engineer and educator at the Eulogio "Amang" Rodriguez Institute of Science and Technology (EARIST). She is currently pursuing a Master of Science in Computer Engineering at Bulacan State University and has professional industry experience as a Technical Engineer at Hewlett-Packard (HP) and an IT Engineer at IBM.She is also a TESDA CSS NC II and Trainers Methodology Level I (TM1) holder, demonstrating her commitment to technical excellence and competency-based education. Passionate about teaching and innovation, she is dedicated to guiding Computer Engineering students in becoming industry-ready, competent, and highly skilled individuals equipped for the evolving demands of the technology industry.