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

Smartpos Budgeteer: Design and Implementation of a Python-Based Sales, Inventory, and Expense Management System Using Oop Principles

Barcena, Jules., Braw, Lhemuel Jr M., Casta, Pamela Ann., Garin, Ma. Soccoro B., Amerie R. , Rosento, Hannah L., Valentin, Patrick Jake D., San Andres, Lorraine, Reyes, Bizzy

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

DOI: https://doi.org/10.51583/IJLTEMAS.2026.150600086

Received: 14 June 2026; Accepted: 19 June 2026; Published: 07 July 2026

ABSTRACT

Manual management of sales, inventory, and expenses in small retail establishments often results in inaccurate calculations, delayed transactions, inconsistent stock tracking, missing records, and challenges in report generation. These issues can negatively affect business operations, financial monitoring, and decision-making processes. To address these challenges, this study developed SmartPOS Budgeteer, a Python-based, Graphical User Interface (GUI)-driven system that integrates Point-of-Sale (POS), inventory management, and expense tracking functionalities into a single platform. The system was designed and implemented using Object-Oriented Programming (OOP) principles to ensure modularity, maintainability, scalability, and efficient code organization.

The graphical user interface was developed using **Tkinter**, providing an intuitive and user-friendly environment for business owners and staff. Data management and storage were implemented through **SQLite** and **MySQL** databases to ensure secure, accurate, and permanent record keeping. The system includes modules for sales processing, inventory monitoring, expense recording, refund management, activity logging, and report generation. Automated calculations and real-time database updates help minimize human errors while improving operational efficiency.

A system development approach based on the Software Development Life Cycle (SDLC) was utilized throughout the planning, design, implementation, testing, and deployment phases. Functional testing was conducted to evaluate the performance, reliability, and accuracy of each module. Results showed that the system successfully processed sales transactions, updated inventory records, recorded expenses, handled refund transactions, validated user inputs, and generated accurate reports. All test cases passed, demonstrating the reliability and effectiveness of the system in supporting retail business operations.

The findings indicate that SmartPOS Budgeteer provides a unified, efficient, and user-friendly solution for small retail establishments. By integrating POS, inventory, and expense management functionalities, the system improves accuracy, accountability, record organization, and business decision-making. Furthermore, the implementation of OOP principles and database integration contributes to system maintainability, data integrity, and long-term usability. The study demonstrates the potential of Python-based business management systems in enhancing operational productivity and supporting the digital transformation of small retail enterprises.

Keywords: SmartPOS Budgeteer, Point-of-Sale, Inventory Management, Expense Tracking, Python, Tkinter, ObjectOriented Programming, SQLite, MySQL, Retail Management System.

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

INTRODUCTION

Background of the Study

The rapid advancement of information technology has significantly transformed the way businesses manage their daily operations. In the retail industry, efficient management of sales transactions, inventory records, and business expenses is essential to maintaining profitability, operational effectiveness, and customer satisfaction. Small retail establishments, however, often rely on traditional methods such as handwritten records, notebooks, and spreadsheetbased systems to manage their operations. While these methods may be sufficient during the early stages of business development, they become increasingly inefficient as transaction volumes grow. Manual record-keeping is susceptible to human errors, duplication of records, inaccurate calculations, misplaced documents, and delays in retrieving important information, which can negatively affect business performance and decision-making.

One of the most common challenges faced by small retail businesses is the difficulty of maintaining accurate and upto-date inventory records. Inventory discrepancies often occur due to delayed stock updates, recording mistakes, and the absence of real-time monitoring mechanisms. These issues may result in overstocking, stock shortages, missed sales opportunities, and increased operational costs. In addition, manual sales recording can lead to incorrect transaction computations, inconsistencies in financial reports, and difficulties in monitoring business performance. Likewise, inadequate expense tracking may result in poor budget management and reduced visibility of business expenditures, making it difficult for owners to evaluate profitability and control operational costs.

To address these challenges, many businesses have adopted Point-of-Sale (POS) systems that automate sales transactions and inventory updates. A POS system allows businesses to process customer purchases efficiently while maintaining accurate records of products sold and inventory levels. Similarly, expense management systems help businesses organize and monitor operational expenses by categorizing expenditures and generating financial summaries. Despite the benefits of these systems, many small businesses still use separate applications or manual processes for sales, inventory, and expense management, resulting in fragmented information and inefficient workflows. Therefore, there is a growing need for an integrated solution that combines these essential business functions into a single, centralized platform.

The integration of sales, inventory, and expense management within one system provides numerous advantages, including improved accuracy, faster transaction processing, better record organization, and enhanced reporting capabilities. By consolidating business data into a unified database, business owners can easily monitor operations, track financial performance, and make informed decisions based on real-time information. Furthermore, automation minimizes repetitive tasks, reduces human intervention, and improves overall operational efficiency.

To respond to this need, the researchers developed **SmartPOS Budgeteer**, a Python-based Sales, Inventory, and

Expense Management System designed specifically for small retail businesses. The system utilizes **Object-Oriented Programming (OOP)** principles to create a modular, maintainable, and scalable application architecture. Through the use of Python programming, Tkinter-based graphical user interfaces, and SQLite/MySQL database integration, the system provides a comprehensive solution for processing sales transactions, monitoring inventory levels, recording business expenses, handling product refunds, and generating reports. The implementation of a user-friendly graphical interface allows users with minimal technical knowledge to navigate the system efficiently and perform business operations with ease.

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

SmartPOS Budgeteer aims to improve the accuracy and reliability of retail management processes by automating essential business functions and maintaining organized digital records. Through the integration of sales, inventory, and expense tracking features, the system seeks to eliminate the limitations associated with manual record-keeping while providing business owners with valuable insights for monitoring performance and supporting strategic decisionmaking. As digital transformation continues to influence business operations, systems such as SmartPOS Budgeteer demonstrate the importance of adopting technology-driven solutions to enhance productivity, accountability, and longterm business sustainability.

Problem Statement

Despite the availability of various digital business tools, many small retail establishments continue to experience difficulties in managing sales transactions, inventory records, and operational expenses efficiently. Manual methods often lead to inaccurate computations, delayed updates, inconsistent record keeping, and challenges in generating reliable reports. These issues can affect business productivity, financial management, and customer service. To address these concerns, this study seeks to answer the following questions:

1. How can sales, inventory, and expenses be efficiently tracked and managed through a unified system for small retail operations?

2. How can the system automate calculations, inventory updates, refund processing, and expense recording to improve operational efficiency?

3. How can a graphical user interface enhance usability and accessibility for business owners and employees?

4. How can Object-Oriented Programming principles contribute to the reliability, maintainability, and scalability of the system?

5. How can database integration ensure secure, accurate, and permanent storage of business records?

6. How effective is the developed system in improving retail management operations based on functional testing results?

Objectives of the Study

General Objective

The primary objective of this study is to design, develop, and implement **SmartPOS Budgeteer**, a Python-based, Object-Oriented Programming-driven Sales, Inventory, and Expense Management System that provides an integrated solution for small retail business operations.

Specific Objectives

Specifically, the study aims to:

1. Design and implement a sales management module capable of recording and processing customer transactions accurately.

2. Develop an inventory management module for monitoring stock levels and managing product information.

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

3. Create an expense management module that records, categorizes, and summarizes operational expenditures.
4. Implement a refund management feature that accurately updates inventory records and transaction histories.
5. Integrate SQLite and MySQL databases for secure, reliable, and permanent data storage.
6. Design a user-friendly graphical user interface using Tkinter for efficient navigation and operation.
7. Generate comprehensive reports for sales, inventory, expenses, and overall business performance.
8. Apply Object-Oriented Programming principles to ensure modularity, maintainability, and scalability of the system.
9. Conduct functional testing to evaluate the accuracy, reliability, usability, and performance of the developed system.

Significance of the Study

The study is expected to provide benefits to various stakeholders.

For **Business Owners**, the system offers an efficient tool for monitoring sales, inventory, and expenses, enabling informed decision-making and improved operational control. For **Employees**, the automated features simplify routine tasks, reduce manual workloads, and minimize errors during transactions and inventory management. For **Future Researchers and Developers**, the study serves as a reference for developing integrated business management systems using Python, Object-Oriented Programming, and database technologies. For **Academic Institutions**, the project demonstrates the practical application of software engineering concepts in solving real-world business problems. Finally, for **Small Retail Businesses**, the study highlights the advantages of adopting computerized systems, including improved efficiency, better record management, enhanced reporting, and increased productivity.

Scope and Limitations

This study focuses on the development and implementation of a desktop-based Sales, Inventory, and Expense

Management System intended for small retail businesses. The system includes modules for sales processing, inventory monitoring, expense recording, refund management, activity logging, database integration, and report generation. It provides automated transaction processing, inventory updates, expense tracking, and report creation through a graphical user interface developed using Tkinter.

The system is designed for local deployment and utilizes SQLite and MySQL databases for data storage and management. The study is limited to a single-user desktop environment and does not include cloud-based storage, online payment processing, mobile application support, barcode scanning, multi-user networking, or web-based access. Furthermore, advanced analytics, artificial intelligence features, and third-party business integrations are beyond the scope of the current implementation. These enhancements may be considered for future development to further improve the functionality, accessibility, and scalability of the system.

REVIEW OF RELEVANT THEORY, STUDIES, AND LITERATURE

This chapter presents the theories, related studies, and literature that serve as the foundation for the development of

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

SmartPOS Budgeteer, a Python-based Sales, Inventory, and Expense Management System developed using ObjectOriented Programming (OOP) principles. The review examines concepts related to Point-of-Sale systems, inventory management, expense tracking, database management systems, Python programming, Object-Oriented Programming, graphical user interface design, and previous studies relevant to integrated retail management solutions. These concepts provide theoretical and practical support for the development and implementation of the proposed system.

Point-of-Sale Systems

A Point-of-Sale (POS) system is a computerized platform designed to automate sales transactions, process customer purchases, calculate totals, maintain inventory records, and generate business reports. Modern POS systems have become essential tools in retail businesses because they improve transaction accuracy, reduce processing time, and provide real-time access to sales information. According to Garados et al. (2026), POS systems significantly enhance operational efficiency by minimizing human intervention and automating repetitive business processes.

Traditional sales recording methods often rely on handwritten receipts and manual computations, which are susceptible to errors and inconsistencies. POS systems address these challenges by automatically calculating transaction totals, recording sales data, and updating inventory levels immediately after each transaction. Additionally, these systems generate detailed reports that enable business owners to monitor performance, analyze sales trends, and make informed decisions. The integration of sales processing and inventory management allows businesses to maintain accurate stock records while improving customer service and operational productivity.

Research findings indicate that POS systems contribute to increased accuracy in transaction processing, improved inventory monitoring, and enhanced reporting capabilities. As businesses continue to adopt digital solutions, POS technology has become a fundamental component of efficient retail operations.

Inventory Management Systems

Inventory management refers to the process of monitoring, controlling, and maintaining product stocks to ensure the availability of goods while minimizing operational costs. Effective inventory management enables businesses to balance supply and demand, prevent product shortages, and reduce excess inventory. According to Madamidola et al. (2024), inventory management systems play a critical role in ensuring business continuity and operational efficiency.

Manual inventory tracking methods often result in inaccurate stock records, delayed updates, misplaced information, and difficulties in monitoring product movement. These problems can lead to stockouts, overstocking, financial losses, and reduced customer satisfaction. Computerized inventory management systems address these challenges by providing real-time stock monitoring, automated inventory updates, and efficient record management.

Modern inventory systems offer functionalities such as stock level monitoring, product categorization, low-stock alerts, inventory history tracking, and report generation. These features help organizations maintain optimal inventory levels and improve decision-making processes. Integrating inventory management with sales processing further enhances operational efficiency because stock quantities are automatically adjusted whenever sales or returns occur.

Expense Management Systems

Expense management systems are designed to record, monitor, categorize, and analyze business expenditures. Financial management is an essential component of business operations because expenses directly influence

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

profitability and organizational sustainability. Small businesses often encounter difficulties in managing expenses manually due to missing records, inaccurate calculations, and inefficient budgeting practices.

According to Bhavani et al. (2025), automated expense management systems improve financial transparency by organizing expenditure data and generating accurate summaries of operational costs. These systems enable businesses to classify expenses into categories such as utilities, supplies, transportation, maintenance, and operational costs. Through automation, businesses can monitor spending patterns, identify unnecessary expenditures, and improve financial planning.

Expense management systems also provide reporting tools that generate daily, weekly, monthly, and annual expense summaries. These reports assist business owners in evaluating financial performance, managing budgets, and making informed business decisions. The integration of expense tracking with sales and inventory management creates a comprehensive business management environment that promotes financial accountability and operational efficiency.

Database Management Systems

A database management system (DBMS) serves as the backbone of modern information systems by providing organized storage, retrieval, and management of data. Database systems are particularly important in retail environments where large volumes of sales transactions, inventory records, and expense data must be stored securely and retrieved efficiently. According to Silberschatz, Korth, and Sudarshan (2019), relational databases provide mechanisms for ensuring data integrity, consistency, security, and reliability.

SmartPOS Budgeteer utilizes both SQLite and MySQL database technologies. SQLite is a lightweight, serverless database engine suitable for local storage and desktop applications, while MySQL provides a robust relational database environment capable of handling large datasets and complex queries. Together, these technologies support data persistence, efficient searching, report generation, and CRUD (Create, Read, Update, Delete) operations.

Database integration provides several benefits, including the prevention of duplicate records, improved data consistency, secure storage, and efficient retrieval of information. Furthermore, databases enable the generation of accurate reports that support managerial decision-making. The implementation of a reliable database system ensures that business information remains organized, accessible, and protected from data loss.

Python Programming and Object-Oriented Programming

Python has become one of the most widely used programming languages due to its simplicity, readability, flexibility, and extensive collection of libraries. It is commonly used for desktop applications, web development, automation, artificial intelligence, and data analysis. Python's ease of use makes it particularly suitable for educational and business applications.

Object-Oriented Programming (OOP) is a programming paradigm that organizes software design around objects, classes, and reusable code components. OOP promotes modularity, scalability, maintainability, and code reusability. According to Birajdar (2025), OOP improves software quality by simplifying system development, debugging, and future enhancements.

In SmartPOS Budgeteer, OOP principles are applied through classes representing business entities such as products, sales transactions, inventory records, expenses, and reports. Methods within these classes perform operations such as processing sales, updating stock quantities, generating reports, and recording expenses. The use of encapsulation, inheritance, abstraction, and polymorphism enhances the system's maintainability and flexibility while reducing software complexity.

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

The combination of Python and OOP enables the development of a structured and scalable application capable of supporting future enhancements and additional functionalities.

Graphical User Interface (GUI) Design

A Graphical User Interface (GUI) allows users to interact with software through visual elements such as buttons, menus, forms, icons, and tables instead of command-line instructions. GUI design plays a significant role in system usability because it directly affects user experience, efficiency, and satisfaction.

SmartPOS Budgeteer utilizes the Tkinter library to develop an interactive desktop interface that supports inventory management, sales processing, expense recording, refund handling, and report generation. According to Shneiderman et al. (2016), well-designed graphical interfaces improve productivity by reducing user errors, simplifying navigation, and enhancing accessibility.

The GUI components of SmartPOS Budgeteer include data entry forms, transaction tables, inventory displays, dashboards, summary panels, and report-generation tools. These features allow users to perform tasks efficiently even without advanced technical knowledge. A user-friendly interface is particularly important for small retail business owners and employees who may have limited experience with computerized systems.

Related Studies

Several studies support the development of integrated business management systems similar to SmartPOS Budgeteer.

Dimentieva et al. (2026) developed a web-based Point-of-Sale system integrated with inventory management using the Rapid Application Development methodology. Their findings indicated improvements in transaction speed, inventory accuracy, and user satisfaction.

Suya et al. (2026) implemented a smart grocery POS system designed to automate retail transactions and inventory

monitoring. The study reported significant reductions in stock discrepancies and faster report generation, demonstrating the effectiveness of integrated retail management systems.

Madamidola et al. (2024) developed an inventory management solution combined with financial tracking functionalities for small businesses. Their findings emphasized the importance of integrating inventory monitoring with financial management to improve operational efficiency and business control.

Bhavani et al. (2025) created a GUI-based expense tracking system using Python. The study demonstrated that automated expense management improves financial record accuracy, simplifies budgeting processes, and enhances decision-making capabilities.

Furthermore, Trappey, Trappey, and Hwang (1996) emphasized the role of computerized retail systems in improving business performance through enhanced data organization, operational control, and strategic decision-making. Their study highlighted the importance of integrating multiple business functions within a single information system.

Collectively, these studies demonstrate that integrating sales, inventory, and expense management into one platform improves efficiency, accuracy, usability, and overall business performance.

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

Synthesis

The reviewed theories, studies, and literature reveal that manual business management methods are often inefficient, time-consuming, and prone to human error. Point-of-Sale systems improve transaction processing and inventory tracking, while inventory management systems ensure accurate stock monitoring and control. Expense management systems provide effective financial tracking and reporting capabilities, contributing to better budget management and decision-making.

The literature further highlights the importance of database integration in maintaining secure, organized, and reliable records. Likewise, Python programming and Object-Oriented Programming provide a strong foundation for developing scalable, maintainable, and efficient software applications. Graphical User Interfaces enhance system usability by allowing users to interact with software through intuitive visual elements.

Based on these findings, SmartPOS Budgeteer was designed and developed as an integrated solution that combines sales management, inventory monitoring, and expense tracking within a single Python-based OOP application. By incorporating database technologies and a user-friendly GUI, the system addresses common operational challenges faced by small retail businesses while improving efficiency, accuracy, accountability, and decision-making capabilities.

METHODOLOGY

This chapter presents the research methodology employed in the design, development, implementation, and evaluation of **SmartPOS Budgeteer**, a Python-based Sales, Inventory, and Expense Management System developed using Object-Oriented Programming (OOP) principles. It discusses the research design, system development process, system architecture, database integration, graphical user interface development, testing procedures, implementation strategies, and technologies utilized throughout the study. The methodology was designed to ensure that the developed system effectively addresses the operational challenges commonly experienced by small retail businesses while providing a reliable, user-friendly, and efficient management solution.

Research Design

The study adopted a **System Development Research Design**, which focuses on the creation, testing, and evaluation of a software application intended to solve specific operational problems. The development process followed the **Software Development Life Cycle (SDLC)** framework, which consists of several interconnected phases, namely planning, analysis, system design, development, testing, implementation, and maintenance. This structured approach ensured that the requirements of the intended users were properly identified, translated into system specifications, and implemented into a functional application.

The development of SmartPOS Budgeteer was motivated by common challenges encountered by small retail businesses, including inaccurate sales calculations, inventory discrepancies, inefficient expense tracking, delayed report generation, and difficulties in maintaining organized business records. Through the application of modern software engineering practices and database technologies, the study aimed to provide an integrated solution that automates business operations and improves management efficiency.

The research design also incorporated iterative evaluation and testing procedures throughout the development process to ensure that each system component functioned correctly and met the intended objectives. This approach enabled the researchers to identify and correct system issues during development, resulting in a more reliable and stable application.

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

System Development Process

The development of SmartPOS Budgeteer followed the phases of the Software Development Life Cycle (SDLC) to ensure a systematic and organized implementation process. Each phase contributed to the successful development of the system.

Planning Phase

The planning phase involved identifying the operational challenges faced by small retail businesses and determining the requirements necessary for the proposed system. Information was gathered through observation of retail management practices and analysis of existing manual processes.

During this phase, the researchers identified several key requirements, including sales transaction processing, inventory monitoring, expense tracking, refund management, report generation, activity logging, and user authentication.

The planning stage also established the project scope, system objectives, development schedule, and resource requirements. Functional and non-functional requirements were documented to guide the design and implementation phases.

Analysis Phase

The analysis phase focused on understanding the flow of information and transactions within retail business operations. Existing processes for recording sales, monitoring inventory, and tracking expenses were examined to identify inefficiencies and opportunities for automation.

System requirements were translated into functional specifications that defined the expected behavior of each module.

The researchers analyzed user needs and identified critical system functionalities such as automated calculations, realtime inventory updates, report generation, secure data storage, and intuitive navigation. This phase provided the foundation for designing the overall system architecture.

System Design Phase

The design phase involved creating the blueprint of the system. Several design tools and models were developed to represent the structure, behavior, and interaction of system components.

System design activities included the creation of:

1. Flowcharts representing the workflow of each module and transaction process
2. Entity Relationship Diagrams (ERD) illustrating database structures and relationships
3. Class diagrams demonstrating the Object-Oriented Programming architecture
4. User interface mockups developed using Figma
5. Navigation structures showing module interactions and workflow transitions

The system was divided into several major modules:

Sales Module – Handles customer purchases, calculates totals, records transactions, and updates inventory automatically.

Inventory Module – Allows users to add, edit, delete, search, and monitor product records and stock levels.

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

Expense Module – Records operational expenses, categorizes expenditures, and generates expense summaries.

Refund Module – Processes returned products, updates stock quantities, and adjusts sales records accordingly.

Reports Module – Generates daily, weekly, monthly, and overall summaries of sales, inventory, expenses, and profit.

Activity Log Module – Records user actions and system activities for accountability and auditing purposes.

User Authentication Module – Restricts system access through secure login credentials.

Exit and Logout Module – Ensures safe termination of sessions while preserving database integrity.

Development Phase

The development phase involved coding and implementing the system using Python programming language. The application was developed following Object-Oriented Programming principles to ensure modularity, scalability, maintainability, and code reusability.

Several classes were created to represent core business entities, including:

- Product Class
- Sales Class
- Expense Class
- Refund Class
- User Class
- Report Class
- Activity Log Class

Methods were implemented to perform operations such as adding products, updating inventory, processing transactions, recording expenses, generating reports, validating inputs, and managing user activities.

The system interface was developed using the Tkinter framework, providing users with an interactive environment for performing business operations efficiently.

System Flow

Its illustrates the overall workflow of SmartPOS Budgeteer. The process begins with user authentication through a secure login interface. Upon successful login, users are granted access to the main dashboard where they can navigate to various modules.

Transactions performed within the Sales, Inventory, Expense, and Refund modules automatically update the database. Generated reports retrieve data directly from stored records to provide accurate summaries and business insights. The flowchart demonstrates the interaction between system modules and highlights the integrated structure of the application.

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

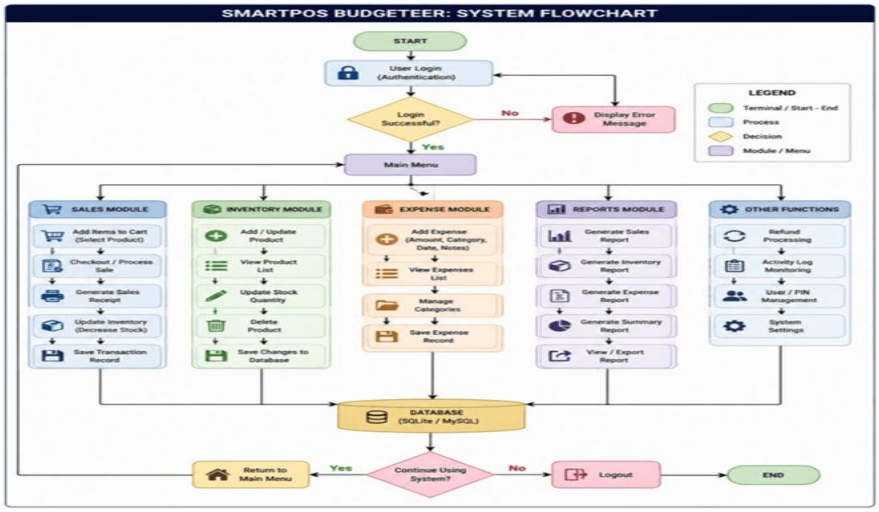

Figure 1. Flowchart of the SmartPOS Budgeteer System

The figure presents the complete workflow of the system from login authentication to transaction processing, inventory monitoring, expense management, report generation, and database updates. It illustrates the decision-making processes and module interactions implemented within the Object-Oriented Programming architecture.

Database Design and Integration

Database integration serves as the foundation for storing, retrieving, and managing business information within SmartPOS Budgeteer. The system utilizes both SQLite and MySQL database technologies to provide efficient data management and persistence.

SQLite functions as a lightweight local database that supports rapid data retrieval and offline functionality. MySQL serves as the primary relational database management system responsible for storing permanent records and supporting complex data operations.

The database consists of several interconnected tables:

Table Name	**Purpose**
products	Stores product information, pricing, stock levels, and categories
sales_history	Records sales transactions and transaction details
expenses	Stores expense categories, descriptions, and amounts
refunds	Maintains records of returned items and refund transactions
users	Stores authentication credentials and user information
activity_log	Tracks administrative actions and user activities
sqlite_sequence	Maintains auto-increment values for database records

The database design supports CRUD (Create, Read, Update, Delete) operations while maintaining data integrity, consistency, and security.

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

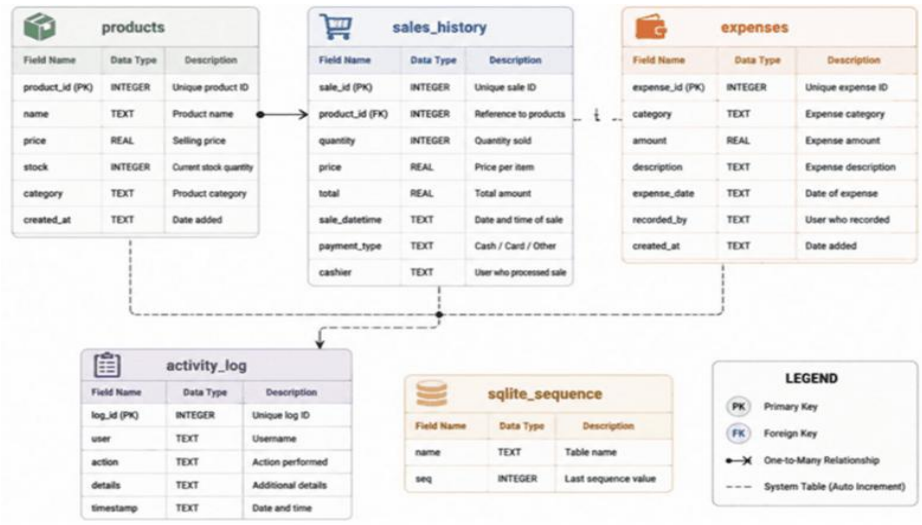

Figure 2. Database Integration Diagram of SmartPOS Budgeteer

The figure illustrates the relationships among database tables and demonstrates how information flows between sales, inventory, expenses, refunds, and activity logging modules. It highlights the interaction between SQLite and MySQL databases and shows how data is stored, retrieved, and updated throughout system operations.

Graphical User Interface Development

The Graphical User Interface (GUI) was developed using Tkinter to provide a user-friendly and visually interactive environment. The GUI was designed according to usability principles to ensure that users can navigate the system efficiently without requiring advanced technical skills.

Key GUI components include:

- Product entry and management forms
- Sales transaction windows
- Expense recording interfaces
- Refund processing forms
- Inventory monitoring tables
- Report generation panels
- Dashboard summary widgets
- Navigation menus and control buttons

The dashboard displays important business information such as total sales, current inventory levels, total expenses, low-stock alerts, and recent activity logs.

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

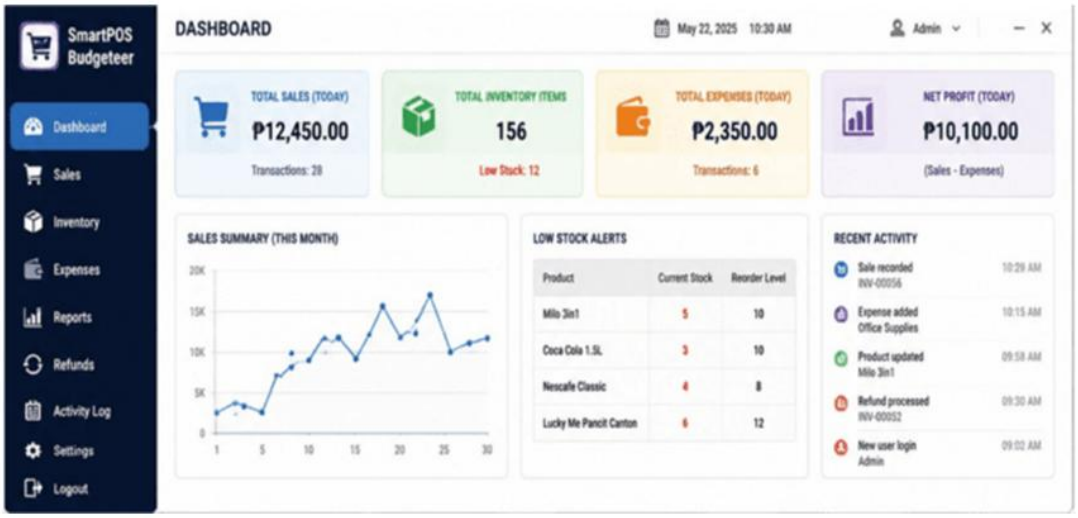

Figure 3. Graphical User Interface of SmartPOS Budgeteer

The figure presents the main dashboard and system modules, including Sales, Inventory, Expenses, Refunds, Reports, and Activity Logs. The interface was designed to promote efficient navigation, minimize user errors, and improve operational productivity.

Testing Procedure

To evaluate system performance and functionality, comprehensive functional testing was conducted. Testing procedures focused on validating system operations, ensuring data accuracy, and verifying database connectivity.

1. Test scenarios included:
2. Product addition, modification, and deletion
3. Sales transaction processing
4. Inventory quantity updates
5. Expense recording and categorization
6. Refund transaction processing
7. Report generation
8. User authentication validation
9. Activity logging
10. Database storage and retrieval operations

Input validation testing was also performed to verify that the system correctly handles invalid data, duplicate entries, missing fields, and unauthorized access attempts.

All identified test cases were executed and documented. Results confirmed that system functionalities performed according to expected outcomes and met the project's requirements.

System Implementation

After successful testing and validation, SmartPOS Budgeteer was deployed as a desktop-based application suitable for small retail businesses. Users can perform sales transactions, monitor inventory levels, record expenses, process refunds, generate reports, and manage business operations through a centralized interface.

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

The deployment process included software installation, database initialization, configuration of user credentials, and system orientation for intended users. The implementation demonstrated that the developed application effectively supports daily retail operations while maintaining accurate records and facilitating decision-making.

Tools and Technologies Used

The development of SmartPOS Budgeteer utilized various software tools and technologies to ensure efficiency and reliability.

Programming Language: Python

Programming Paradigm: Object-Oriented Programming (OOP)

GUI Framework: Tkinter

Database Systems: SQLite and MySQL

Development Environment: Visual Studio Code (VS Code)

UI/UX Design Tool: Figma

Version Control: Git (optional for project management)

Testing Method: Functional Testing

These technologies were selected due to their flexibility, reliability, ease of implementation, and suitability for desktop-based business applications.

Summary

The methodology employed a structured Software Development Life Cycle approach combined with Object-Oriented Programming, database integration, and graphical user interface development. Through careful planning, analysis, design, implementation, and testing, SmartPOS Budgeteer was successfully developed as a comprehensive retail management solution. Functional testing confirmed that the system accurately performs sales processing, inventory monitoring, expense tracking, refund management, and report generation while maintaining secure and organized records. The resulting application provides a practical, efficient, and user-friendly tool that supports the operational needs of small retail businesses.

RESULTS AND DISCUSSION

This chapter presents the results and discussion of the developed **SmartPOS Budgeteer** system, highlighting the system overview, main features, graphical interface, functional testing results, and overall system performance.

System Overview

SmartPOS Budgeteer was developed as a unified desktop application for small retail operations. The system integrates **sales processing**, **inventory management**, and **expense tracking**, providing store owners with a single platform to manage business operations efficiently.

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

Figure 4. Graphical User Interface of the SmartPOS Budgeteer system, showing the main dashboard and individual modules for Sales, Inventory, Expenses, Refunds, Reports, and Activity widgets. The interface provides an organized layout with summary metrics, tables, forms, and charts for easy navigation, real-time tracking, and efficient management of retail operations.

The system uses:

Python with **Object-Oriented Programming** for modular, maintainable code

Tkinter GUI for a user-friendly interface

SQLite/MySQL databases for persistent storage of sales, inventory, and expense records

The system starts with **PIN authentication** to secure access. After successful login, the user can navigate to modules for **Sales**, **Inventory**, **Expenses**, **Refunds**, and **Reports**, with all operations updating the database in real-time.

System Features and Results

Sales Module

The Sales module allows staff to record customer purchases, compute totals automatically, and update inventory quantities.

Result:

All valid sales were recorded correctly in the database

Stock was deducted automatically from inventory

Invalid or duplicate entries were blocked through input validation

Inventory Module

This module manages products including adding, editing, and deleting inventory items.

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

Result:

Product records were successfully added, updated, and deleted

Stock quantities were tracked accurately in real-time

Searching and filtering of products returned correct results

Expense Module

The Expense module tracks operational costs, categorizes expenses, and allows for report generation.

Result:

Expenses were correctly saved and categorized

Daily, monthly, and total expense summaries were accurate

Invalid entries were blocked, ensuring correct financial tracking

Refund Module

The Refund module processes returned items, updates stock, and reverses sales transactions if necessary.

Result:

Refunds were accurately recorded in the database

Inventory was restored correctly after a refund

Reports reflected adjusted sales totals

Report Generation

The Reports module generates summaries for sales, expenses, refunds, and net totals.

Result:

Reports matched database entries

Users could generate daily, monthly, and overall summaries

Dashboard totals reflected correct calculations of sales, stock, and expenses

Graphical User Interface

The GUI, implemented using Tkinter, includes:

Forms for sales, expenses, and inventory entry

Tables displaying sales, stock, and expense records

Buttons and menus for module navigation

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

Dashboard with summary metrics

Result:

Users could easily navigate modules and perform operations

Visual interface reduced the need for technical knowledge

Dashboard provided quick insights into sales, stock, and expenses

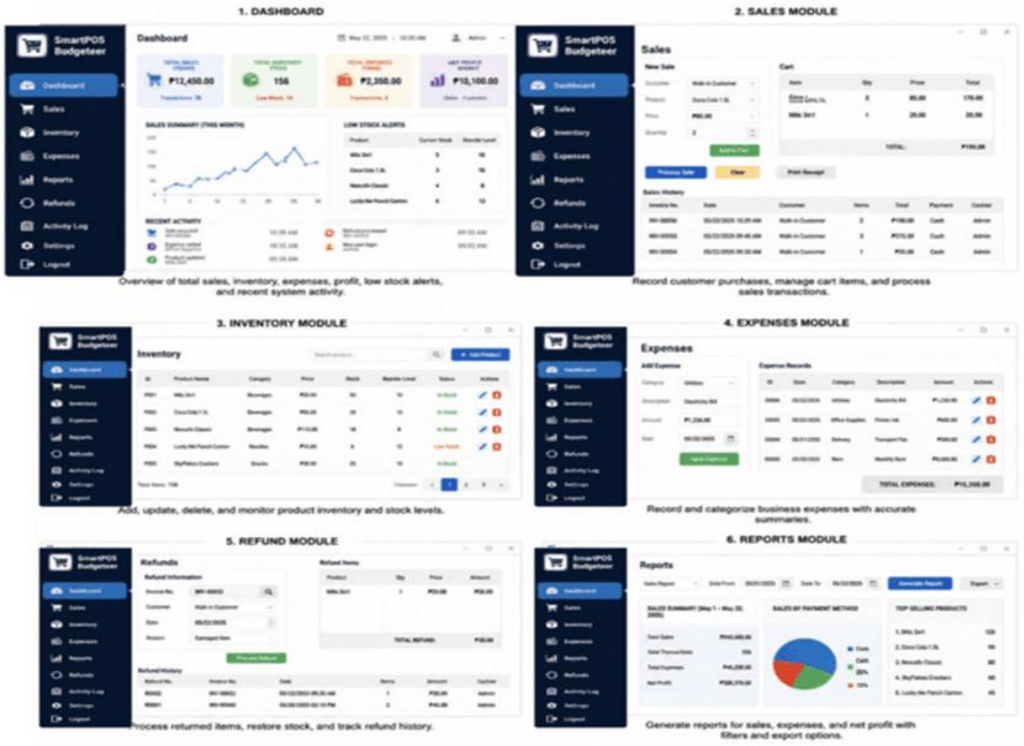

Figure 5. Graphical User Interface of the SmartPOS Budgeteer

System displayed in six separate modules: Dashboard, Sales, Inventory, Expenses, Refunds, and Reports. Each module features intuitive panels, forms, tables, and summary widgets, designed for easy navigation, efficient transaction processing, inventory and expense management, and quick access to reports.

System Testing Results

To evaluate the functionality, reliability, and accuracy of SmartPOS Budgeteer, comprehensive functional testing was conducted on all major system modules. The testing process aimed to verify whether the system performed according to its specified requirements and whether all modules interacted correctly with the integrated database. Each test case was designed to assess the system's ability to process transactions, validate inputs, maintain data integrity, and generate accurate outputs.

The testing covered critical operations including sales processing, inventory management, expense recording, refund handling, report generation, user validation, and database storage. Both normal and exceptional scenarios were tested to ensure that the application could handle valid transactions as well as invalid user inputs effectively.

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

Functional Testing Results

Test Case	Action	Expected Result	Actual Result	Status
Blank Input	Left a required field empty	System blocks saving	Blocked with warning message	PASS
Entered incorrect quantity or Invalid Sales Entry	negative value	Error message displayed	Error displayed successfully	PASS
Duplicate Product ID	Added a product with existing ID	Duplicate blocked System blocks duplicate entry successfully		PASS
Add Expense	Recorded a valid expense	Expense saved in database Successfully saved		PASS
Borrow/Refund	Processed sale return	Stock restored and transaction Updated successfully reversed		PASS
Generate Report	Requested sales and expense summary	Report matches database Correct report generated PASS records		
Inventory Update	Modified product information	Changes saved and reflected Saved successfully PASS in inventory		
Exit System	Closed application	Data remains saved in Data persisted correctly PASS database		

The results indicate that all tested functionalities operated according to their expected behavior. Input validation mechanisms successfully prevented invalid or incomplete data entries, thereby ensuring the accuracy and integrity of stored records. Database operations, including record insertion, updating, retrieval, and deletion, were executed correctly without data loss or corruption.

Additional testing confirmed that inventory quantities were automatically adjusted after sales and refund transactions, ensuring real-time stock monitoring. Expense entries were correctly categorized and included in generated financial summaries. Report outputs were verified against database records and were found to be accurate and consistent.

Overall, all modules passed functional testing, demonstrating that SmartPOS Budgeteer is capable of supporting dayto-day retail operations with reliability, accuracy, and efficiency.

System Performance Evaluation

The developed system exhibited stable performance during testing. Transactions were processed efficiently, and database retrieval operations were completed without significant delays. The graphical user interface responded appropriately to user actions, providing immediate feedback and validation messages when necessary.

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

The integration of Python, Tkinter, SQLite, and MySQL contributed to the overall responsiveness and reliability of the system. No critical system failures, data inconsistencies, or application crashes were encountered during the testing phase. These results indicate that the system is suitable for deployment in small retail environments where accurate transaction processing and record management are essential.

DISCUSSION OF FINDINGS

The results of the study demonstrate that SmartPOS Budgeteer successfully achieved its primary objective of integrating sales management, inventory monitoring, and expense tracking into a single desktop-based application. The system effectively addressed common challenges associated with manual retail operations, including inaccurate calculations, inconsistent record-keeping, delayed reporting, and inefficient inventory monitoring.

One of the most significant findings is the effectiveness of automated sales processing and inventory management. By automatically updating stock levels after each transaction, the system eliminates the need for manual inventory adjustments and reduces the likelihood of stock discrepancies. This feature improves operational efficiency while minimizing human errors that commonly occur in traditional record-keeping systems.

The expense management module also proved effective in organizing and monitoring business expenditures. Through automated expense recording and categorization, business owners can easily track operational costs and generate financial summaries. This capability provides greater financial visibility and supports more informed budgeting and decision-making processes.

Another important finding is the positive impact of the graphical user interface on system usability. The Tkinter-based GUI allows users to perform transactions, manage inventory, record expenses, and generate reports through intuitive forms and controls. As a result, even users with limited technical expertise can operate the system efficiently. The user-friendly design contributes to reduced training requirements and improved overall user satisfaction.

Database integration through SQLite and MySQL played a critical role in ensuring data persistence, consistency, and reliability. All transactions, inventory records, expense entries, and activity logs were stored accurately and remained accessible throughout testing. The use of relational database technologies enhanced record organization and facilitated efficient report generation.

The refund management feature further strengthened the system by maintaining accurate inventory counts and transaction histories. Returned products were successfully restored to inventory, and corresponding sales records were updated appropriately. This functionality helps maintain accurate financial and stock records while supporting customer service operations.

The findings also demonstrate the effectiveness of applying Object-Oriented Programming principles in business application development. The use of classes, methods, and modular structures improved system organization, maintainability, and scalability. This design approach facilitates future enhancements and simplifies system maintenance.

Overall, the results confirm that SmartPOS Budgeteer provides a reliable, accurate, and user-friendly solution for small retail businesses. By automating essential business processes and integrating multiple management functions into a single platform, the system significantly improves operational efficiency, reduces human error, enhances data management, and provides valuable decision-support information through comprehensive reporting features.

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

CONCLUSIONS AND RECOMMENDATIONS

Conclusions

The primary objective of this study was to design, develop, implement, and evaluate **SmartPOS Budgeteer**, a Pythonbased Sales, Inventory, and Expense Management System utilizing Object-Oriented Programming (OOP) principles. Based on the development process, functional testing results, and overall system evaluation, the following conclusions were drawn:

The study successfully developed an integrated retail management system that combines sales processing, inventory monitoring, expense tracking, refund management, activity logging, and report generation into a single application. By consolidating these essential business functions, SmartPOS Budgeteer provides a centralized platform that improves operational efficiency and eliminates the need for multiple disconnected record-keeping systems.

The implementation of automated sales computation, inventory updates, and expense recording significantly reduced the possibility of human errors commonly associated with manual record management. The automation of routine business processes improved transaction accuracy, increased productivity, and minimized the time required to perform daily retail operations. As a result, users can focus more on managing business activities rather than maintaining records manually.

The integration of SQLite and MySQL databases proved effective in ensuring data persistence, consistency, and reliability. The database architecture enabled efficient storage, retrieval, updating, and management of records related to products, transactions, expenses, refunds, and system activities. This capability supports long-term record preservation and enhances the overall reliability of the system.

The use of Object-Oriented Programming principles contributed significantly to the maintainability, scalability, and modularity of the application. Through the implementation of classes, methods, and reusable code structures, the system became easier to manage, debug, and extend for future enhancements. The adoption of OOP principles also improved software organization and promoted efficient system development practices.

The Tkinter-based Graphical User Interface successfully provided an interactive and user-friendly environment for users with varying levels of technical expertise. The interface simplified navigation, reduced operational complexity, and improved user experience through intuitive forms, dashboards, tables, and controls. This finding demonstrates the importance of GUI design in enhancing system usability and user acceptance.

Functional testing confirmed that all modules performed according to their intended functions. Sales transactions were processed accurately, inventory levels were updated correctly, expenses were recorded successfully, refunds were handled appropriately, and reports reflected accurate database records. Furthermore, input validation mechanisms effectively prevented invalid, incomplete, or duplicate data entries, thereby maintaining data integrity and system reliability.

The reporting capabilities of SmartPOS Budgeteer also proved beneficial in supporting managerial decision-making. Through automated report generation and real-time access to business information, users can monitor sales performance, inventory status, operational expenses, and overall business activities more efficiently. These features provide valuable insights that can support strategic planning and operational improvements.

Overall, the findings demonstrate that SmartPOS Budgeteer successfully achieved its objectives and provides a reliable, accurate, efficient, and user-friendly solution for managing small retail business operations. The

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

system addresses common challenges associated with manual record-keeping while improving productivity, accountability, data management, and decision-making capabilities.

Recommendations

While SmartPOS Budgeteer successfully met the objectives of the study and demonstrated satisfactory performance during testing and evaluation, several enhancements may be considered to further improve the system's functionality, usability, scalability, and security.

One recommended enhancement is the implementation of **role-based user access control**. Future versions of the system may include separate user roles such as administrators, managers, cashiers, and inventory personnel. This feature would improve system security by restricting access to sensitive functionalities based on user responsibilities while enhancing accountability through user activity monitoring.

The integration of **barcode scanning technology** is also recommended to streamline product identification and sales processing. Barcode support would enable faster transaction processing, reduce manual data entry, minimize errors, and improve inventory accuracy.

To further improve customer service and transaction documentation, future versions may incorporate a **receipt generation and printing module**. This feature would allow automatic generation of printable receipts containing transaction details, purchased items, payment information, and business branding.

The reporting module may also be enhanced through the inclusion of **advanced analytics and visualization tools**.

Future developments could provide graphical dashboards, charts, trend analyses, and key performance indicators (KPIs) to help business owners gain deeper insights into sales performance, inventory movement, and expense patterns. Additionally, report export functionalities in PDF, Excel, and CSV formats would increase reporting flexibility.

Given the growing importance of data security and accessibility, implementing **cloud-based backup and synchronization features** is highly recommended. Cloud integration would provide automatic backup services, reduce the risk of data loss, and enable remote access to business records from multiple locations.

To support larger business operations, future versions of SmartPOS Budgeteer may include **multi-user and networkbased functionality**. This enhancement would allow multiple users to access and utilize the system simultaneously while maintaining synchronized records across different terminals.

The development of **web-based and mobile application versions** is also recommended to improve accessibility and convenience. Mobile and web platforms would allow business owners to monitor operations, review reports, and manage inventory remotely using smartphones, tablets, or web browsers.

Inventory management can be further strengthened through the implementation of **automated low-stock notifications and reorder alerts**. Such features would help users monitor inventory levels proactively and prevent stock shortages that could negatively impact business operations.

Future researchers are encouraged to conduct **usability testing and user acceptance evaluations** involving actual retail business operators. Gathering feedback from real users can provide valuable insights into system effectiveness, usability, and areas requiring improvement.

Finally, future development efforts may explore the integration of additional business management features such as **online payment processing, customer relationship management (CRM), loyalty programs,**

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

supplier management, accounting integration, tax computation, and artificial intelligence-based sales forecasting. These advanced features would expand the capabilities of the system and support the evolving needs of modern retail businesses.

Final Statement

In conclusion, SmartPOS Budgeteer has demonstrated its effectiveness as a comprehensive retail management solution that integrates sales, inventory, and expense management into a single platform. Through automation, database integration, Object-Oriented Programming, and user-friendly interface design, the system significantly improves operational efficiency and business record management. Although the developed system successfully fulfills its intended objectives, the implementation of the recommended enhancements will further strengthen its functionality, scalability, security, and long-term value for retail businesses and future users.

ACKNOWLEDGEMENT

The researchers would like to express their heartfelt appreciation to **Engr. Meshelle N. Fabro, PCpE**, for her dedicated mentorship, professional guidance, and expert supervision throughout the completion of this study. Her valuable insights, constructive suggestions, and continuous encouragement significantly contributed to the successful development and improvement of this research.

The researchers also extend their sincere gratitude to the **Eulogio "Amang" Rodriguez Institute of Science and Technology (EARIST)**, particularly the **Department of Computer Engineering**, together with its faculty members, for providing the academic knowledge, technical resources, and institutional support necessary for the accomplishment of this project. The institution's commitment to excellence in education, innovation, and research played an essential role in the success of this study.

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

Above all, the researchers offer their deepest gratitude to **Almighty God** for the wisdom, guidance, strength, and perseverance bestowed upon them throughout this endeavor. This work is humbly dedicated to His glory, acknowledging that every achievement was made possible through His grace and blessings.

REFERENCES

1. Bhavani, V. T., Chinthana, S., Aishwarya, K., Shweta, B., & Nirmala, S. (2025). Implementation of an expense tracker using Python. International Journal for Multidisciplinary Research.
2. Birajdar, R. S. (2025). Personal expense tracker: Advancing financial management through software solutions. IIP International Multidisciplinary Research Journal.
3. Bourgeois, D. T., Smith, J. L., Wang, S., & Mortati, J. (2019). Information systems for business and beyond. Saylor Foundation. https://saylordotorg.github.io/text_information-systems-for-business-and-beyond/
4. Dimentieva, A., Li, J., Zhao, G., Li, J., Chen, X., & Wong, D. S. (2026). Designing cloud-based electronic systems with secure data management and application integration.
5. Garados, M., Smith, J., & Wang, S. (2026). POS impact on operational performance and financial success.
6. 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. Malaysian Journal of Intelligent Engineering and Advanced Systems (MYJIEAS), 4(1), 332–339.

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

7. Madamidola, A., Daramola, O. A., Akintola, K. G., & Adeboje, O. T. (2024). A review of existing inventory management systems. International Journal of Research in Engineering and Science (IJRES), 12(9), 40–50.
8. Python Software Foundation. (n.d.). Python documentation. https://docs.python.org/3/
9. Python Software Foundation. (n.d.). Tkinter — Python interface to Tcl/Tk. https://docs.python.org/3/library/tk.html
10. Pressman, R. S., & Maxim, B. R. (2020). Software engineering: A practitioner's approach (9th ed.). McGraw-Hill Education.
11. Silberschatz, A., Korth, H. F., & Sudarshan, S. (2019). Database system concepts (7th ed.). McGraw-Hill Education.
12. Shneiderman, B., Plaisant, C., Cohen, M., Jacobs, S., Elmqvist, N., & Diakopoulos, N. (2016). Designing the user interface: Strategies for effective human-computer interaction (6th ed.). Pearson Education.
13. Suya, F., Li, J., Zhao, G., Li, J., Chen, X., & Wong, D. S. (2026). Smart inventory management for mini grocery stores.
14. Sommerville, I. (2016). Software engineering (10th ed.). Pearson Education.
15. 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. https://doi.org/10.1016/0360-8352(96)00014-8

About The Authors

Jules Barcena is a Bachelor of Science in Computer Engineering student at Eulogio “Amang” Rodriguez Institute of Science and Technology (EARIST). He is a hardworking and determined student who is passionate about learning new technologies and improving his skills in programming and computer systems. Despite the challenges that come with college life, he remains committed to achieving his academic goals and continuously developing his knowledge in the field of technology. He enjoys exploring different areas of computer engineering, learning new concepts, and working on projects that help enhance his technical abilities. Through dedication, perseverance, and continuous learning, he hopes to become a successful computer engineer in the future.

Lhemuel M. Braw Jr. is currently pursuing a degree in Computer Engineering at Eulogio "Amang" Rodriguez Institute of Science and Technology. He is dedicated on learning about technology and continuously works toward achieving his academic and professional goals. Lhemuel is passionate and responsible student who is eager to develop his technical knowledge and explore different fields in engineering. He is committed to overcoming challenges in his studies while improving his analytical thinking, problem-solving, and programming skills. Through continuous learning, determination, and self-improvement, he aspires to become a skilled computer engineer in the future.

Pamela Ann T. Casta is a passionate and goal-oriented student currently studying at Eulogio "Amang" Rodriguez Institute of Science and Technology. She is eager to expand her knowledge and develop valuable skills that will help in achieving future career goals. As a responsible and hardworking individual, she continuously works on improving academic performance, communication skills, and technical abilities. She enjoys participating in school activities, learning new technologies, and collaborating with others to gain more experience and understanding. Despite challenges and difficulties, she remains motivated and determined to succeed. With dedication, perseverance, and confidence, she hopes to make a positive contribution to society and become successful in the chosen profession someday.

Ma. Socorro B. Garin is a Computer Engineering student at the Eulogio “Amang” Rodriguez Institute of Science and Technology. She is passionate about technology and dedicated to improving her knowledge and skills in programming, computer systems, and engineering. She is hardworking, responsible, and determined

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

to achieve her academic goals while continuously learning new things that can help her grow both personally and professionally. Despite challenges, she remains motivated and committed to becoming a successful computer engineer in the future.

Hannah L. Rosento is a first-year Bachelor of Science in Computer Engineering student at the Eulogio "Amang" Rodriguez Institute of Science and Technology (EARIST) Manila Campus. She is known for being hardworking, adaptable, and determined to continuously improve her abilities through learning and experience. She is interested in programming, GUI design, and system development. Through the development of the system, she gained experience in coding, debugging, documentation, and teamwork. This project helped her improve her problem-solving skills and deepen her understanding of object-oriented programming and database management. She hopes to continue learning more about software development while enhancing her technical and analytical skills in the field of computer engineering.

Bizzy Amerie R. Reyes is a sophomore Bachelor of Science in Computer Engineering student at the Eulogio "Amang" Rodriguez Institute of Science and Technology (EARIST). Driven by a strong passion for technology and computer systems, his academic focus centers on software development, artificial intelligence, and hardware design. He aspires to engineer innovative, tech-driven solutions that solve real-world challenges and advance modern technology.

Patrick Jake D. Valentin is currently pursuing his degree in Computer Engineering at the Eulogio "Amang" Rodriguez Institute of Science and Technology. He is deeply committed to his technological education and consistently strives to reach his academic and career objectives. Driven by a strong interest in technology and systemic design, he consistently explores ways to deepen his expertise in hardware-software integration, system analysis, and engineering principles. He places a high value on sharpening his critical thinking and collaborative abilities, understanding that seamless teamwork and clear communication are essential for delivering reliable technical solutions. Through hard work, a commitment to lifelong learning, and practical engineering experience, he aims to establish himself as a skilled, forward-thinking, and resourceful computer engineer in the technology industry.

Lorraine San Andres is a dedicated and enthusiastic student currently studying at **Eulogio 'Amang' Rodriguez Institute of Science and Technology**. She is committed to enhancing her knowledge and developing essential skills that will support her academic and professional growth. As a diligent and responsible individual, she continuously strives to improve her learning, communication abilities, and technical competencies. She actively participates in educational activities, embraces new learning opportunities, and values teamwork and collaboration. Through determination, hard work, and perseverance, she remains focused on achieving her goals and overcoming challenges. Lorraine aspires to become a successful professional who can contribute positively to her community and make a meaningful impact in her chosen field.