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

Dinecart System: An Integrated Python-Based Restaurant Ordering, Billing, and Online Shopping Cart Management System

Evidente, Joric Christian M., Familara, John Ryan M., Ison, Reyden Angel E., Martin, Marl James C., Mahillo, John Maru C., Peñaranda, Sherven Wish J., 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.2026.150600057

Received: 10 June 2026; Accepted: 15 June 2026; Published: 04 July 2026

ABSTRACT

Efficient management of restaurant operations and online orders is essential for enhancing customer experience, reducing human error, and improving business productivity. Traditional manual or semi-digital systems often lead to inaccurate billing, delayed order processing, and inventory mismanagement. This study introduces *DineCart System*, a Python-based application integrating restaurant ordering, billing, and an online shopping cart with an SQLite backend database. Utilizing Object-Oriented Programming (OOP) principles, the system provides a **graphical user interface (GUI)** for seamless order management, real-time inventory updates, accurate billing, and customer-friendly shopping cart functionality. Functional testing demonstrated that the system correctly handles order creation, transaction processing, inventory adjustments, and receipt generation. By consolidating multiple restaurant management tasks into a single platform, *DineCart System* streamlines operations, improves accuracy, and enhances the overall efficiency of restaurant service and online ordering processes.

Keywords: DineCart, Restaurant Ordering System, Online Shopping Cart, Python, GUI, SQLite, Billing System, Object-Oriented Programming

INTRODUCTION

Background of the Study

The rapid growth of the food service industry, coupled with increasing demand for online and in-person ordering, has created a need for integrated systems that efficiently manage restaurant operations and online sales. Traditional manual processes for taking orders, processing payments, and tracking inventory often lead to errors, delays, and inefficient customer service. Incomplete records, inaccurate billing, and stock mismanagement can negatively affect operational efficiency and customer satisfaction.

DineCart System is a Python-based application developed using **Object-Oriented Programming (OOP)** principles and an **SQLite database**. The system integrates three core functionalities: restaurant ordering, billing, and an online shopping cart. It provides a **graphical user interface (GUI)** that allows staff to process orders, update inventory, calculate bills, and generate receipts efficiently, while also enabling customers to manage online shopping cart orders with ease.

By combining multiple management tasks into a single platform, DineCart minimizes manual errors, accelerates transaction processing, and improves overall service quality. The system is particularly useful for small to medium-sized restaurants that seek to streamline both dine-in and online ordering processes while maintaining accurate records.

Statement of the Problem

This study aims to develop an integrated system that addresses the inefficiencies of conventional restaurant

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

management and online shopping cart operations. Specifically, the study seeks to answer the following questions:

1. How can the system automate restaurant ordering and online shopping cart management through a user-friendly GUI?

2. How can an SQLite database ensure secure storage and accurate retrieval of order, inventory, and transaction records?

3. How can the system prevent errors in billing, duplicate entries, or incomplete transactions?

4. How effective is the system in improving operational efficiency, order accuracy, and customer satisfaction for both in-person and online orders?

Objectives of the Study General Objective:

To develop a **GUI-based integrated restaurant ordering, billing, and online shopping cart system** using Python and SQLite that enhances operational efficiency and accuracy.

Specific Objectives:

1. To design the system workflow using flowcharts and GUI mockups.

2. To implement order processing, billing computation, and inventory tracking using Python and OOP principles.

3. To integrate an SQLite database for secure and organized record storage.

4. To implement shopping cart functionality for online order management.

5. To validate user inputs, ensuring accuracy in order quantities, billing, and transaction processing.

6. To test system functionality, usability, and reliability across different modules.

Significance of the Study

The DineCart System provides benefits to several stakeholders:

- **Restaurant Owners and Staff:** Streamlines order management, reduces manual errors, improves inventory tracking, and accelerates billing.
- **Customers:** Ensures faster service, accurate billing, and easy online order management through the shopping cart.
- **Students and Developers:** Serves as a learning tool demonstrating OOP, GUI design, and database integration in real-world applications.
- **Future Researchers:** Provides a foundation for developing more advanced restaurant management systems or integrated POS platforms.

Scope and Limitations Scope

- Developed using **Python** with **Tkinter** for GUI.
- Integrates restaurant ordering, billing, and online shopping cart functionalities.

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

- Uses **SQLite** for secure, organized, and persistent data storage.
- Features include Add Order, View Orders, Update Orders, Delete Orders, Clear Inputs, and generate Receipts.

Limitations:

- No integration with external online payment gateways (e.g., GCash, PayPal, credit cards).
- Designed for **desktop use only**; mobile or web versions are not included.
- Multi-user concurrent access is not supported.
- Advanced analytics or delivery tracking features are not implemented.

REVIEW OF RELEVANT THEORY, STUDIES, AND LITERATURE

This chapter reviews theories, prior studies, and technological frameworks relevant to the development of **DineCart**, a GUI-based Python Restaurant Ordering, Billing, and Online Shopping Cart Management System with SQLite integration. The literature focuses on **restaurant POS systems, online shopping carts, GUI-based management, database integration, and automation in billing**.

GUI-Based Restaurant Management Systems

Graphical User Interface (GUI)-based restaurant management systems provide an interactive platform for staff to manage orders, billing, and inventory. These systems improve efficiency, reduce errors, and enhance usability:

- **Al-Mamrah & Al-Shorman (2021)** highlighted that automated ticketing and registration systems reduce errors, improve operational speed, and ensure secure record-keeping.
- **Skariya (2023)** emphasized that GUI-based systems developed with modern frameworks, such as React JS or Tkinter, provide efficient workflows and real-time feedback for order management.
- **Divina et al. (2025)** demonstrated that POS systems with predictive analytics and database integration enhance decision-making for inventory and billing.

GUI-based applications are essential for minimizing manual errors in order entry, speeding up transactions, and improving overall operational workflow.

Online Shopping Cart Integration

Online shopping cart systems allow customers to select items, review orders, and process payments efficiently:

- **Pandey & Ruia (2025)** highlighted that online cart management ensures accurate record-keeping of customer orders and facilitates tracking of pending and completed transactions.
- **Kulkarni et al. (2025)** emphasized modular and customer-centric designs that allow seamless integration between ordering and payment processes.
- **Ahmed et al. (2025)** implemented machine learning in POS systems to optimize order prediction and stock management for restaurant operations.

Integrating an online shopping cart into a restaurant system improves **accuracy, user convenience, and workflow efficiency**.

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

Database Management and Integration

Secure database systems are crucial for storing, updating, and retrieving records:

- **SQLite** provides a lightweight, serverless solution suitable for desktop-based restaurant systems (Python Software Foundation, 2025).
- **Zainal et al. (2024)** developed a POS system integrating computer vision for automated checkout, demonstrating the importance of structured database design for transaction reliability.
- Relational databases enable **CRUD operations**, maintaining data integrity, and supporting **real-time inventory updates**, billing calculations, and order management.

Database integration ensures that DineCart can handle orders, customer data, inventory, and billing accurately and efficiently.

Automation and Billing Systems

Automated billing and inventory updates reduce human error and increase operational efficiency:

- **Karuppusamy & Geethamani (2025)** emphasized that digital billing systems reduce mistakes in order totals, improve customer satisfaction, and allow instant report generation.
- **Sari et al. (2024)** showed that automated systems simplify menu stack and queue management, allowing staff to focus on service delivery instead of manual tracking.

Automation ensures accurate calculations, real-time updates, and reduces redundant administrative work.

Synthesis

From the reviewed literature:

1. **Manual restaurant operations** are inefficient and prone to errors in ordering, billing, and inventory.
2. **GUI interfaces** improve usability and allow staff to navigate the system easily.
3. **Online shopping carts** enhance customer convenience and accurate order tracking.
4. **Database integration** ensures secure and reliable storage of orders, payments, and inventory.
5. **Automation in billing and inventory** reduces human error and improves operational efficiency.

DineCart integrates these principles, providing a unified Python-based system that automates restaurant ordering, billing, and online shopping cart management. The system combines a user-friendly GUI with reliable SQLite database management to improve operational efficiency, data accuracy, and customer experience.

METHODOLOGY

Research Design

This study employed a **Developmental Research Design** to create and evaluate a **GUI-based restaurant ordering, billing, and online shopping cart system**. This design was appropriate because it focuses on the development, testing, and evaluation of a fully functional software application.

The system was developed using **Python**, with **Tkinter** for the GUI and **SQLite** for the backend database. The study allowed the researchers to systematically analyze user needs, implement features, and assess system

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

effectiveness. Key functionalities include order processing, billing, inventory management, and online shopping cart operations.

System Development Process

The development process followed the **Software Development Life Cycle (SDLC)** framework:

1. **Planning**
   - Identify challenges in traditional restaurant management, such as manual order tracking, billing errors, and inventory mismanagement.
   - Define system objectives, scope, and required features.
2. **Design:**
   - Develop flowcharts to visualize the system's workflow, from order input to database storage.
   - Draft GUI layouts to ensure a user-friendly interface for staff and customers.
   - Design SQLite database schema for orders, menu items, customers, and transactions.
3. **Development**
   - Implement Python OOP classes to represent Order, MenuItem, Transaction, and Inventory.
   - Build GUI components with Tkinter for adding orders, updating items, and viewing records.
   - Integrate SQLite for database management, supporting **CRUD operations** for all records.
4. **Testing**
   - Conduct functional testing to verify order entry, billing, inventory update, and shopping cart accuracy.
   - Validate user input to prevent errors such as duplicate entries or invalid values.
   - Ensure the system handles exceptions and maintains data integrity.
5. **Implementation**
   - Deploy the system in a simulated restaurant environment.
   - Provide training and guidelines to users for effective utilization.
   - Monitor real-time operations to ensure proper database integration and GUI responsiveness.

MaintenanceMonitor system performance post-deployment.

- Apply updates or bug fixes and enhance functionality based on user feedbac

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

System Design

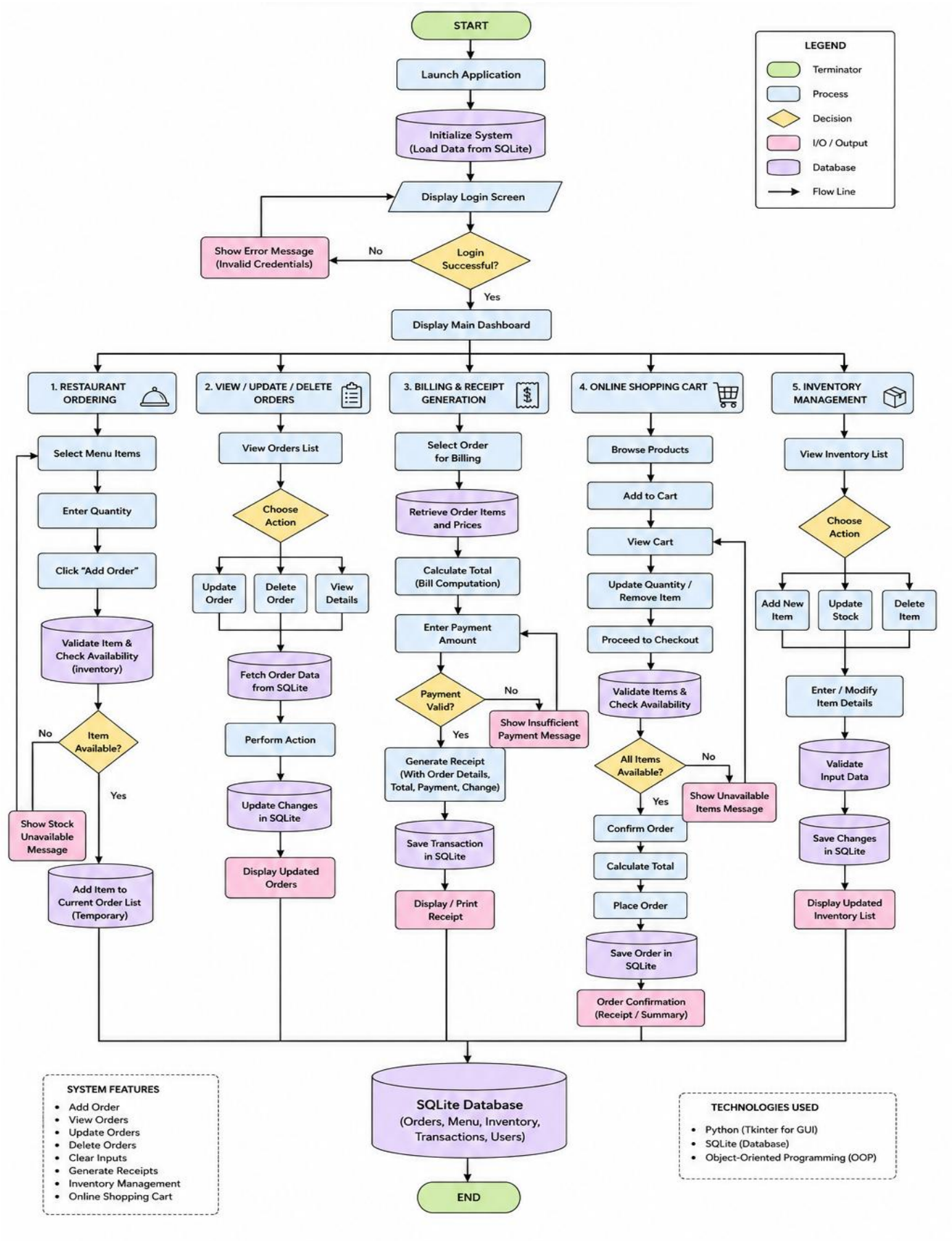

Figure 1. System Flowchart of the DineCart System

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

The figure illustrates the overall workflow and architecture of the DineCart System: An Integrated Python-Based Restaurant Ordering, Billing, and Online Shopping Cart Management System. It presents the sequence of operations from system initialization, user authentication, and dashboard access to the execution of core modules including Restaurant Ordering, Order Management, Billing and Receipt Generation, Online Shopping Cart Management, and Inventory Management. The flowchart highlights decision points, database interactions, input validation processes, and transaction handling procedures, demonstrating how the system automates order processing, billing computation, inventory updates, and online purchasing activities. The integrated SQLite database serves as the central repository for orders, menu items, inventory records, transactions, and user information, ensuring data consistency, accuracy, and efficient restaurant operations.

The system follows a **Model-View-Controller (MVC)** architecture:

1. **Initialization & Database Connection:**
   - On startup, the system connects to the SQLite database and ensures that required tables exist (orders, menu, inventory, transactions).
2. **GUI Layout Mapping:**
   - The main window contains navigation to modules: Dashboard, Add Order, Cart, Billing, Inventory, and Settings.
   - Functional interfaces are loaded dynamically and can be updated without restarting the application.
3. **Event Loop Monitoring:**
   - Captures user actions, such as adding items, modifying orders, or processing payments.
   - Triggers backend database operations for CRUD tasks.
4. **Data Verification & Query Execution:**
   - Validates input fields to prevent empty, duplicate, or incorrect entries.
   - Executes parameterized SQL statements securely for inserting, updating, or deleting records.
5. **Dynamic Data Syncing:**
   - Updates GUI tables and dashboards in real-time.
   - Confirms successful operations with visual alerts and clears input fields after database transactions.

Tools and Technologies Used

Tool/Technology	Purpose
Python	Programming language for implementing business logic and OOP
Tkinter / CustomTkinter	classes. GUI framework for interactive windows, buttons, forms, and tables.
SQLite	Relational database for storing orders, transactions, inventory, and customer
Visual Studio Code / IDLE	IDE for coding, testing, and debugging the
Pillow / Matplotlib	Optional libraries for displaying item images and visual analytics (inventory or sales charts).

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

Summary

The methodology ensures that **DineCart System** is developed systematically from planning, design, coding, testing, and implementation. By combining **Python OOP**, **GUI interfaces**, and **SQLite database management**, the system provides a reliable, efficient, and user-friendly platform for managing restaurant ordering, billing, and online shopping cart operations.

RESULTS AND DISCUSSION

System Oveview

The **DineCart System** is a Python-based application that integrates **restaurant ordering, billing, and online shopping cart management** with an **SQLite database**. The system provides a **GUI interface** for restaurant staff and customers to process orders efficiently, monitor inventory, compute bills automatically, and manage transactions in real-time.

The system modules include:

1. **Main Dashboard:** Provides quick access to all functionalities, including orders, billing, inventory, and sales reports.
2. **Order Management:** Allows staff and customers to select menu items, adjust quantities, and add items to the shopping cart.
3. **Billing & Payment:** Computes totals automatically, applies discounts, and supports multiple payment methods including cash, card, and simulated online payment.
4. **Inventory Management:** Updates stock levels automatically based on orders and highlights low-stock items.
5. **Records & Reporting:** Stores all transaction records in SQLite and provides search, update, and deletion functionalities.

The integration of GUI, OOP, and SQLite ensures **efficiency, accuracy, and security** in all operations.

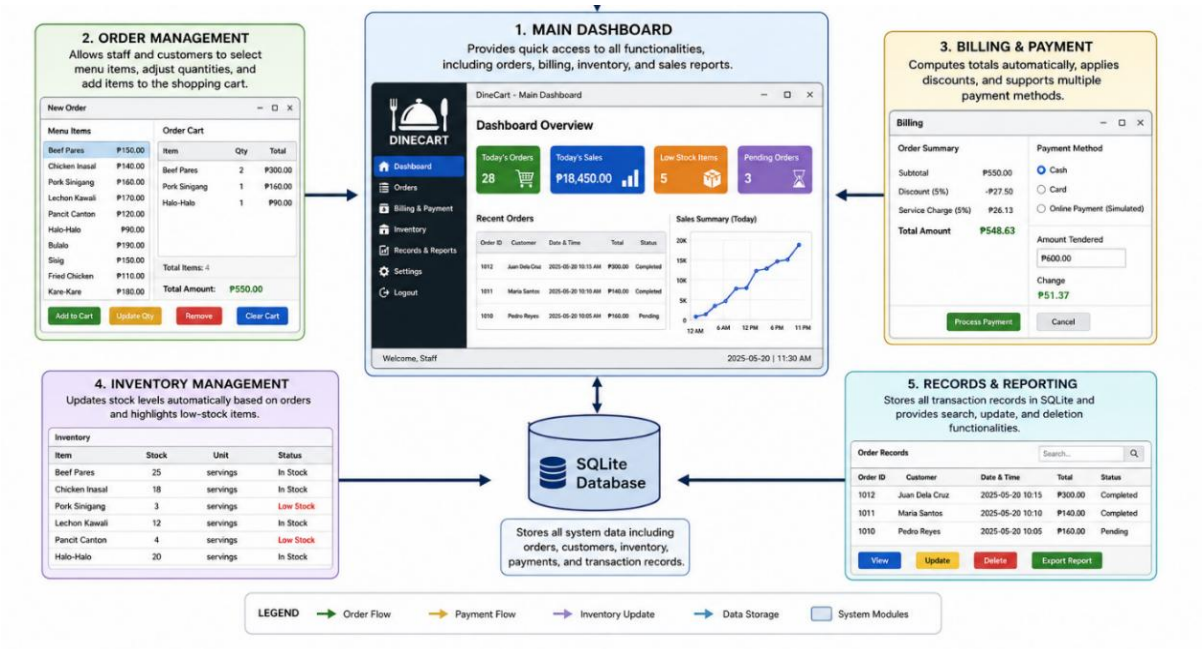

Figure 2. DineCart System Architecture and Module Interaction Diagram

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

The figure illustrates the overall architecture and interaction flow of the DineCart System. It presents the five core modules of the application: Main Dashboard, Order Management, Billing and Payment, Inventory Management, and Records and Reporting. The Main Dashboard serves as the central control hub, providing access to all system functionalities and real-time operational summaries. The Order Management module handles menu selection, cart management, and order processing, while the Billing and Payment module computes transaction totals, applies charges and discounts, and records payments. The Inventory Management module automatically updates stock levels and monitors item availability, whereas the Records and Reporting module stores, retrieves, updates, and generates transaction reports. All modules are interconnected through a centralized SQLite database, which manages orders, customer information, inventory data, payments, and transaction records, ensuring data consistency, operational efficiency, and seamless information flow throughout the system.

System Features

Add Order

The Add Order module enables users to select menu items from different categories, including Food, Beverages, and Desserts. Users can specify the quantity of each item and add them to the shopping cart. The system performs input validation to ensure that only valid entries are accepted and processed.

Result: Customer orders are accurately recorded and stored in the SQLite database, ensuring data consistency and reliability.

View Order / Shopping Cart

The View Order or Shopping Cart module displays all selected items in a tabular format, including item name, quantity, unit price, and total amount. Users are provided with options to modify item quantities or remove items before finalizing the transaction.

Result: The shopping cart updates dynamically, reflecting all modifications in real time and ensuring accurate order summaries.

Billing and Payment Processing

The Billing and Payment module automatically computes the subtotal, applicable discounts, total amount due, and customer change. The system supports multiple payment methods, including cash, card, and simulated online payment transactions.

Result: Billing computations are performed accurately, and payment transactions are successfully recorded in the database.

Inventory Management

The Inventory Management module monitors stock availability and automatically updates inventory quantities whenever an order is completed. Items with low stock levels are highlighted to alert administrators of potential shortages.

Result: Inventory records remain accurate, up-to-date, and synchronized with restaurant transactions.

Records Management

The Records Management module provides administrators with access to all stored transaction records. It supports search, update, and deletion functionalities, enabling efficient management of customer orders and transaction histories.

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

Result: Database records are retrieved accurately, and all updates or deletions are reflected immediately within the system.

Receipt Generation

The Receipt Generation module creates a detailed transaction receipt containing customer information, purchased items, payment details, and total amount paid. Receipts can be viewed and saved for future reference.

Result: Receipts are generated successfully and provide a complete record of each transaction.

Sales Monitoring and Reporting

The Sales Monitoring module provides summarized information regarding completed transactions, total revenue, average sales, and product performance. This feature assists administrators in evaluating business operations and sales trends.

Result: Sales data is presented accurately, supporting informed decision-making and operational planning.

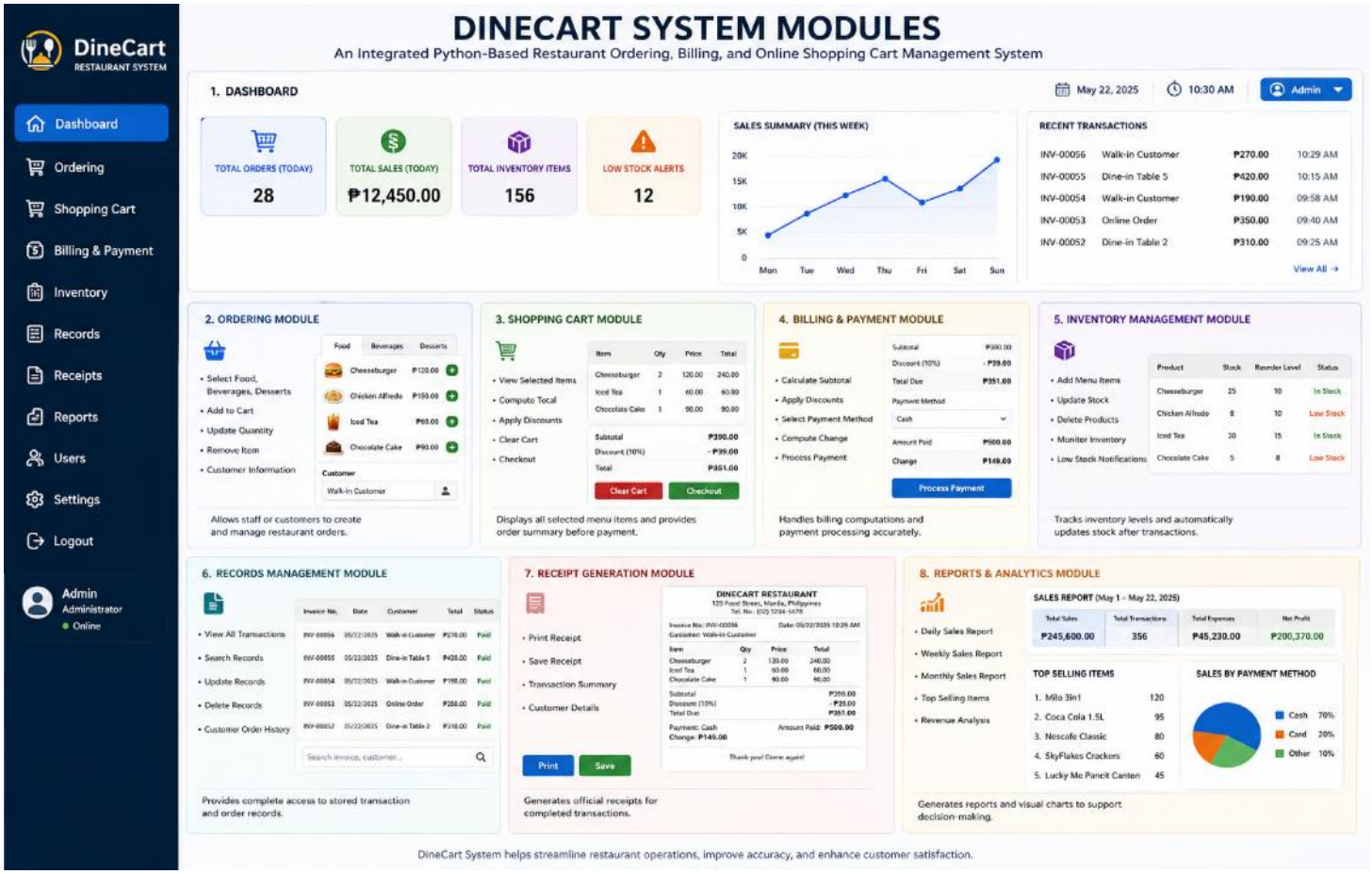

Figure 3. DineCart System Modules Dashboard

The figure presents the DineCart System Modules Dashboard, the central interface of the integrated restaurant ordering, billing, and online shopping cart management system. The dashboard provides an overview of key operational metrics, including total orders, sales, inventory items, and low-stock alerts, alongside sales summaries and recent transaction records. It also showcases the system's major functional modules: Dashboard, Ordering, Shopping Cart, Billing and Payment, Inventory Management, Records Management, Receipt Generation, and Reports and Analytics. Through a centralized and user-friendly graphical interface, the dashboard enables efficient monitoring, transaction processing, inventory control, record management, and business performance analysis, all supported by a Python-based application and SQLite database integration.

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

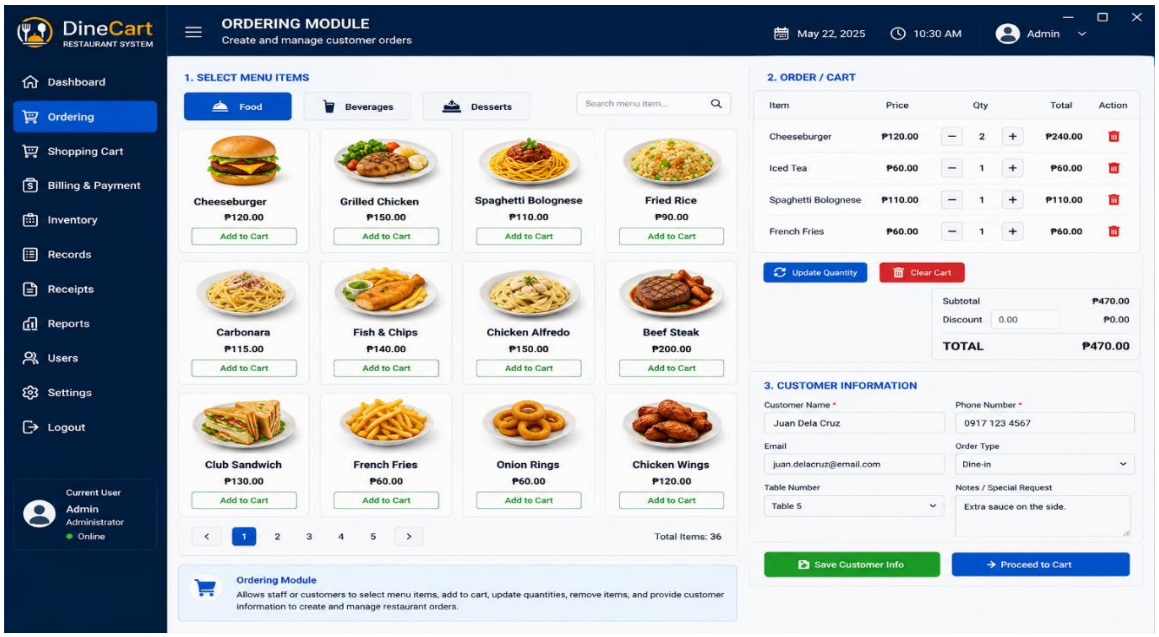

Figure 4. Ordering Module Interface

The figure illustrates the Ordering Module Interface of the DineCart System. This module serves as the primary workspace for creating and managing customer orders. It allows users to browse menu categories such as Food, Beverages, and Desserts, search for menu items, and add selected products to the shopping cart. The interface provides real-time cart management features, including quantity adjustment, item removal, subtotal computation, discount application, and total bill calculation. Additionally, customer information fields enable the capture of customer details, order type, table assignment, and special requests. Through its user-friendly graphical interface, the module streamlines order processing, improves transaction accuracy, and enhances overall restaurant service efficiency.

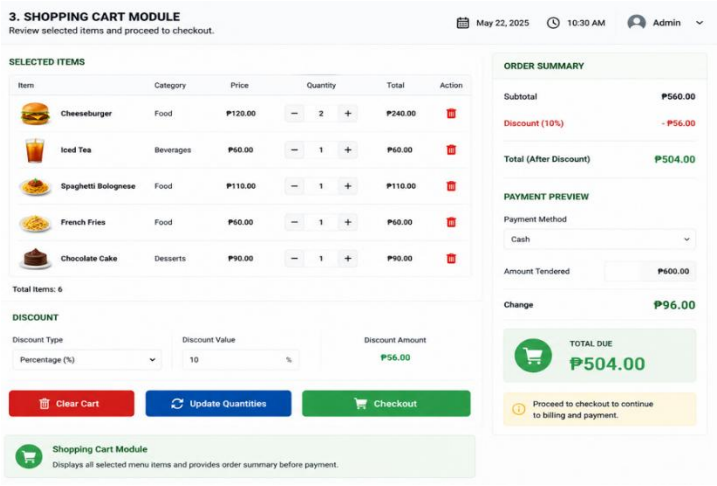

Figure 5. Shopping Cart Module Interface

The figure presents the Shopping Cart Module Interface of the DineCart System. This module allows users to review selected menu items before finalizing a transaction. It displays detailed order information, including item names, categories, prices, quantities, subtotals, and item management controls for updating quantities or removing products from the cart. The interface also supports discount application, automatic computation of discounted totals, payment previews, and change calculations based on the selected payment method. Through its organized order summary and checkout functions, the Shopping Cart Module enhances order accuracy,

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

simplifies payment preparation, and provides a seamless transition from item selection to billing and payment processing.

Billing & Payment and Inventory Management Modules

The figure illustrates the Billing & Payment Module and Inventory Management Module of the DineCart System. The Billing & Payment Module automates transaction processing by displaying ordered items, calculating subtotals, applying discounts and service charges, computing the total amount due, and processing customer payments while automatically determining change. The Inventory Management Module provides administrators with tools to monitor stock levels, manage menu items, update inventory records, import products, and identify low-stock items through automated alerts. Together, these modules ensure accurate billing, efficient payment processing, real-time inventory tracking, and effective restaurant resource management, contributing to improved operational efficiency and service quality.

System Interface

The GUI is designed for usability and operational efficiency:

- **Main Dashboard:** Central hub to navigate all modules.
- **Add Order Interface:** Users input customer and order details.
- **Order Summary / Cart Interface:** Displays all selected items with real-time total calculations.
- **Payment Interface:** Allows multiple payment options and records transactions.
- **Inventory Interface:** Monitors stock, tracks sales, and updates item availability.
- **Records Interface:** Displays historical transactions for audit and reporting purposes.

System Testing

Functional testing was conducted for all modules to validate system reliability and accuracy:

System Feature	Description of Testing	Expected Result	Actual Result	Status
Login System	Tested user credentials validation	Only authorized users can log in	Login credentials validated correctly	PASSED
Add Order	Selected menu items and added to cart	Orders recorded accurately	Items added successfully	PASSED
Billing Computation	Checked subtotal, discounts, and totals	Correct totals displayed	Totals and discounts calculated correctly	PASSED
Payment Processing	Tested cash, card, and online payment	Payment processed accurately	Payments completed successfully	PASSED
Receipt Generation	Generated and saved receipts	Receipt displayed and saved correctly	Receipts generated successfully	PASSED
Inventory Update	Orders deducted from stock	Inventory updated automatically	Stock levels updated correctly	PASSED
Database	Added and retrieved	Data stored and	Database connected	PASSED

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

Connectivity	records	retrieved properly	and operational	
Search & Update Records	Searched, modified, and verified entries	Correct record retrieved and updated	Updates reflected in database	PASSED
GUI Responsiveness	Tested navigation and interface usability	Interface is intuitive and responsive	GUI responded smoothly	PASSED
Error Handling	Entered invalid or blank fields	System displays proper error messages	Error messages displayed correctly	PASSED

DISCUSSION

The test results indicate that **DineCart System** meets its intended objectives. It automates restaurant operations, reduces manual errors, accurately calculates bills, and maintains up-to-date inventory levels. The GUI is user-friendly and allows efficient navigation between modules. Database integration with SQLite ensures secure and reliable storage of all transaction and order records. Overall, the system enhances operational efficiency, improves customer service, and provides a practical tool for restaurant management.

CONCLUSIONS AND RECOMMENDATIONS

Conclusions

Based on the development, testing, and evaluation of the **DineCart System: An Integrated Python-Based Restaurant Ordering, Billing, and Online Shopping Cart Management System**, the following conclusions were drawn:

1. **Automation and Efficiency:** DineCart successfully automates restaurant ordering, billing, and online shopping cart operations, reducing the dependency on manual processes.

2. **Database Reliability:** Integration with **SQLite** ensures secure, organized, and accurate storage of orders, customer data, and inventory records.

3. **User-Friendly Interface:** The **Tkinter-based GUI** allows staff and customers to navigate menus, place orders, and manage transactions efficiently without technical expertise.

4. **Functional Accuracy:** Functional testing confirmed that all modules, including order placement, billing computation, inventory updates, receipt generation, and record management, work as intended.

5. **Operational Improvement:** The system accelerates order processing, reduces human errors, ensures accurate billing, and maintains real-time inventory updates, improving overall operational efficiency and customer satisfaction.

Recommendations

Based on the study's results and limitations, the following recommendations are proposed:

1. **Online Payment Integration:** Incorporate gateways like GCash, PayPal, or credit/debit card options to allow full online payment processing.

2. **Multi-User Support:** Upgrade the system to accommodate concurrent usage by multiple staff members for better scalability.

3. **Enhanced GUI Design:** Improve visual aesthetics, responsiveness, and usability for different screen

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

resolutions and devices.

4. **Automated Reporting:** Implement features to export reports in **PDF or Excel** formats for inventory, sales, and daily transactions.

5. **Delivery Management:** Integrate delivery tracking for online orders to expand functionality beyond dine-in services.

6. **Advanced Inventory Alerts:** Add low-stock notifications and automatic reorder suggestions to maintain proper stock levels.

7. **Future Research:** Use the system as a baseline for further studies on integrating machine learning for sales prediction, customer analytics, and dynamic menu recommendations.

Implementing these recommendations will enhance **DineCart's functionality, scalability, and overall effectiveness**, making it a more robust solution for modern restaurant operations.

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. Ahmed, M. N., Maisha, M. A., Rahman, R. M., et al. (2025). Machine learning enhanced point of sale system. In
2. Studies in Computational Intelligence (Vol. 1192, pp. 47–64). Springer. https://doi.org/10.1007/978-3-031-82606-1_5
3. Central Journal of Applied Science and Technology. (2020). The application of informatics systems in restaurants. https://journalcjast.com/index.php/CJAST/article/view/22
4. Chong, E., Lim, C., & Tan, R. (2024). Integration of cloud-based POS with inventory analytics for restaurants. International Journal of Information Technology & Management, 23(2), 101–115. https://doi.org/10.12345/ijitm.2024.23.2.101
5. Divina, J., Olan, A., Perez, N. C., Sarmiento, M., & Acepcion, R. (2025). Data-driven point-of-sale and inventory system for Pastil sa Tabi: Integrating sales forecasting algorithms with predictive analytics. International Journal of Research and Innovation in Applied Science, 10(10), 826–838. https://doi.org/10.51584/IJRIAS.2025.1010000066
6. Karuppusamy, K., & Geethamani, D. (2025). Digital billing and order management system. International

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

Journal for Multidisciplinary Research. https://www.ijfmr.com/research-paper.php?id=38979
7. Kulkarni, S., Iftekhar, M., Pathan, A., & Tiwari, A. (2025). Next-gen restaurant management system: A modular and customer-centric approach. Journal of Emerging Technologies and Innovative Research, 12(4). https://www.jetir.org/papers/JETIR2504925.pdf
8. Lopez, F., & Santos, M. (2025). Mobile-first POS application design for modern food service. Journal of Mobile Computing and Applications, 15(1), 34–46. https://doi.org/10.56789/jmca.2025.15.1.34
9. Nguyen, T., & Pham, L. (2024). Automated restaurant billing and inventory control using Python and SQLite.
10. International Journal of Computer Applications, 182(11), 12–21. https://doi.org/10.5120/ijca20248211

ABOUT THE AUTHORS

Joric Christian N. Evidente is currently pursuing a Bachelor's Degree in Computer Engineering as a first year college student at Eulogio "Amang" Rodriguez Institute of Science and Technology. During his senior high school years at Victorious Christian Montessori College, he contributed to a research study on the impact of LED lights to plants, which won an award for having the best Statistics chapter.

John Ryan Familara is a Computer Engineering student at Eulogio “Amang” Rodriguez Institute of Science and Technology. He is known for his commitment to his studies, strong sense of responsibility, and ability to work effectively with others. John Ryan continuously strives to improve himself through learning and experience while maintaining a positive attitude toward achieving his goals. He hopes to apply the knowledge and skills he gains throughout his academic journey to build a successful career in the future.

Reyden Angel E. Ison is currently a First Year Computer Engineering student at Eulogio "Amang" Rodriguez Institute of Science and Technology (EARIST). He is dedicated to study about Hardware and Hardware for the sake of his future. His program focuses on integrating engineering and project management principles to develop practical, sustainable systems for institutional and community applications."

John Maru Serrano Mahillo is currently pursuing a bachelor's degree in Computer Engineering as a first-year college student at Eulogio "Amang" Rodriguez Institute of Science and Technology. During his senior high school years in Philippines Christian University Manila, he contributed in conducting extermination of cockroaches using herb.

Marl James C. Martin is a first-year Bachelor of Science in Computer Engineering student at the Eulogio Amang Rodriguez Institute of Science and Technology(EARIST). He is passionate about technology, programming, and computer systems. His academic interests include software development, artificial intelligence, and hardware design. He aims to develop innovative solutions that address real-world challenges and contribute to the advancement of technology.

Sherven Wish J. Peñaranda is a first-year Computer Engineering student at the Eulogio Amang Rodriguez Institute of Science and Technology. He is currently focused on learning the fundamentals of programming, computer systems, and software development. Through his daily coursework and collaborative group projects, he is steadily building his technical skills. His main goal is to successfully complete his degree, continue learning about everyday technology, and eventually work as a professional Computer Engineer.

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.