APPLIED INFORMATION TECHNOLOGY AND COMPUTER SCIENCE
e-ISSN: 2773-5141

AITCS

Vol. 7 No. 1 (2026) 842-860
https://publisher.uthm.edu.my/periodicals/index.php/aitcs

Bliciousseries Ordering and Sales Management System

Lily Natasha Teo Ah Hing@Tieo Ah Hing[1], Hannani Aman[1]*

[1] *Fakulti Sains Komputer dan Teknologi Maklumat,*
Universiti Tun Hussein Onn Malaysia, Parit Raja, Batu Pahat, 86400, MALAYSIA

*Corresponding Author: hanani@uthm.edu.my
DOI: https://doi.org/10.30880/aitcs.2026.07.01.045

Article Info

Received: 15 June 2025
Accepted: 22 June 2026
Available online: 30 June 2026

Keywords

Order Management, Web-Based System, Referral System

Abstract

Bliciousseries, a cookie brand company based in Seri Medan, Johor, faces operational challenges in managing orders and payments manually, particularly during high-demand seasons such as Hari Raya Aidilfitri. The business currently relies on WhatsApp to communicate with agents, resulting in delays, miscommunication, and errors. This project presents the development of a web-based Ordering and Sales Management System using Python to streamline order processing, payment verification, and sales tracking for both agents and customers. The system focuses on enhancing agent-based order management with direct customer ordering capabilities and also referral and commission module. System testing with users from all roles confirmed that all relevant features functioned correctly and met user expectations. The system significantly improves efficiency by reducing manual errors, speeding up order processing, and enabling real-time sales tracking. Future enhancements will focus on integrating e-wallet payments, implementing an agent tier system, and developing a mobile-friendly version.

1. Introduction

Manual order and sales management is inefficient and error-prone, especially for businesses in the food and beverage industry operating during peak seasons. With the rise of digital commerce, many companies are shifting to online systems to optimize operations and reduce human errors.

Bliciousseries, a local cookie brand in Johor, currently manages its orders from 30 agents via WhatsApp. While this process is sufficient during low-demand periods, it becomes increasingly ineffective during festive seasons such as Hari Raya Aidilfitri. Additionally, the absence of an online ordering system limits the company's ability to reach a broader customer base. To overcome these challenges, the company has initiated the development of a digital Ordering and Sales Management System using Python.

The system is designed to digitize order tracking, streamline payment processing, and automate sales reporting. It incorporates a referral and commission system to incentivize agent performance and will be progressively enhanced to include customer-facing functionalities.

The objectives of this study are:

1. To design a web-based order and sales management system.
2. To implement core modules for agents, clerks, and administrators.
3. To evaluate the functionality of the system through testing.

The expected outcomes include improved order accuracy, enhanced tracking of commissions, and better accessibility for customers and agents. The system also supports real-time reporting to inform business decisions.

This paper is structured as follows: Section 2 reviews related work and existing systems. Section 3 outlines the adopted methodology. Section 4 presents the system's analysis and design. Section 5 describes implementation and testing. Section 6 concludes with findings and future work.

Applied Information Technology and Computer Science Vol. 7 No. 1 (2026) p. 842-860

2. Related Work

This section reviews technologies that are used in project development, which are Python with Flask framework, Jinja Templating and as-is Bliciousseries ordering and sales management system and 3 systems related to order and sales management which are Lizkrispychip Kerepek ordering system, Brew Beauty e-Commerce System and The Bliese System.

2.1 Python with Flask Framework

Flask is a lightweight and flexible web framework written in Python that is widely used for developing web applications and APIs. It follows a microframework architecture, meaning it does not require particular tools or libraries, making it ideal for small to medium-sized projects that need rapid development and customization. Flask supports extensions that add application features as if they were implemented in Flask itself, such as form validation, object-relational mappers, and user authentication. Its simplicity and integration with Python make it a popular choice for academic and commercial projects alike [1].

2.1.1 Jinja Templating

Jinja is a high-performance templating engine for Python that allows dynamic content generation in web applications by embedding expressions and logic directly into HTML templates. It supports features like template inheritance, macros, and automatic HTML escaping, making it a powerful tool for rendering secure and maintainable user interfaces. Commonly used with the Flask framework, Jinja facilitates a clean separation between application logic and front-end presentation [2].

2.2 As-Is Bliciousseries Ordering and Sales Management System

Currently, Bliciousseries relies on manual workflows and WhatsApp communication between agents and the administrative team. This informal process contributes to inefficiencies and inconsistencies, particularly during peak periods. Fig. 1 illustrates the existing ordering workflow using a Swimlane diagram.

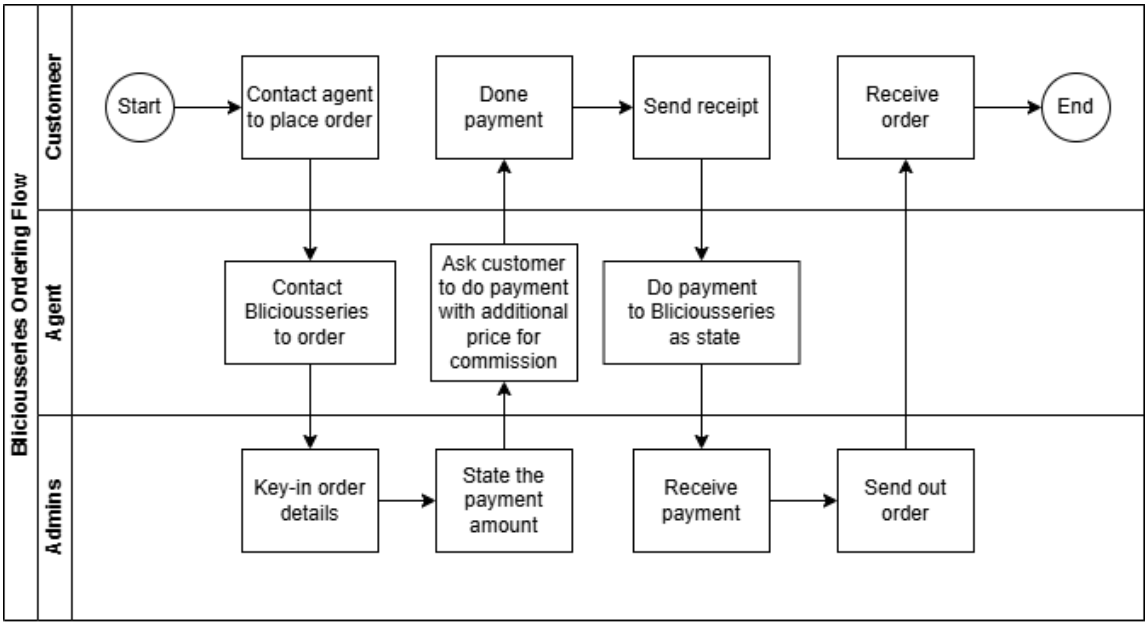


Fig. 1 *Swimlane Diagram for the Current Bliciousseries Ordering Flow (As-Is Model)*

2.3 Comparison with Existing Systems

Table 1 presents a comparison between the Bliciousseries system and three other related solutions: Lizkrispychip Kerepek Ordering System [3], Brew Beauty e-Commerce System [4], and The Bliese System [5].

Table 1 *System Comparison of existing system and proposed system*

Features/System	Lizkrispychip Kerepek ordering system	Brew Beauty e-Commerce System	The Bliese System	Bliciousseries System
Login and registration	✓	✓	✓	✓
Order Management	✓	✓	✓	✓
Payment Management	✓	✓	✓	✓
Commission Tracking	✗	✗	✗	✓
Referral Program	✗	✗	✗	✓
Generate Report	✓	✓	✓	✓

The comparison highlights the unique features of the Bliciousseries system, such as referral and commission tracking.

3. Methodology

This section will discuss the methodology used in the development of the ordering system.

3.1 Prototype Model

The Prototyping Model was chosen for this project because it enables early development of a working system that can be refined through user feedback. This approach helps identify issues early, improves usability, and ensures the final product aligns with user needs [6].

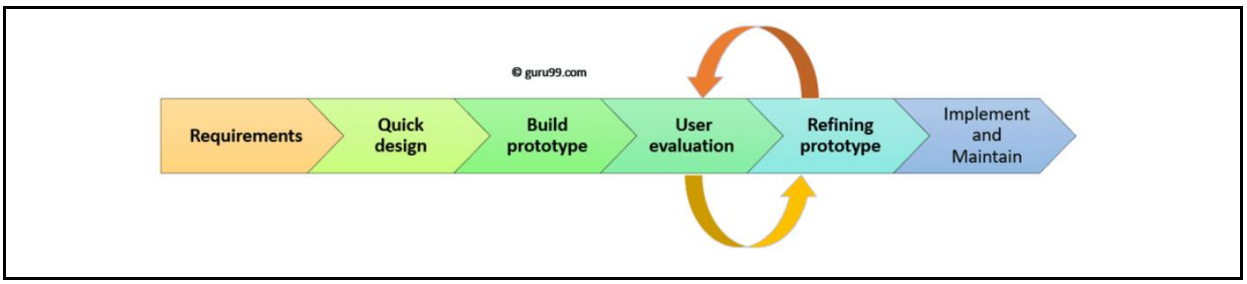


Fig. 2 *Prototyping Model Phases [7]*

Table 2 outlines the tasks and corresponding outputs for each phase of the Prototyping Model used in developing the Bliciousseries Ordering and Sales Management System. The process began with Requirements Gathering and Analysis, where user needs and project goals were identified, resulting in documents like the project proposal and user requirements. In the Quick Design phase, essential design elements such as UML diagrams, UI layouts, and database schemas were created. Prototype Development followed, focusing on implementing core features into a functional version of the system. The prototype was then evaluated during the User Evaluation phase through feedback from stakeholders. Based on this input, the Prototype Refinement phase involved enhancing system features and addressing issues. Finally, the Implementation and Maintenance phase included full system testing, deployment, and continuous support to ensure smooth operation and future updates.

Table 2 *Task and Output of Each Phase*

Phase	Task	Output
Phase 1: Requirements Gathering and Analysis	1. Conduct initial meetings and interviews with stakeholders (agents, admins, and customers). 2. Identify project objectives, problem statement, scope, and system goals.	1. Project proposal 2. Gantt chart 3. User requirements document

Table 2: *(cont)*

Phase	Task	Output
	3. Determine project timeline, activities, and expected deliverables 4. Analyze the current system's strengths, weaknesses, and areas for improvement.	
Phase 2: Quick Design	1. Design UML diagrams (use case, activity, and sequence diagrams) for the proposed system. 2. Design user interface (UI) for the system. 3. Design the database structure to support the system's requirements.	1. Use case diagram 2. Activity diagram 3. Sequence diagram 4. Class diagram 5. Database schema
Phase 3: Development of Prototype	1. Develop a functional prototype of the proposed system based on the design and specifications. 2. Implement core features, such as order management, payment options, and referral tracking.	1. System prototype
Phase 4: User Evaluation	1. Present the prototype to stakeholders (admins, agents, and customers) for evaluation. 2. Gather feedback on the prototype's functionality and usability.	1. User evaluation report
Phase 5: Prototype Refinement	1. Refine and improve the prototype based on the feedback from stakeholders. 2. Enhance system features, improve user interface, and resolve any identified issues.	1. Refined system prototype
Phase 6: Implementing and Maintenance	1. Perform system testing on the entire system, including functional testing and user acceptance testing. 2. Deploy the system to a live production environment. 3. Provide ongoing maintenance, support, and system updates based on user feedback.	1. Bliciousseries Ordering and Sales Management System (operational)

4. Analysis And Design

This section discusses the analysis and design of the system.

4.1 System Requirement Analysis

The system requirements analysis covered user requirements, functional requirement and non-functional requirement. Table 2 shows the user requirements, consisting of the end users' and stakeholders' needs, expectations and specifications for the system.

Table 3 *User Requirements*

User	Requirement
Owner	• Able to register for clerk. • Able to login to the system. • Able to review the full report.
Clerk	• Able to register for agent. • Able to login to the system. • Able to manage product.

Table 3: *(cont)*

User	Requirement
	• Able to manage order. • Able to manage payment and commission. • Able to check on agent's referral code. • Able to manage sales performance • Able to review certain report.
Agent	• Able to login to the system. • Able to manage order. • Able to manage payment and commission. • Able to get a referral code. • Able to review personal sale report.
Customer	• Able to register. • Able to login to the system. • Able to manage order. • Able to manage payment and commission.

Functional requirements define the specific actions and behaviors the system must perform to fulfill user needs and achieve business objectives. These include the tasks the system must support for each user role, such as registration, login, order management, and payment handling. Table 4 summarizes the system's functional requirements based on its key modules.

Table 4 *System functional Requirements*

Modules	User	Requirement
Registration module	Customer	• The system shall allow customer to register an account by providing their name, mobile phone, email, date of birth, password, and confirm password.
	Clerk	• The system shall allow clerk to register an account for agent by providing their name, mobile phone, email, date of birth, password, and confirm password.
	Owner	• The system shall allow clerk to register an account for clerk by providing their name, mobile phone, email, date of birth, password, and confirm password.
Login module	All	• The system shall allow users to login to the system.
Product management module	Clerk	• The system shall allow clerks to add new products. • The system shall allow clerks to edit existing product details. • The system shall allow clerks to delete products from the system.
Order Management Module	Customer	• The system shall allow customers to place and cancel orders.
	Agent	• The system shall allow agents to place and cancel orders.
	Clerk	• The system shall allow clerks to manage and update orders.
Payment and Commission Management Module	Agent	• The system shall allow the agent to choose a payment option ("Pay Now" or "Pay Later") and upload a payment receipt if "Pay Now" is selected.
	Customer	• The system shall allow the customer to enter a referral code and upload a payment receipt.

Applied Information Technology and Computer Science Vol. 7 No. 1 (2026) p. 842-860

Table 4: *(cont)*

Modules	User	Requirement
	Clerk	• The system shall allow the clerk to approve or decline the agent's commission based on order details and payment verification.
Referral Code Management Module	**Agent**	• The system shall generate and associate a unique referral code with the agent's account upon creation. • The system shall allow the agent to copy or share their referral code with customers.
	Clerk	• The system shall allow the clerk to view and verify the agent's referral code in the "Agent List" for tracking usage and commission purposes.
Generate Sales Report Module	Clerk	• The system shall allow the clerk to navigate to the report generation page. • The system shall allow the clerk to select a date range for report generation. • The system shall display the sales report data for the selected date range.
Sales Performance and Reporting Module	Owner	• The system shall allow the owner to access sales performance data. • The system shall allow the owner to select reporting criteria. • The system shall allow the owner to review the performance of agents and customer interactions.
	Clerk	• The system shall allow the clerk to view the total sales for a selected time period.
	Agent	• The system shall allow the agent to view their performance data.

A non-functional requirement defines the quality attributes, system performance, and constraints of the system, such as security, reliability, scalability, and usability, that the system must meet in order to operate effectively and efficiently. [9] Table 4 shows the non-functional requirements of the system.

Table 5 *Non-functional Requirements*

Requirement	Description
Performance	• The system shall have a reasonable response time.
Usability	• The system should be easy to use with a simple interface. • The system shall have a consistent layout across pages.
Security	• User log in password should have minimum length of 8 characters, mix of letters and numbers. • The system shall deny login credentials that do not match with database.
Availability	• The system shall always be available for use.
Operational	• The system shall be able to work with any web servers.

4.2 System Analysis

This report applied specific diagrams to detail the system's design which is use case diagram shown in Appendix A, outlines the interactions between users and the system, highlighting the roles and functionalities assigned to each user group. Another diagram applied is class diagram, provided in Appendix B, illustrates the system's structure by defining its classes, attributes, and relationships.

Additional diagram applied is flowchart, which has been developed to represent the workflow and logical sequence of operations within the system. The flowchart simplifies the understanding of the processes involved in executing key functions and is available in Appendix C.

5. Implementation and Testing

This section discusses about the implementation and testing of the developed system.

5.1 Implementation

The system was implemented following a modular development approach based on the Prototyping Model. A total of eight modules were developed to support the core functionalities of the system. These modules include the registration module, login module, product management module, order management module, payment and commission management module, referral code management module, generate sales report module, and sales performance and reporting module. The backend was developed using Python with the Flask framework, while the frontend was designed using HTML, CSS, and Jinja templating. MySQL was used as the relational database to store and manage persistent data across all modules.

5.1.1 Registration Module

This section introduces the registration functionality based on user roles. Customers can register through a public form, while Clerks and Agents must be registered by the Owner and Clerk respectively. Fig.3, 4 and 5 illustrate each registration page, showcasing fields for user details such as name, contact information, and password. This ensures secure role-based account creation.

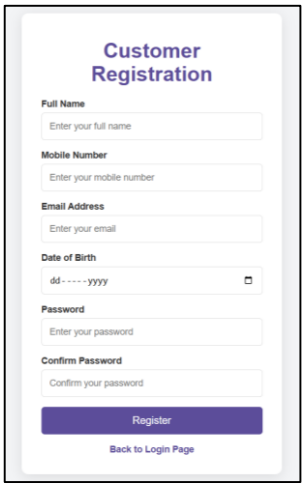


Fig. 3 *Customer Registration Page*

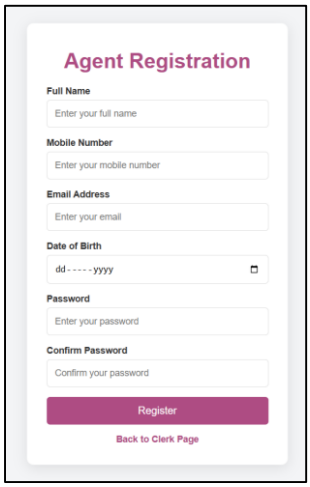


Fig. 4 *Agent Registration Page (By Clerk)*

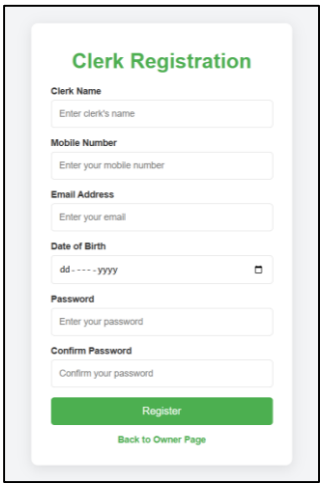


Fig. 5 *Clerk Registration Page (By Owner)*

5.1.2 Login Module

The login module supports access for all four roles which are owner, clerk, agent, and customer. Each user enters their credentials and is authenticated before being redirected to their respective dashboard. Fig. 6 shows a unified login page, which manages secure entry to the system and ensures that users access only the modules assigned to their role.

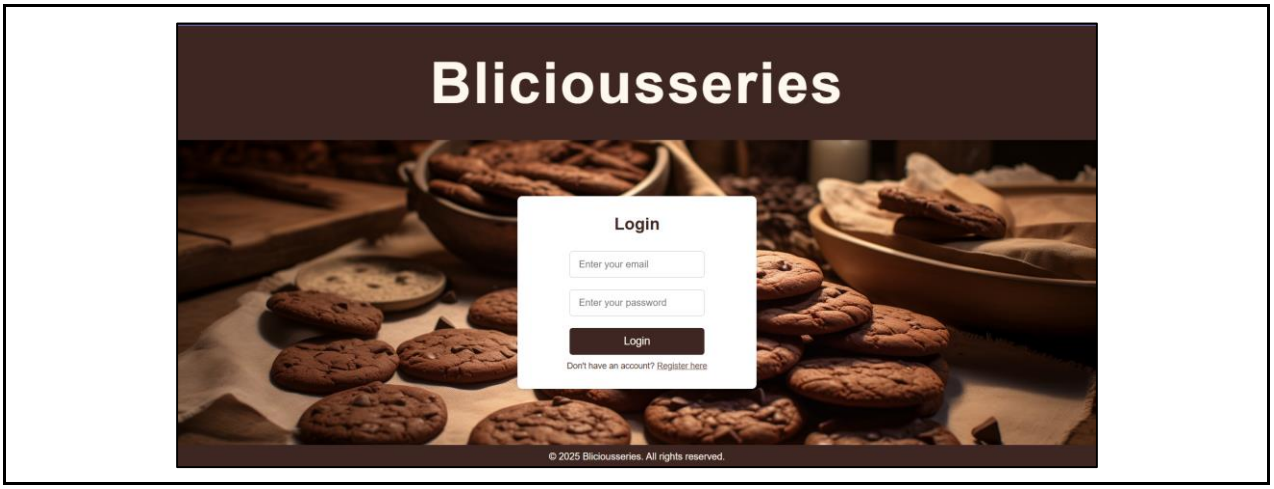


Fig. 6 *Login Page (All Roles)*

5.1.3 Product Management Module

This module enables clerks to manage product information including names, descriptions, quantities, images, and pricing for both agents and customers. It supports product creation, editing, and deletion. Fig. 7 shows the product management page used by clerks, while Fig. 8 presents the form for adding new products. Fig. 9 displays the agent's product view with pricing and commission information, and Fig. 10 illustrates the customer's product view with retail pricing.

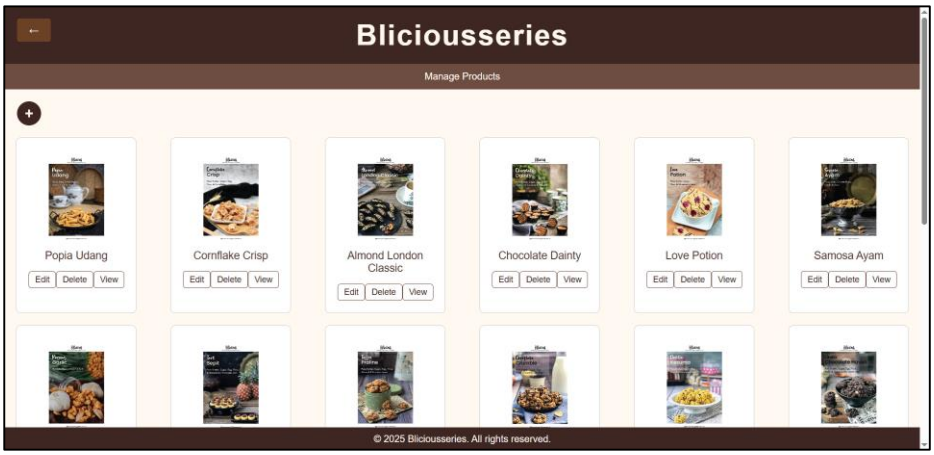


Fig. 7 *Clerk Product Management Page*

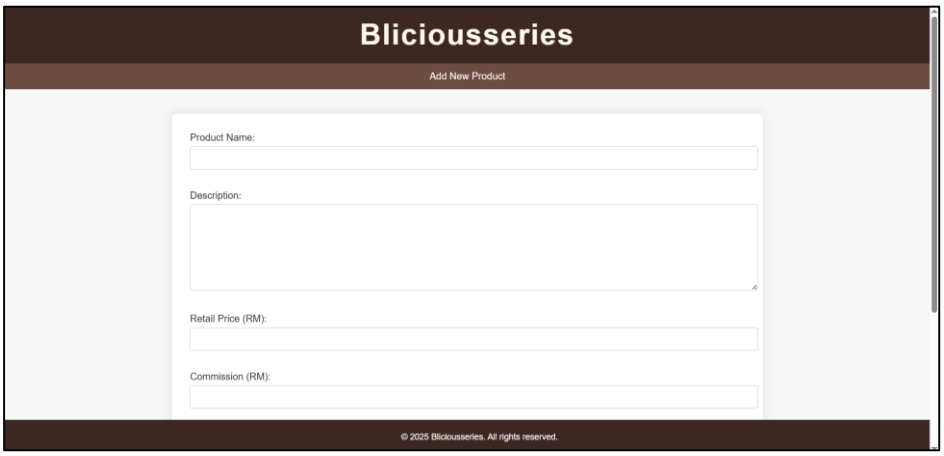


Fig. 8 *Clerk Add Product Form*

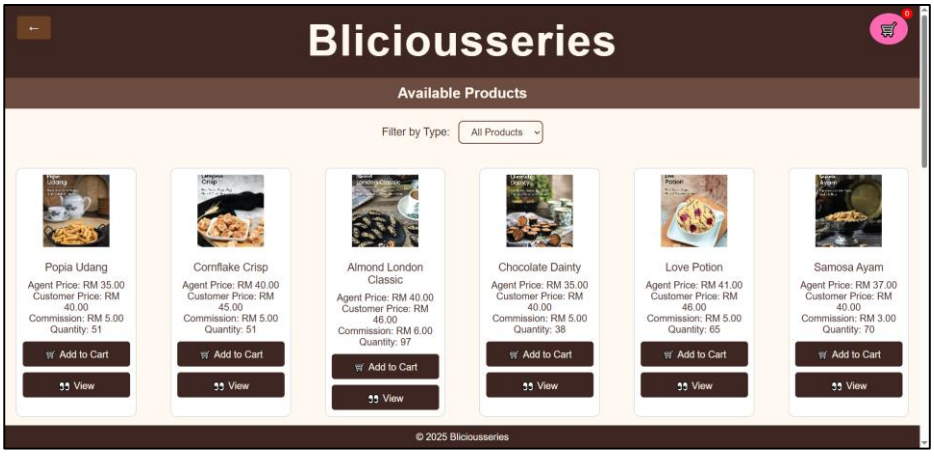


Fig. 9 *Agent Product View with Pricing and Commission*

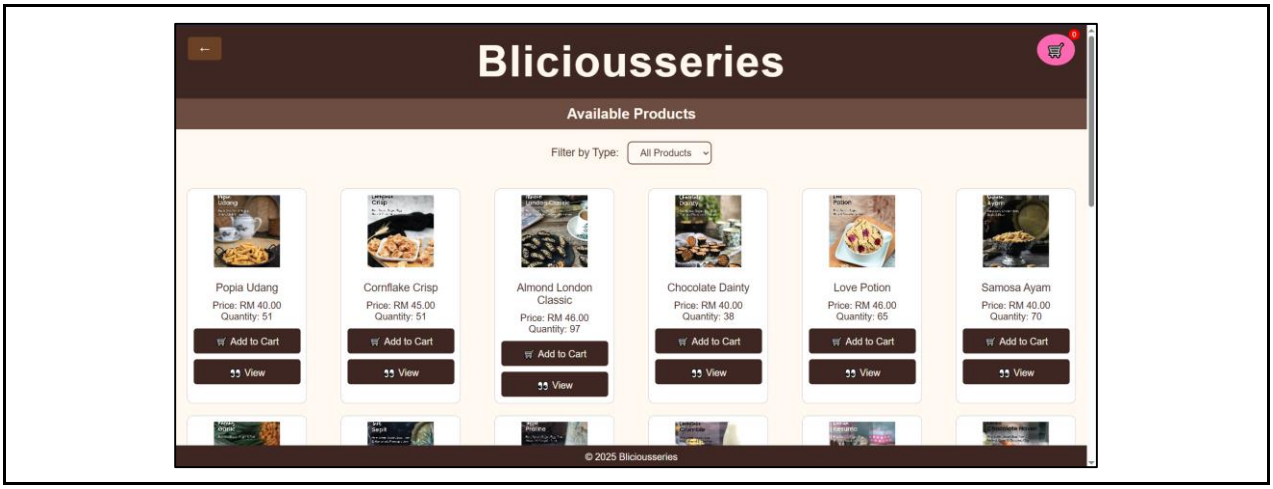


Fig. 10 *Customer Product View with Retail Pricing*

5.1.4 Order Management Module

The order management process begins with customers and agents placing orders, each assigned a unique ID and a status that can be updated by the clerk. Agents can cancel pending orders, while clerks are responsible for approving or declining them based on payment verification. Fig. 11 shows the customer's order history and status view, while Fig. 12 displays the agent's order view. Fig. 13 illustrates the clerk's interface for managing pending and approved orders, and Fig. 14 presents the dedicated interface for handling "Pay Later" orders.

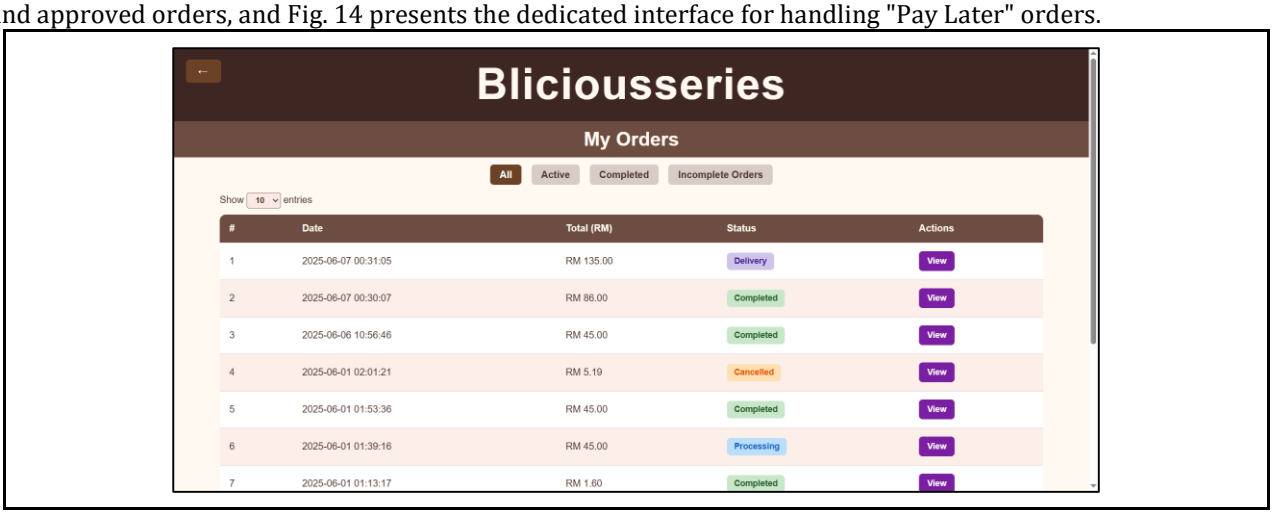


Fig. 11 *Customer Order History and Status View*

Fig. 12 *Agent Order History and Status View*

Fig. 13 *Clerk Manage Orders Page (Pending and Approved)*

Fig. 14 *Clerk Pay Later Orders Management*

5.1.5 Payment and Commission Management Module

Since this system does not use an online payment gateway, users are required to upload payment receipts. Agents can choose between "Pay Now" or "Pay Later" options, while customers must pay immediately. Clerks are responsible for reviewing the uploaded receipts and updating the order status accordingly. Commission is calculated automatically based on the order. Fig. 15 shows the agent's payment upload interface with the pay later option, Fig. 16 displays the customer's payment upload interface, and Fig. 17 illustrates the clerk's receipt approval and status update panel.

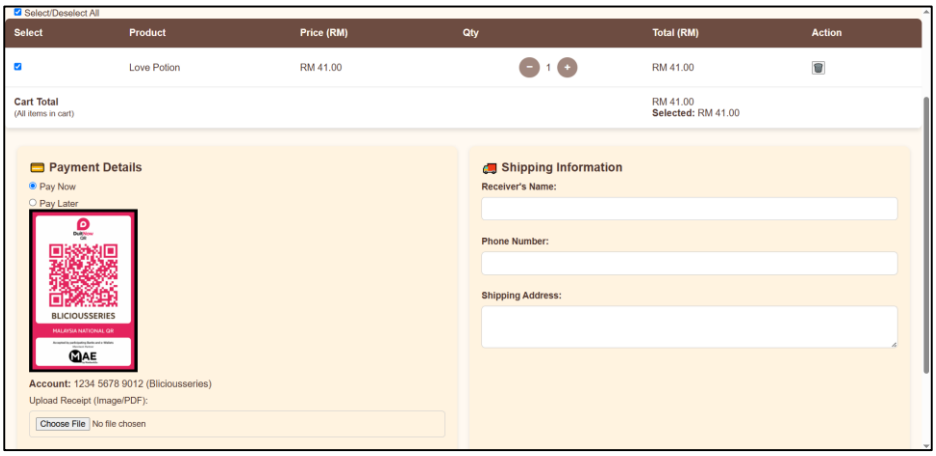


Fig. 15 *Agent Payment Upload and Pay Later Option*

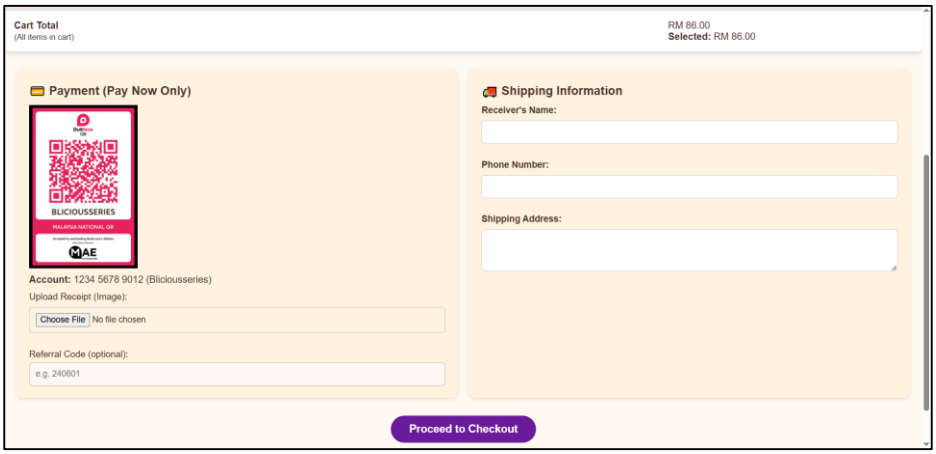


Fig. 16 *Customer Payment Upload Interface*

Fig. 17 *Clerk Receipt Approval and Status Update*

5.1.6 Referral Code Management Module

Agents are automatically assigned a referral code upon registration. When customers apply this code during checkout, the system awards the corresponding agent a commission. Fig. 18 shows the clerk's view of the agent list with referral code management, while Fig. 19 displays the agent's overview of their referral code performance and usage.

Fig. 18 *Clerk Agent List with Referral Code Management*

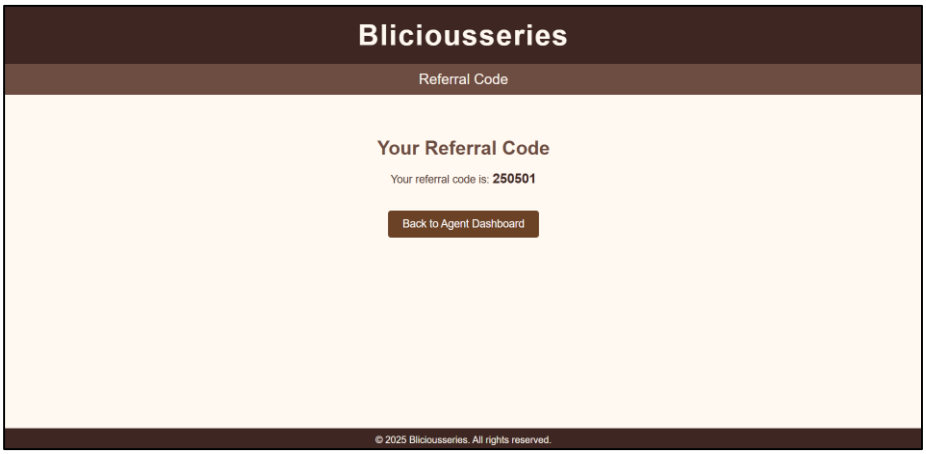


Fig. 19 *Agent Referral Code Overview*

5.1.7 Generate Sales Report Module

This module allows clerks to generate sales reports filtered by date and category. These reports display key metrics such as total sales, number of orders, and detailed breakdowns by agent, customer, or product. Fig. 20 presents a sample of the clerk's sales report interface, showcasing both tabular and graphical formats to support effective sales monitoring and informed decision-making. The reports can also be printed in PDF format for documentation and sharing purposes.

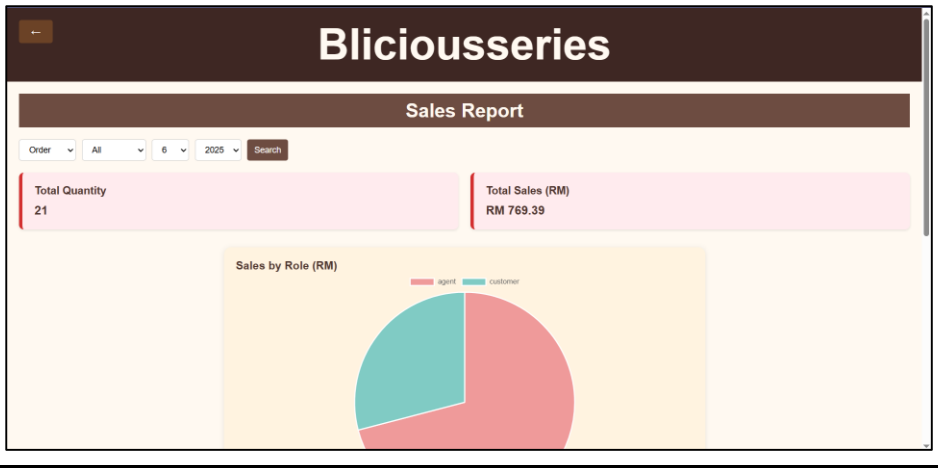


Fig. 20 *Clerk Sales Report*

5.1.8 Sales Performance and Reporting Module

This section provides an overview of the sales reporting tools available to each major user role. Fig. 21 presents the agent's sales report, which displays their personal sales and commission records. Fig. 22 presents the owner's monthly sales report page, including summaries, top-performing agents, and product sales rankings. Fig. 23 presents the clerk's login and logout logs used to monitor system access and agent performance. These figures demonstrate how the reporting tools are tailored to support the specific responsibilities of each role.

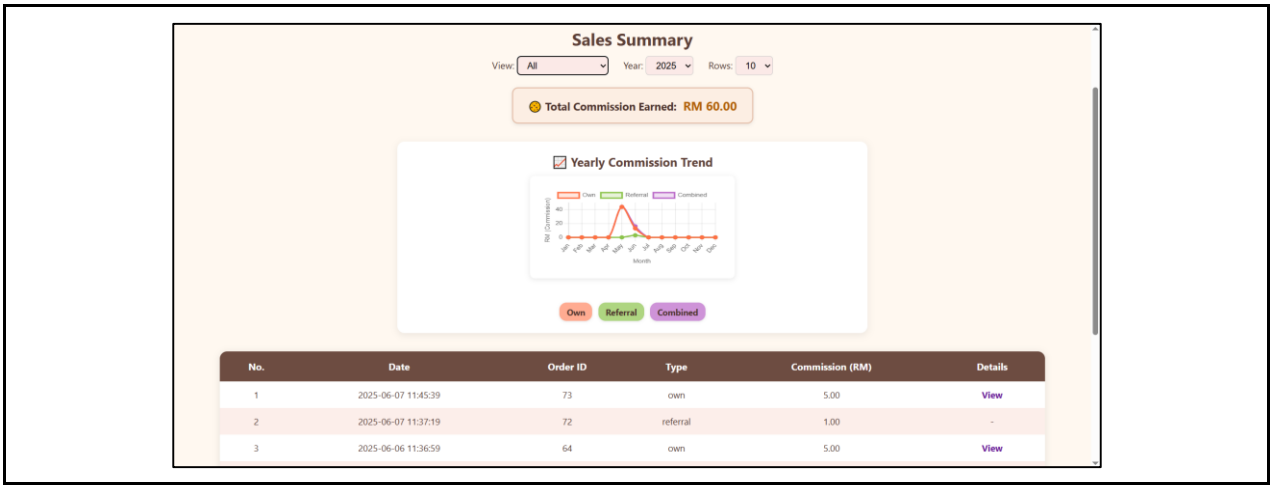


Fig. 21 *Agent Sales Report*

Fig. 22 *Owner Monthly Sales Report Page*

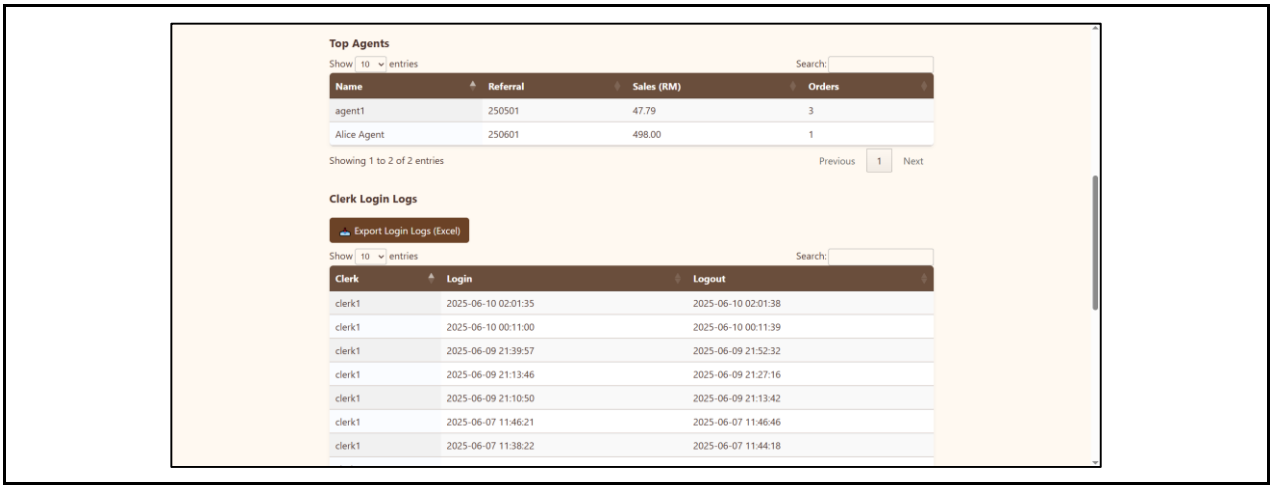


Fig. 23 *Clerk Login and Logout Logs*

5.2 Testing

A structured System Testing Form was distributed to four testers representing each user role.

5.2.1 System Testing

Each tester marked the test cases as either pass or not applicable, depending on whether the feature was relevant to their role. All role-relevant test cases were successfully passed, confirming that the system functions as

expected for each user type. Irrelevant features that did not apply to a tester's role were excluded from their testing scope.

Table 5 summarizes the test case results based on each user role. All relevant features assigned to each role were successfully tested and marked as "Pass," confirming that the system performs as intended. Features not applicable to a particular role were appropriately excluded from their testing scope. Overall, the results demonstrate that the system is functionally complete and reliable across all user types.

Table 5 *Test Case Results by User Role*

Test Case	Feature	Owner	Clerk	Customer	Agent
TC-01	Login	Pass	Pass	Pass	Pass
TC-02	Dashboard	Pass	Pass	Pass	Pass
TC-03	Product Page	-	-	Pass	Pass
TC-04	Cart & Checkout	-	-	Pass	Pass
TC-05	Referral Code	-	-	Pass	-
TC-06	Pay Later Option	-	-	-	Pass
TC-07	Upload Receipt	-	-	Pass	Pass
TC-08	Approve/Decline Orders	-	Pass	-	-
TC-09	Sales Reports	Pass	Pass	-	Pass
TC-10	Product Stock	-	Pass	-	Pass
TC-11	Owner Reports	Pass	-	-	-
TC-12	Logout Security	Pass	Pass	-	Pass

5.2.2 User Feedback Summary

Feedback was collected from all four testers after completing their test cases. All of them found the system very easy to use. The Owner, Helyzawati Binti Zaharan, liked the referral code feature but felt the system only partially met her expectations. She suggested adding an agent tier system. The Clerk, Nur Liyana, said the system met her expectations and suggested adding bulk selection in order management. Customer tester Balqis found the system easy to understand and suggested adding e-wallet payment options. The Agent, Alya Maisarah, liked the sales report with graphs and suggested showing daily sales. Overall, testers were satisfied and gave helpful suggestions for future improvements.

6. Conclusion

The Bliciousseries Ordering and Sales Management System was developed to solve key challenges faced by the company in managing agent-based orders through manual methods such as WhatsApp. By successfully achieving its three main objectives, the system has been designed using an object-oriented approach, developed with complete functionalities for all user roles, and verified through structured system testing. The implementation of this web-based system improves order tracking, payment processing, and sales reporting, providing a more efficient and organized platform for both agents and the company. It also allows customers to place orders directly, helping the business expand its reach and improve customer experience.

While the system offers many advantages, such as reducing errors and increasing productivity, it also has some limitations, including the need for internet access, lack of e-wallet integration, and the absence of a mobile application. These areas present opportunities for future development. Planned improvements include adding more payment options, introducing agent tier features, enabling daily sales tracking, and developing a mobile version. Overall, the system marks a significant step forward in digitizing Bliciousseries' operations and sets a strong foundation for continued growth and enhancement.

Acknowledgement

The authors would like to thank the Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia for its support.

Conflict of Interest

Authors declare that there is no conflict of interests regarding the publication of the paper.

Author Contribution

This journal requires that all authors take public responsibility for the content of the work submitted for review. The contributions of all authors must be described in the following manner:

The authors confirm contribution to the paper as follows: ***study conception and design:*** *Lily Natasha Binti Teo Ah Hing@Tieo Ah Hing, Hannani Binti Aman;* ***data collection:*** *Lily Natasha Binti Teo Ah Hing@Tieo Ah Hing, Hannani Binti Aman;* ***analysis and interpretation of results:*** *Lily Natasha Binti Teo Ah Hing@Tieo Ah Hing, Hannani Binti Aman;* ***draft manuscript preparation:*** *Lily Natasha Binti Teo Ah Hing@Tieo Ah Hing, Hannani Binti Aman. All authors reviewed the results and approved the final version of the manuscript.*

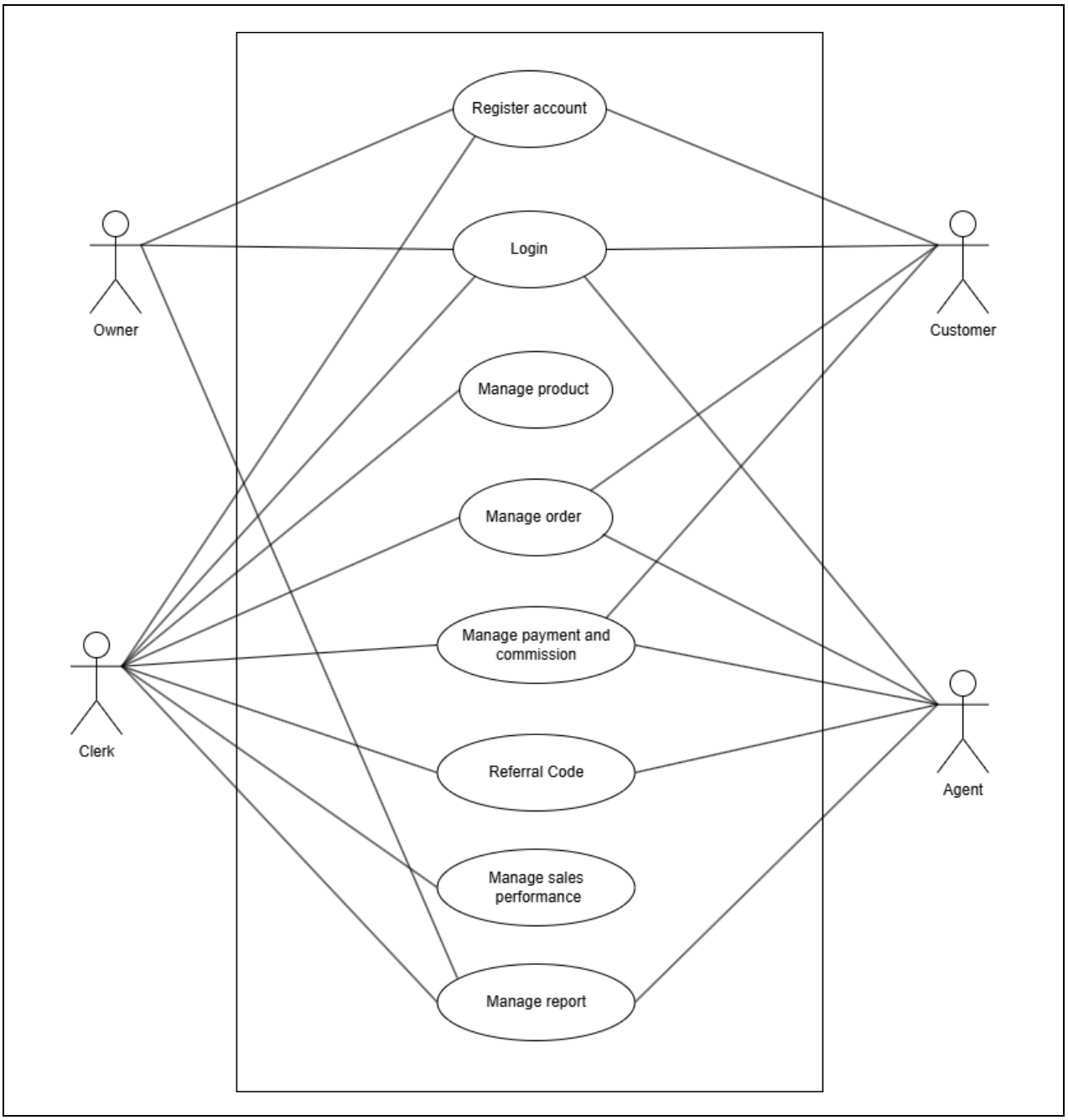
References

[1] M. Grinberg, *Flask Web Development: Developing Web Applications with Python*, 2nd ed. Sebastopol, CA: O'Reilly Media, 2018.
[2] H. M. Rashid and R. Aman, "Web development using Flask: A Python-based micro web framework," *International Journal of Computer Science and Mobile Computing*, vol. 11, no. 4, pp. 101–107, 2022.
[3] N. L. B. C. Mu'minin, *Lizkrispychip Kerepek Ordering System (Report)*, Batu Pahat, Malaysia: PTTA, Universiti Tun Hussein Onn Malaysia, 2023.
[4] N. A. B. M. Radzuan, *Brew Beauty e-Commerce System (Report)*, Batu Pahat, Malaysia: PTTA, Universiti Tun Hussein Onn Malaysia, 2024.
[5] N. D. Shahrul Amer, "Bliese System: A web-based management system for small business," *Advances in Information Technology and Computer Science*, vol. 3, no. 1, p. 71, 2022. [Online]. Available: https://doi.org/10.30880/aitcs.2022.03.01.071
[6] R. S. Pressman and B. R. Maxim, *Software Engineering: A Practitioner's Approach*, 8th ed. New York, NY: McGraw-Hill, 2014.
[7] Guru99. (2024). Prototype Model in Software Engineering. Retrieved from https://www.guru99.com/software-engineering-prototyping-model.html
[8] R. S. Pressman and B. R. Maxim, *Software Engineering: A Practitioner's Approach*, 8th ed. New York, NY: McGraw-Hill, 2014.

Applied Information Technology and Computer Science Vol. 7 No. 1 (2026) p. 842-860

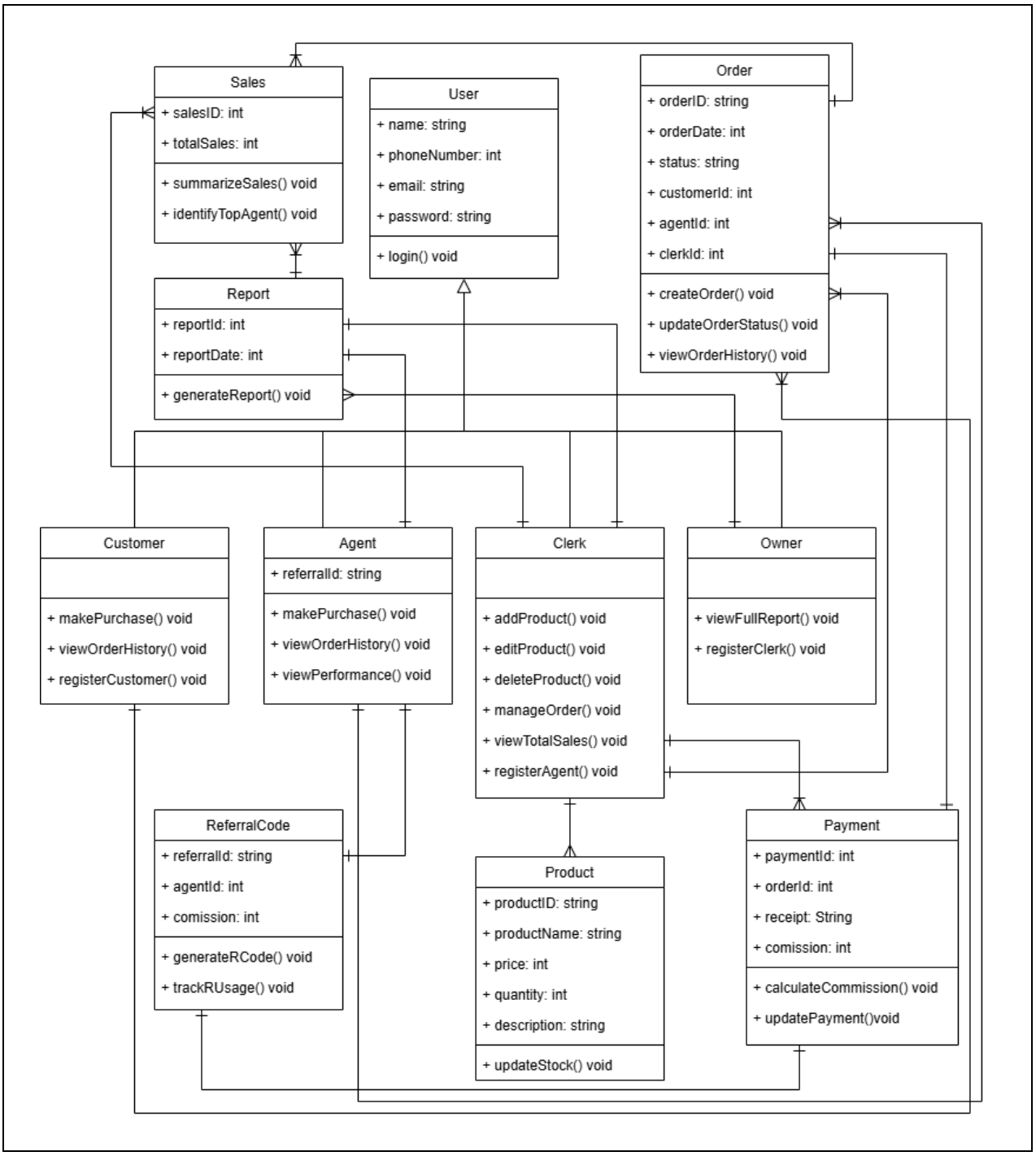
Appendix A

Use Case Diagram



Appendix B

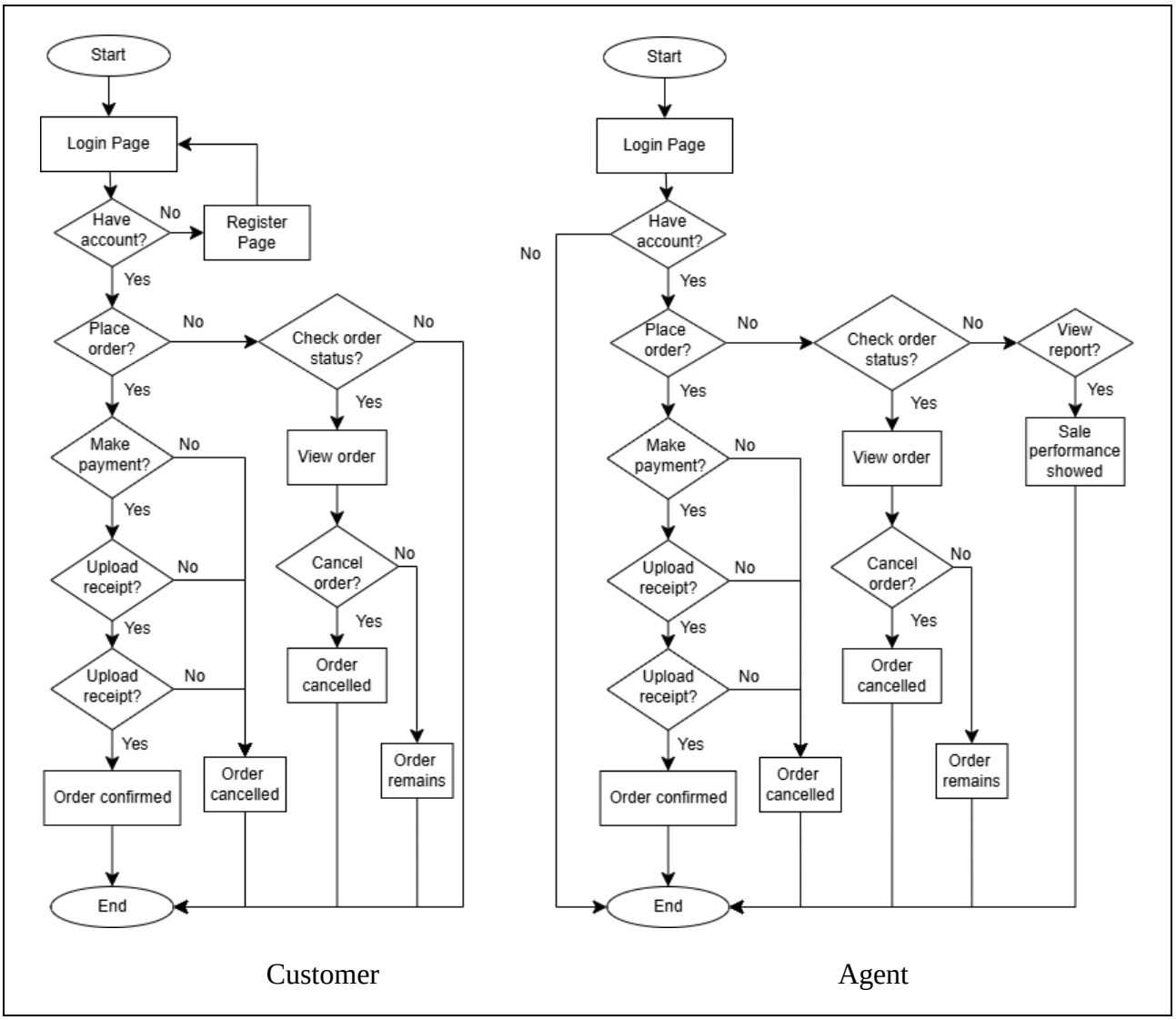
Class Diagram

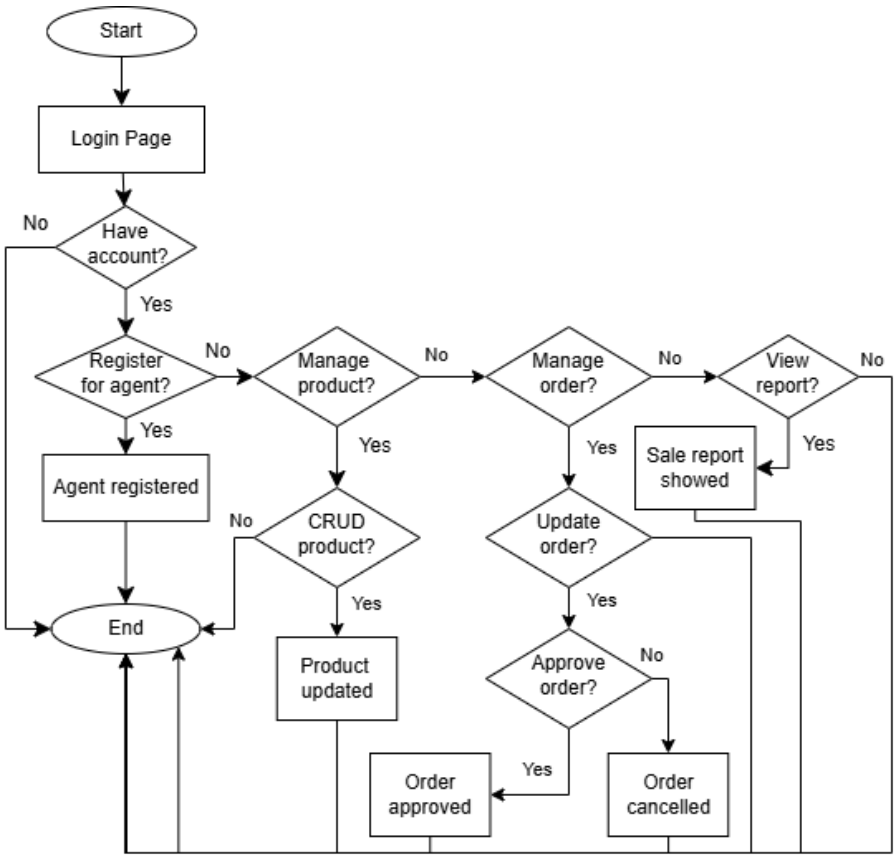


Applied Information Technology and Computer Science Vol. 7 No. 1 (2026) p. 842-860

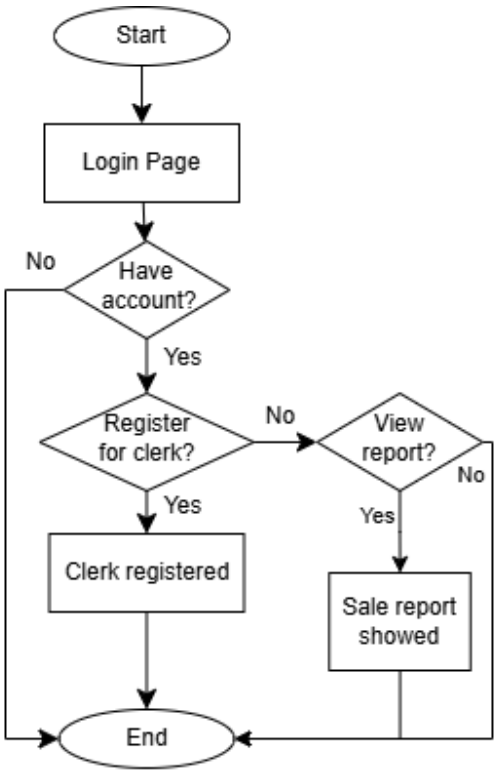
Appendix C

Flowchart




Start
Login Page
Have account?
No
Yes
Register for agent?
No
Yes
Agent registered
Manage product?
No
Yes
CRUD product?
No
Yes
Product updated
Manage order?
No
Yes
Update order?
Yes
Approve order?
No
Yes
Order approved
Order cancelled
View report?
No
Yes
Sale report showed
End

Clerk


Start
Login Page
Have account?
No
Yes
Register for clerk?
No
Yes
Clerk registered
View report?
No
Yes
Sale report showed
End

Owner