



University Distrital
Francisco Jose de Caldas

Technical report on the development of the
Temmu-Based application

Jefferson Rico

Nelson Navarro

Supervisor: Carlos Sierra

A technical report based on the progress and results obtained
from the development of the application

July 08, 2025

Declaration

Jefferson David Rico Ruiz and Nelson de Jesus Navarro de la Rosa of the Francisco José de Caldas District University confirm that this is my own work and figures, tables, equations, code snippets, artworks, and illustrations in this report are original and have not been taken from any other person's work, except where the works of others have been explicitly acknowledged, quoted, and referenced. I understand that if failing to do so will be considered a case of plagiarism. Plagiarism is a form of academic misconduct and will be penalised accordingly. I give consent to a copy of my report being shared with future students as an exemplar. I give consent for my work to be made available more widely to members of UoR and public with interest in teaching, learning and research.

I give consent to a copy of my report being shared with future students as an exemplar.

I give consent for my work to be made available more widely to members of UoR and public with interest in teaching, learning and research.

Jefferson David Rico Ruiz and Nelson de Jesus Navarro de la Rosa
July 08, 2025

Abstract

We created an application to satisfy a user's need: buying and/or selling products. The proposed solution uses notions and paradigms related to object-oriented programming to apply its concepts and achieve a goal: enabling a user to interact with the application and buy and sell products.

Keywords: Application, Development, Design, Oriented Object Programming, Purchase

Acknowledgements

Special thanks to Professor Carlos Sierra for guidance and feedback throughout the project, and to the team members who contributed to design and testing phases.

Contents

List of Figures	v
List of Tables	vi
1 Introduction	1
1.1 Background	2
1.2 Problem statement	2
1.3 Aims and objectives	2
1.4 Solution approach	3
1.4.1 User Stories	3
1.4.2 Chosen Paradigm	3
1.5 Summary of contributions and achievements	3
1.6 Organization of the report	4
2 Literature Review	5
3 Methodology	6
3.1 Methodology	6
4 Results	11
5 Discussion and Analysis	12
5.1 Discussion	12
5.2 Scope	12
5.3 Assumptions	13
5.4 Limitations	13
6 Conclusions and Future Work	14
6.1 Conclusions	14
6.2 Future Work	14
References	15

List of Figures

3.1	User role selection interface	6
3.2	Class diagram of the application	7
3.3	Main Seller interface	8
3.4	Pay Product(Customer)	9
3.5	Database	10

List of Tables

4.1 Functional Testing Summary	11
--	----

Chapter 1

Introduction

The expansion of e-commerce platforms in recent years has transformed how users interact with digital marketplaces. Applications like Temu and Shopee offer seamless user experiences and efficient product navigation, setting a high standard for modern online shopping. In this context, the challenge arises when attempting to replicate such a platform in an academic environment using simplified tools and limited backend capabilities.

This project focuses on the development of a desktop-based marketplace application inspired by the Temu model. Developed entirely in Java using the Eclipse IDE, the application simulates the main functionalities of a real marketplace, such as product browsing, role-based access, publication of items by sellers, and payment simulation by buyers. The system was implemented using the Model-View-Controller (MVC) architectural pattern to ensure modularity, code clarity, and separation of responsibilities. User interface designs were created with Figma, diagrams with Draw.io, and collaborative development was supported by GitHub.

The primary goal of the project is to provide an academic prototype that demonstrates how a simplified version of an online marketplace can be designed and developed while reinforcing core programming concepts, object-oriented design, and interface development. One key feature of the project is its use of plain text files for data persistence, enabling login functionality, product tracking, and purchase simulation without requiring a real database.

Methodologically, the project followed an iterative development approach. Functional and integration tests were performed to ensure that users could log in, publish products (if sellers), browse and simulate purchases (if buyers), and that all data was stored and retrieved correctly from '.txt' files.

The main outcome of this project is a fully functional academic simulation of a marketplace that highlights the power of Java's GUI capabilities combined with good architectural practices. The system allows users to interact through a role-based interface, publish and view products, and simulate payments, all while reinforcing programming skills.

This report is organized as follows: Section 2 details the methodology and tools used during development. Section 3 presents the results and validation of the system. Section 4 discusses the implications and challenges of the implementation. Finally, Section 5 provides conclusions and potential directions for future work.

1.1 Background

The rapid growth of e-commerce platforms over the last decade has reshaped how people interact with online markets. Applications like Temu, Shopee, and AliExpress provide highly dynamic and user-centered shopping experiences that rely on sophisticated systems, scalable architectures, and seamless user interfaces. While these platforms are built using complex technologies such as distributed systems, cloud computing, and large-scale databases, their core principles—user role management, product publication, browsing, and payment simulation—can be abstracted and replicated in a simplified academic environment.

In computer engineering education, replicating simplified versions of real-world systems provides a valuable learning experience. It offers a practical context in which to apply fundamental concepts such as object-oriented programming, software architecture, graphical user interface design, and data persistence. By modeling a basic version of a marketplace platform, students are able to explore and implement core patterns like the Model-View-Controller (MVC) structure, which promotes clean separation of concerns and modular development.

The main motivation behind this project stems from the ongoing digital transformation that is reshaping how individuals engage in commercial activities. As user needs and expectations evolve, it becomes essential to understand and simulate the dynamics of modern platforms. Additionally, this project provides a valuable opportunity to enhance technical skills in design and development by applying engineering knowledge to build a complete, functional application from scratch. The goal is not only to replicate the essential features of a marketplace, but also to reinforce programming concepts through the creation of a structured and purposeful solution.

1.2 Problem statement

Online marketplace platforms have become increasingly complex, integrating advanced technologies to provide seamless user experiences, real-time data synchronization, and scalable backend infrastructures. While these systems are highly effective in real-world applications, their complexity makes it difficult to study their architecture and logic in a controlled, academic setting. Most educational software development environments lack access to the computational and financial resources necessary to replicate the full functionality of platforms like Temu or Shopee.

As a result, students are often limited to studying theoretical models or isolated components, rather than developing complete, working systems. There is a clear need for a simplified yet comprehensive prototype that simulates key functionalities of a marketplace—such as user authentication, role-based access (buyer/seller), product publication, navigation, and simulated payment processes—using accessible tools and technologies.

The problem addressed in this project is the lack of an academic-oriented, fully functional simulation of a marketplace application that allows students to understand core design patterns, user interface development, and data management without relying on complex backend infrastructure. This simulation is intended to bridge the gap between theoretical concepts and practical implementation in the context of computer engineering education.

1.3 Aims and objectives

Aim: The primary aim of this project is to design and implement a functional simulation of an online marketplace platform inspired by Temu, using Java and accessible development tools. The project seeks to provide a practical educational tool for applying and reinforcing computer

engineering concepts, particularly object-oriented programming, system architecture (MVC), and user interface design.

Objectives:

- To analyze and abstract the core functionalities of a real-world marketplace application.
- To design the software architecture using the Model-View-Controller (MVC) pattern for modularity and maintainability.
- To implement essential features such as user authentication, buyer/seller roles, product publishing, browsing, and simulated payment processes.
- To ensure data persistence through structured text files for storing user data, product records, and payment history.
- To validate the application's functionality through unit testing and simulated use cases.

1.4 Solution approach

The design process is based on analyzing functional and non-functional requirements based on previous user stories to ensure that the application effectively meets its objectives. The interface is designed with simplicity, clear information presentation and a visual hierarchy, therefore, taking out of the way unconcern by removing any complexity or inappropriate menus.

1.4.1 User Stories

User stories guide the project's development and influence architectural decisions and the application of Object-Oriented Design (OOD) principles.

1.4.2 Chosen Paradigm

Taking in care the best option, we considered object-oriented programming, this consists in identifying and organizing the objects of our program. With this the solution will be the interaction between the objects. This paradigm allows us to have good flexibility because we can make objects that are independent from the others, allowing to the modification and scalability of the program.

1.5 Summary of contributions and achievements

This project resulted in the development of a functional simulation of an e-commerce marketplace inspired by Temu. Key contributions include:

- A role-based system that allows users to register and log in as either buyers or sellers.
- A seller interface where users can publish products, view their listings, and track sold items.
- A buyer interface that enables product browsing, category filtering, and purchase simulation.

- A lightweight data persistence mechanism using plain text files to store users, products, and payment records.
- The application of CRC cards and user stories for structured design and requirement gathering.

The project demonstrates the practical application of object-oriented programming principles, graphical user interface (GUI) design, and system modeling. It also proves the viability of building educational simulations of complex systems using accessible tools, which is particularly valuable in the context of computer engineering education.

1.6 Organization of the report

This report is organized into seven chapters. Chapter 1 provides the general introduction to the project, including the context, motivation, objectives, problem statement, and the adopted solution approach. Chapter 2 presents the literature review, highlighting previous work, related models, and relevant theoretical foundations.

Chapter 3 describes the methodology applied throughout the development of the system, including the design process, tools used, software architecture, and implementation strategy. Chapter 4 outlines the main results obtained during the development, including the core features implemented.

Chapter 5 analyzes and interprets the results, discussing their implications, strengths, and limitations. Chapter 6 summarizes the conclusions drawn from the project and proposes future improvements or directions for further work. Finally, Chapter ?? provides a personal reflection on the overall development process, lessons learned, and challenges faced during the project.

Chapter 2

Literature Review

The rapid growth of e-commerce in recent decades has transformed the way people buy and sell products. Digital platforms such as Amazon, eBay, and Shopee have redefined consumer behavior and set new standards for user experience, convenience, and market reach. Temu, launched by PDD Holdings, introduced a competitive marketplace model with aggressive pricing, social integration features, and a mobile-first design, which has drawn increasing attention from both users and developers.

Several studies have analyzed the impact of marketplace applications on consumer choice and seller dynamics. Research by Ghosh et al. (2021) discusses the importance of recommendation systems and intuitive UI/UX design in influencing purchasing behavior. In addition, Xia and Zhang (2022) explore how mobile commerce apps can enhance user retention through features like personalized interfaces and simplified checkout processes.

From a technical perspective, developing such applications involves challenges in software architecture, role management, and data handling. Java remains a widely adopted programming language for educational projects due to its object-oriented nature, strong typing, and rich ecosystem of development tools. According to Horstmann (2020), Java provides an ideal environment for introducing design patterns and GUI-based applications in academic settings. The Eclipse IDE, in particular, offers students a reliable platform to write, debug, and manage code efficiently.

For data persistence, previous academic works often leverage flat-file systems or lightweight databases such as SQLite to simulate backend functionality without complex server infrastructure. This approach aligns with the educational purpose of this project, which avoids real-time databases or cloud storage in favor of local '.txt' files.

Overall, this literature provides the theoretical and practical foundation upon which this project is built. By simulating a simplified version of a well-known marketplace app, students can explore software development lifecycles, user interface design, and basic data management strategies.

Chapter 3

Methodology

3.1 Methodology

The application was developed incrementally using object-oriented principles. Java was selected due to its robustness and alignment with academic programs. The Eclipse IDE was used alongside tools like Figma (UI design), Draw.io (UML), and GitHub (version control).

3.1. Requirements Analysis

Functional and non-functional requirements were defined. Two roles were established: customer and seller. Use case diagrams and high-level architecture were drafted.

3.2. System Design

A modular architecture was chosen, separating concerns into packages for GUI, logic, models, and controllers. Class and navigation diagrams were developed to ensure scalability.

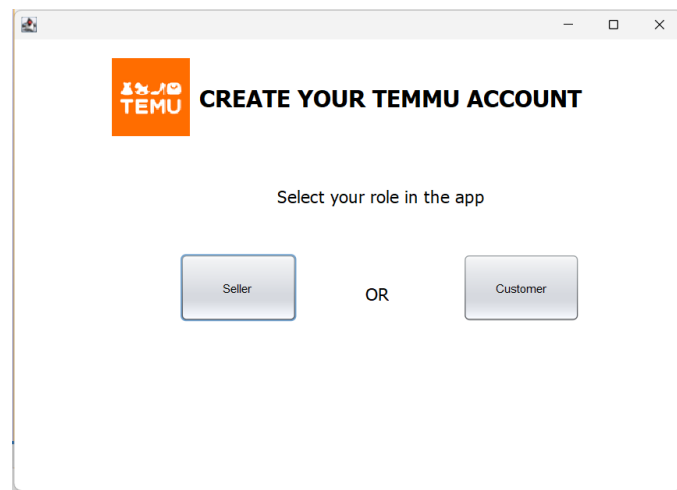


Figure 3.1: User role selection interface

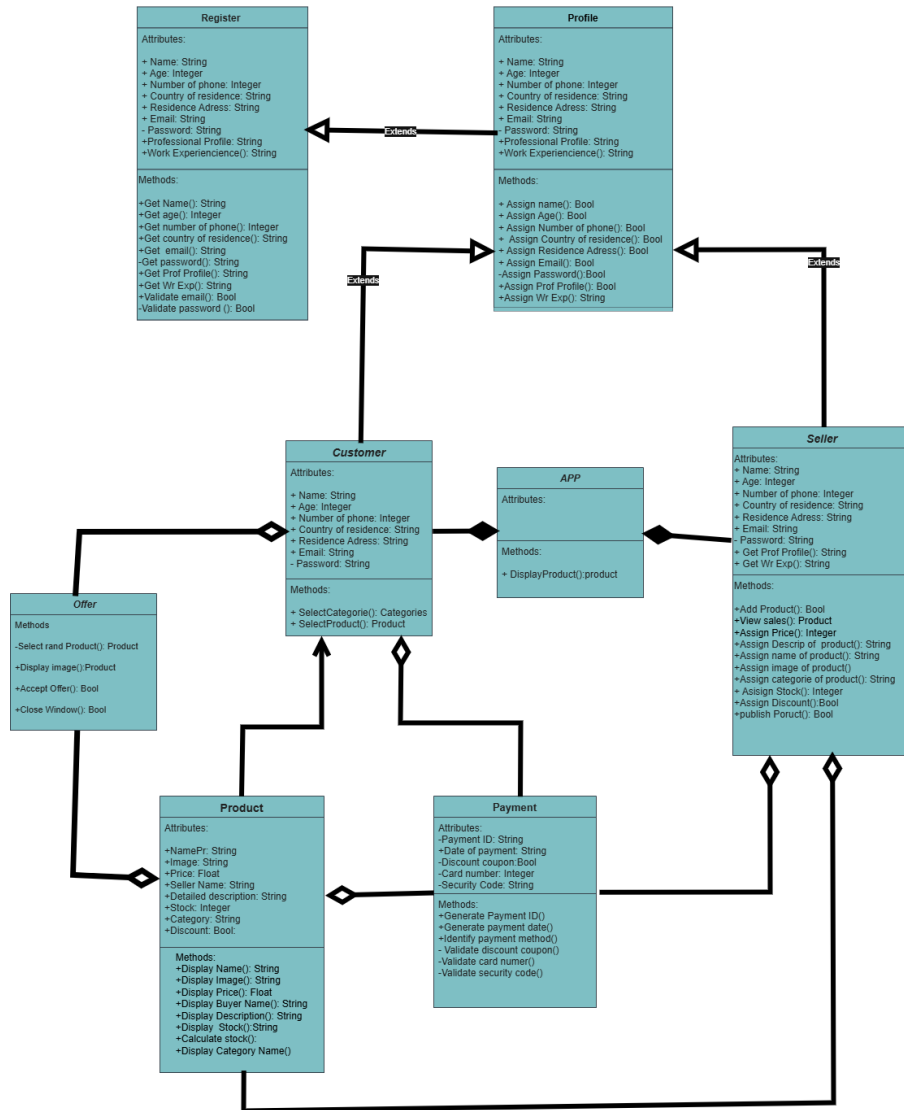


Figure 3.2: Class diagram of the application

3.3. Implementation

Java Swing was used to build interfaces. Role-specific panels were created. Seller can publish and view products. customer can browse, filter, and simulate purchases. OOP principles were consistently applied.

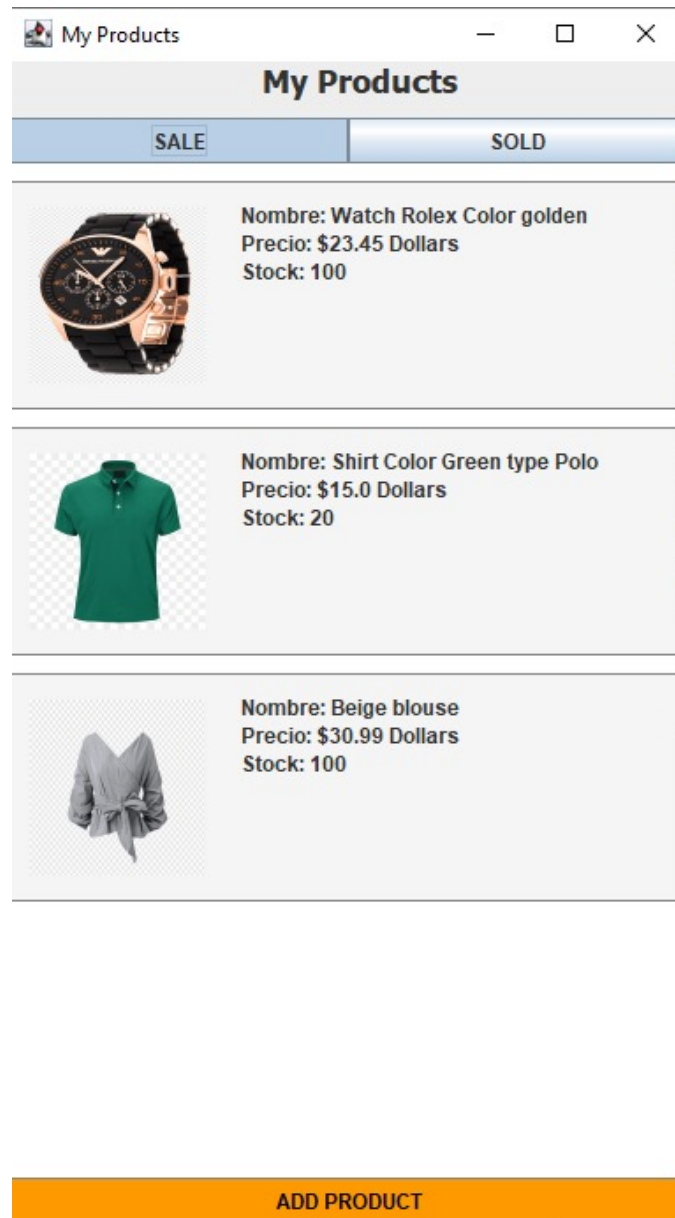


Figure 3.3: Main Seller interface

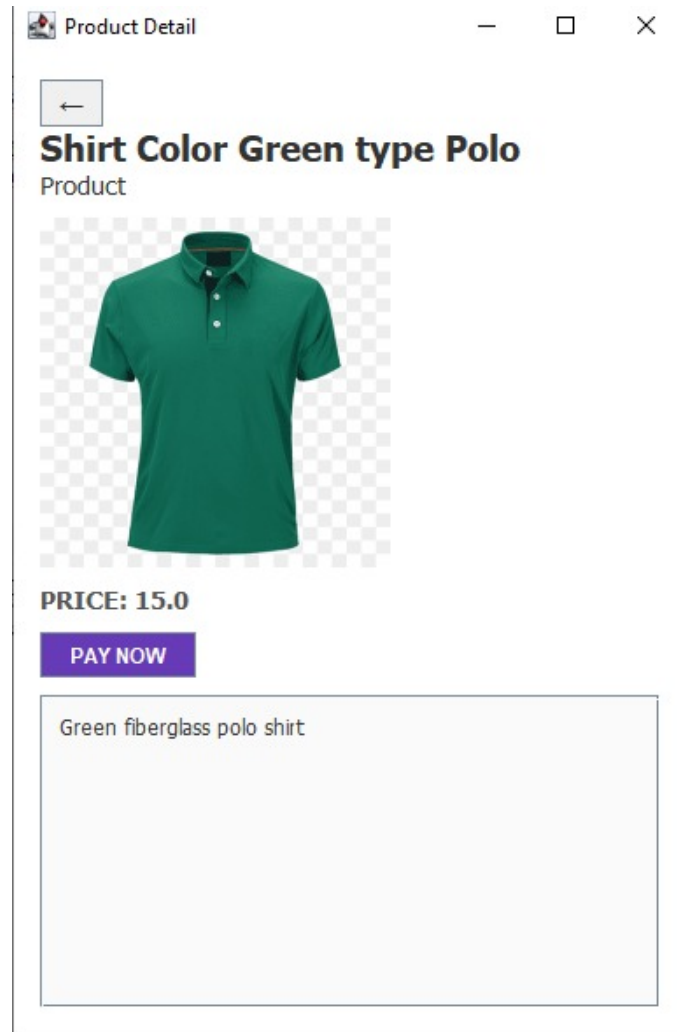


Figure 3.4: Pay Product(Customer)

3.4. Testing

Manual testing was performed to validate panel navigation, button logic, cart behavior, and simulated purchases. While automated testing was not implemented, all test cases were documented.

3.5. Data Persistence with Text Files

To support simple persistence, the application reads from and writes to structured '.txt' files. These files store the data of registered users (used for login), products published by sellers, records of sold products, and simulated payments. Java's standard file I/O operations are used, enabling basic storage without database integration. This method is ideal for educational purposes and ensures that information is retained between application sessions.

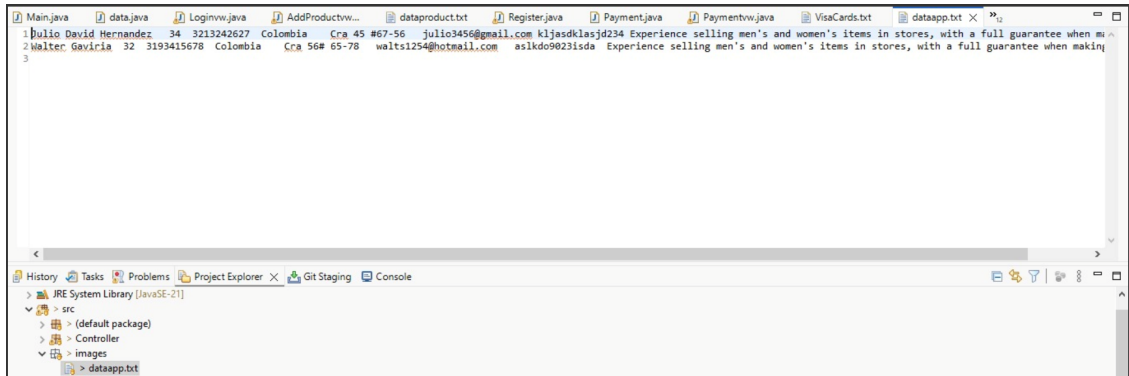


Figure 3.5: Database

Chapter 4

Results

The prototype meets all defined objectives. Users can select their role and access corresponding interfaces:

- Seller: browse products, use filters, view details, simulate purchases.
- Customer: publish products, view listings, and track sold items.
- The application successfully stores and retrieves user login data, published products, sales records, and payment simulations through plain text files. This implementation demonstrates a basic persistence strategy without requiring external database systems.

Table 4.1: Functional Testing Summary

Functionality	Input	Expected Result	Observed Result
Role selection	Click "Seller" or "Customer"	Access corresponding panel	Success
select product to buy	Product selected by Customer	Redirect to the selected product panel	Success
Simulate purchase	Confirm purchase	update sales	Success
View sold products	Seller with sales	Sold items listed	Success
Category filter	Select category	Display matching products	Success

Chapter 5

Discussion and Analysis

5.1 Discussion

The results show that it is possible to simulate a marketplace flow using Java and Eclipse. Modular design facilitated clear role separation and future extensibility. The inclusion of product publishing, filtering, and purchase simulation closely mimics the real behavior of the marketplace.

However, some limitations such as lack of persistence, authentication, and automated tests restrict the project to academic purposes. Still, it offers a strong foundation for more advanced implementations in future work.

5.2 Scope

This project aims to simulate a simplified version of a marketplace application inspired by Temu. The scope includes user authentication through a login system, role-based interfaces for buyers and sellers, product publication by sellers, display of sold products, browsing available items, and a simulated payment process.

The application also implements basic data persistence using '.txt' files to store user credentials, published products, sales history, and payment records. While it does not integrate with real-time databases or external payment gateways, the system effectively simulates core marketplace interactions in an academic setting.

Features included:

- GUI developed in Java Swing.
- Role selection at login.
- Product browsing with category filters.
- Purchase simulation.
- Seller view: published products and sold items.
- Product publishing by seller.

5.3 Assumptions

- The application is run in a local environment with a single user session at a time.
- Users understand their selected role and use the interface properly.
- Java Swing is supported by the host machine.

5.4 Limitations

- The application does not support real-time concurrent access.
- Payment simulation does not involve real money or actual financial APIs.
- Data stored in '.txt' files is not encrypted or secured.
- No integration with external APIs.
- UI optimized only for desktop.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

This project successfully developed a simulated marketplace application inspired by the Temu model, using Java and Swing for the interface, and implementing key functionalities for both buyers and sellers. The modular design allowed the creation of user-specific panels and logic while maintaining clean code structure and separation of concerns.

A major enhancement was the integration of simple data persistence using '.txt' files, enabling the storage and retrieval of essential data such as user credentials, product listings, sales history, and payment records. This feature strengthens the application's academic value and demonstrates a foundational approach to persistent storage without requiring a database.

Overall, the application meets the defined goals by replicating a functional marketplace environment in a controlled academic context, providing a useful platform to apply software engineering concepts and object-oriented programming principles.

6.2 Future Work

While the current application meets the essential functionalities—such as user registration, login, product publishing, and simulated payments—there is room for further improvement.

In future iterations, data persistence can be enhanced by replacing .txt files with a relational database to ensure better structure and scalability. Security can also be improved by encrypting user credentials and strengthening authentication processes.

Additionally, the user interface could be modernized using JavaFX or web technologies to provide a more responsive and accessible experience. Real-time features, such as payment processing or order tracking, may also be integrated in later stages to simulate a more complete marketplace environment.

References

- [1] Laudon, K. C., & Traver, C. G. (2020). *E-commerce 2020: Business, technology, society*. Pearson.
- [2] Temu Official Website. Retrieved from: <https://www temu.com>
- [3] Oracle. (2024). *The Java™Tutorials*. Retrieved from: <https://docs.oracle.com/javase/tutorial/>
- [4] Ghosh, A., Liu, Y., & Chen, S. (2021). *Designing Intelligent Recommendation Systems for E-Commerce Platforms*. *Journal of Digital Commerce*, 15(2), 45–60.
- [5] Xia, M., & Zhang, Y. (2022). *User Retention in Mobile Commerce: The Role of Personalization*. *International Journal of Mobile Applications*, 9(3), 78–92.
- [6] Horstmann, C. S. (2020). *Core Java Volume I–Fundamentals*. Pearson Education.