PHARMACY PRESCRIPTION TRANSACTION MANAGER

Proponents: Makigod, Trisha Eunice C. , Quijada Shenna , Dotillos Hannah

Course / Instructor: CCE 105L / Althea Owe

Date Submitted: 10/09/25

Abstract

The project is a pharmacy management system database designed to address the problem of organizing and managing pharmacy operations efficiently. The objective is to create a structured database that handles medicines inventory, patient records, prescriptions, payments, and transactions. The method involves designing relational tables such as medicines, patients, prescriptions, prescription items, payments, transactions, and users, with appropriate relationships and data constraints. The system tracks medicine stock status, patient prescription history, payment details, and transactional records. The outcome is a comprehensive, normalized database schema supporting data integrity and facilitating smooth pharmacy workflow management, including inventory control, prescription processing, and payment tracking. This enables pharmacists and administrators to manage resources and services effectively in a digital environment.

INTRODUCTION

Background

In today's fast-paced world, applying technology to everyday tasks is very essential for efficiency and accuracy. This proposal presents the application development of Pharmacy Prescription Transaction Manager featuring a graphical user interface (GUI) to operate simple pharmacy operations. By utilizing this modern technology, this system will reduce manual paperwork and it speeds up the service. It also emphasizes future growth by designing a friendly and easy to use and secure interface system that is adaptable. It is vital for ensuring timely, reliable and safe services, while also supporting pharmacy staff in managing workload effectively.

The project was developed to modernize and digitize manual pharmacy operations, which are often prone to errors, inefficiencies, and difficulty managing large volumes of data related to medicines, patients, prescriptions, and payments. The goal was to create a centralized, computerized system that streamlines pharmacy workflows, improves inventory and prescription management, and enhances overall service quality to meet growing demands in healthcare delivery.

1.1 Problem Statement

The primary issue addressed is the challenge faced by pharmacies in efficiently managing their operations manually or using fragmented systems. Problems include inventory mismanagement, difficulty tracking prescriptions and patient records, delayed payment processing, and lack of real-time data for decision-making. These operational inefficiencies can lead to medicine shortages, errors, and poor patient service.

1.2 Objectives

To develop a reliable pharmacy management system that automates inventory, prescription, payment, and patient record management.

1.2 Scope and Limitations

The system covers managing medicine stock, patient information, prescriptions, payments, and sales within the pharmacy to help improve daily operations. Hence, it excludes connecting to outside health or insurance systems, checking for drug interactions and depends on having good computer equipment and internet to work well.

Methodology

In developing our program, we used the Waterfall methodology. Waterfall methodology is a way to build software (or do a project) by following a set of steps, one after another, in a fixed order. You finish one step completely before moving to the next. Imagine water flowing down steps in a waterfall once it goes down, it doesn't go back up. This means that in our project, each phase (such as planning, design, coding, testing, and deployment) was done one at a time. Only after we finished and reviewed one step did we move on to the next, which helped us stay organized and focused during development. This approach also made it easier for us to track our progress and ensure that every part of the project met our requirements before moving forward.

- **Requirements**

  We defined clear features and modules up-front: login, patient management, admin, transactions, prescription, payments.

- System and Software Design

  We designed database schemas first, and worked on class diagrams (UML) and GUI layouts before coding the logic.

- **Implementation (Coding)**

  We developed each module (Login, Admin Panel, Payment, Prescription etc.) sequentially, testing one panel/class at a time. Coding focused on one clear function before moving to the next module.

- **Integration and Testing**

  We connected Java Swing GUIs to MySQL with JDBC after building and testing GUI elements. Testing is performed as each panel or feature is implemented.

- **Deployment**

  The program is used on a local machine, with transfer to new devices via file copy or Google Drive/Github.

- **Maintenance**

  We debug, refactor, and modify code as you discover issues during usage or integration.

Other factors that we used

- **Prototyping**

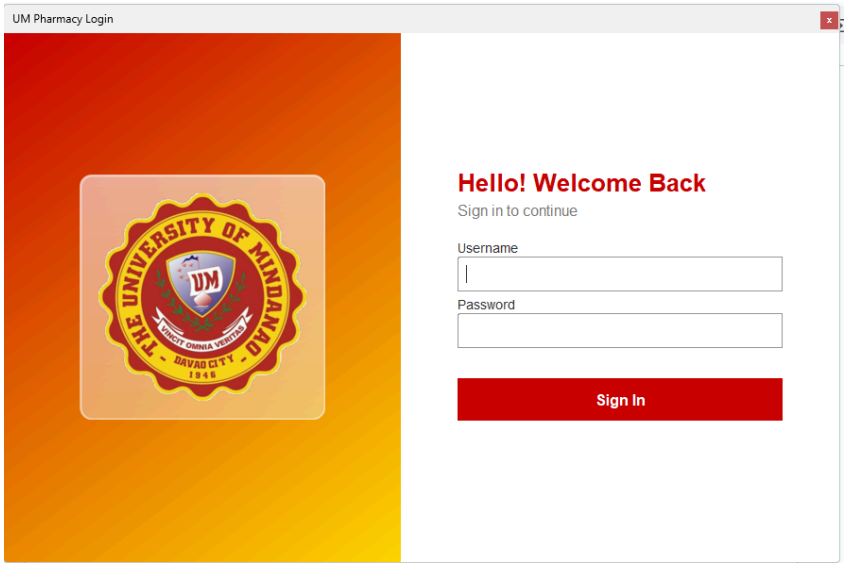
  Each GUI panel/class was prototyped and repeatedly improved

- **Iterative refinement**

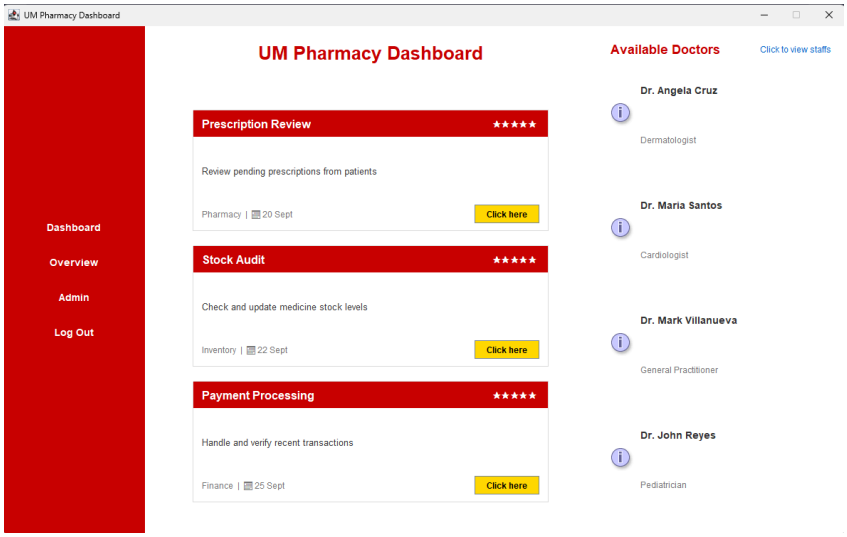
  We made several changes and corrections to individual classes and workflows over time.

System Design

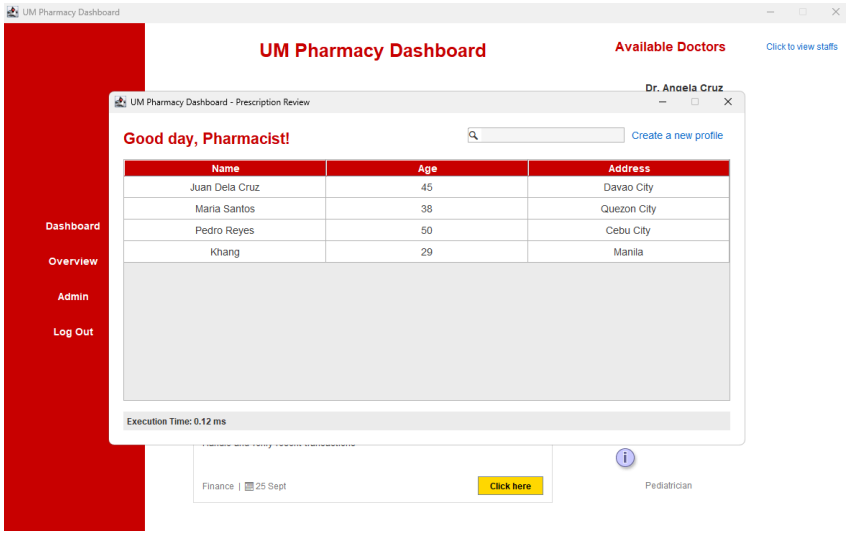
1.1 Login Frame



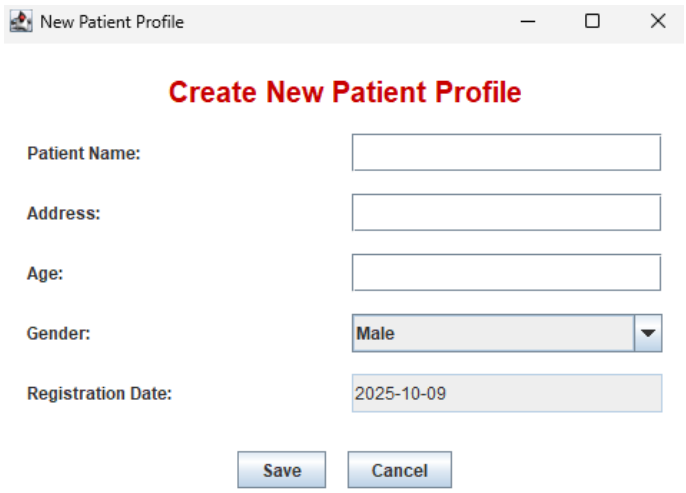
1.2 Dashboard Frame



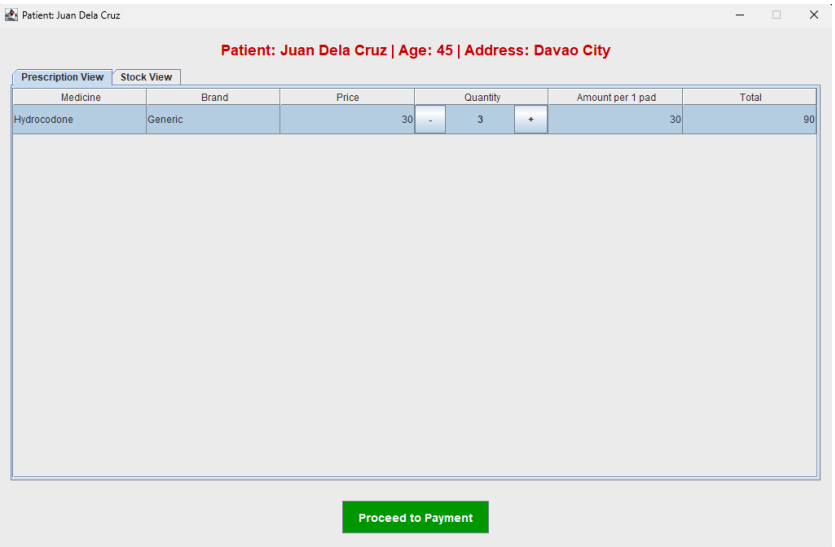
1.3 Prescription Review Frame



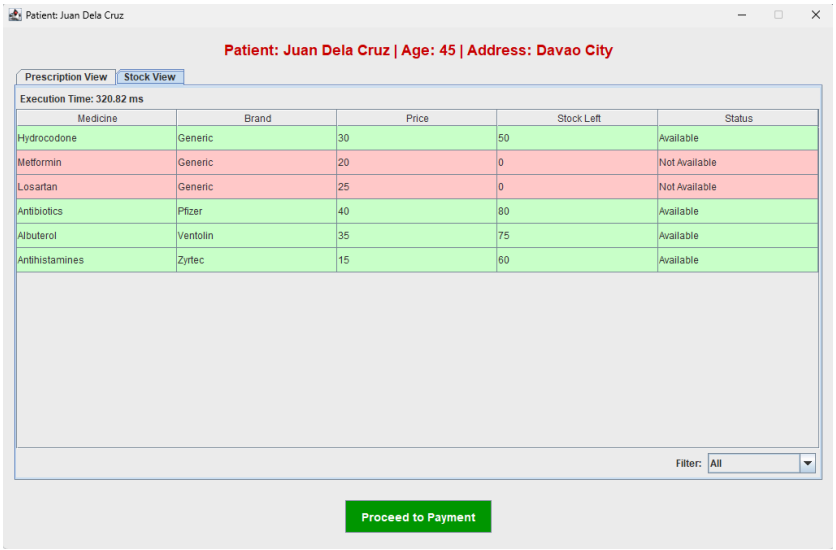
1.4 Create Patient Information Frame



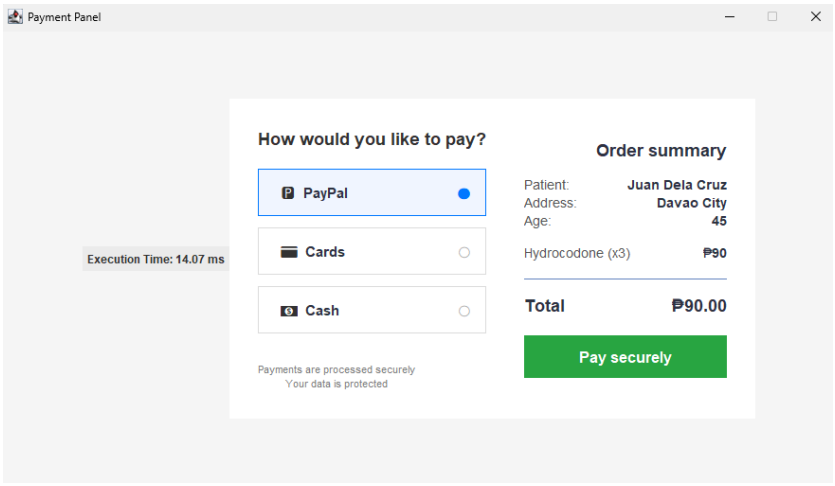
1.5 Prescription View Tab



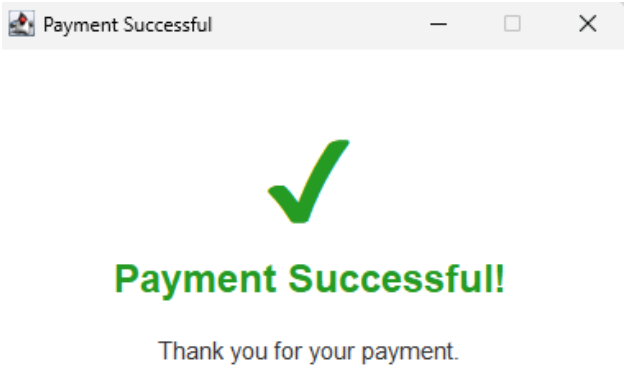
1.6 Stock View Tab



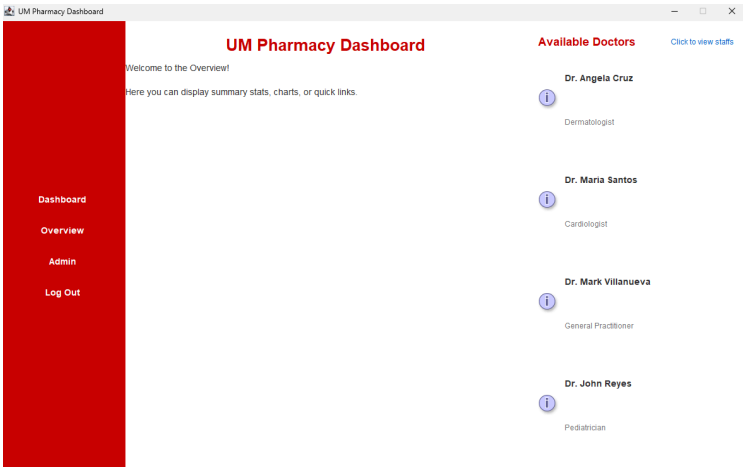
1.7 Payment Frame



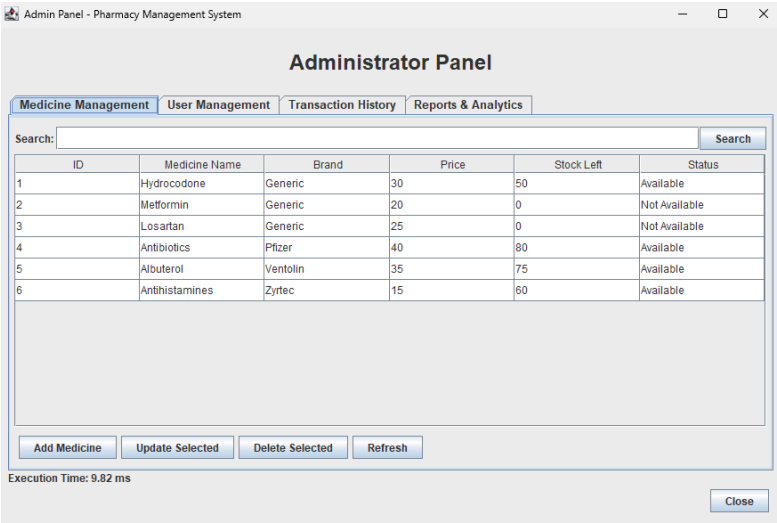
1.8 Payment Success Frame



1.9 Overview Frame



2.0 Admin Frame



Results and Discussion

After following the Waterfall methodology for our software project, we were able to build a pharmacy management system with key features such as user authentication, medicine and patient management, prescription handling, transaction recording, and payment processing. Using Java Swing for the user interface and MySQL for the database allowed us to create a system that is easy to use and helps organize pharmacy operations more efficiently.

Results:

- The program successfully allowed users to log in (with hardcoded sample credentials), add, search, and update medicine information, manage patient and prescription details, process payments, and maintain a record of transactions.
- The database integration using JDBC worked as intended, enabling persistent storage and retrieval of pharmacy records.

- Various GUIs (Admin Panel, Payment Frame, Patient Details, etc.) were created for different modules, each interfacing with the database as required.
- Features like a search bar, filters, and prescription quantity controls enhanced usability.

Discussion:

- The system improves traditional pharmacy work by reducing errors (e.g., by tracking available stock and patient histories), speeding up transaction recording, and simplifying report generation.
- One limitation we observed is that user authentication is currently based on hardcoded usernames and passwords, not live database records. This means new users need to be programmed in, and security is basic.
- The system depends on the local setup of MySQL and proper configuration of JDBC; if these are incorrect, the program will not run.
- While most core features are covered, some advanced needs (like user role permissions, real-time concurrent access, or online backup) are not included.
- User interface adjustments (panel resizing, error dialogs) and method timing feedback were incorporated in later stages, demonstrating iterative improvement even within a mostly Waterfall approach.

Conclusion and Recommendations

Our Pharmacy Prescription Transaction Manager successfully integrates a Java Swing user interface with a MySQL backend, providing pharmacy users with streamlined management of medicines, patients, transactions, and payments. By following the Waterfall methodology, we methodically planned, designed, implemented, and tested each feature, resulting in a functional and organized system. The program helps reduce errors, speeds up recording and searching of

pharmacy data, and offers a user-friendly experience. While core goals were achieved, our project also revealed some limitations, such as the lack of database-driven user authentication, limited security, minimal input validation, and no role-based permissions.

To further improve the system, we recommend:

- Implementing secure user authentication that checks login credentials directly against the database, rather than using hardcoded sample users.
- Adding user roles and permissions (e.g., admin, pharmacist, cashier) to control access to sensitive features.
- Improving input validation and error handling to prevent invalid entries and provide helpful feedback.
- Strengthening data security and privacy, including password hashing and secure database connections.
- Enabling scalable features such as backup and restore, report export, and support for remote/cloud deployment.
- Enhancing the user interface for better responsiveness and accessibility on different screen sizes.

Study of Algorithm Origin Used

Rule-Based Algorithm

A rule-based algorithm uses a clear set of "if this happens, then do that" rules created by experts to automate decisions. It processes input by checking conditions and performing actions based on fixed, human-coded logic. Originating from AI expert systems, it is widely used in pharmacy for

automating tasks like medicine distribution and inventory management because it is transparent, flexible, and reliable.

Mufadhol M., Mustafid M., Jie F., Hidayah Y. N. (2023). The new model for medicine distribution by combining supply chain and expert system using rule-based reasoning method. *International Journal of Artificial Intelligence*, 12(1), 295-304. https://doi.org/10.11591/ijai.v12.i1.pp295-304

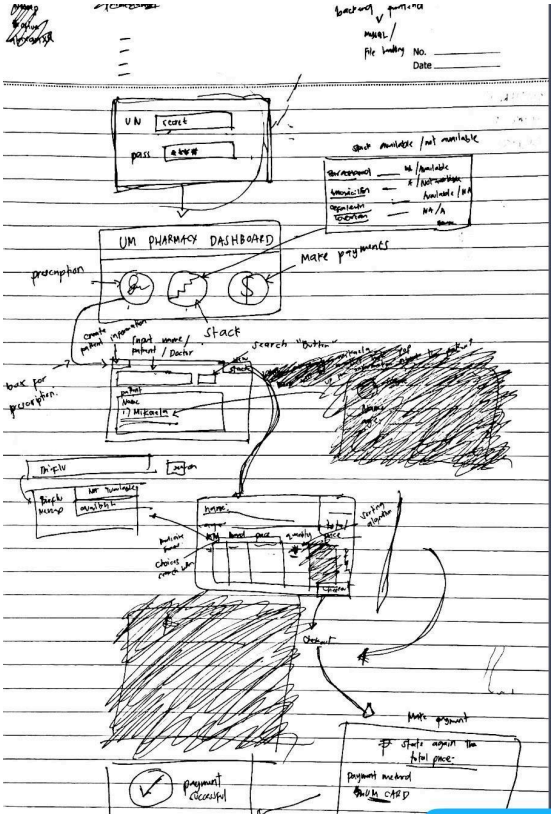
1.2 Algorithm Selection

In developing our Pharmacy Prescription Transaction Manager, finding an algorithm fit for our code will help the system manage prescriptions and transactions efficiently. As for this project, the Searching Algorithm was chosen.

The searching algorithm was chosen because it will help our pharmacy system find the prescriptions, medicines, the patient's name and background, and even the past transactions quickly. This searching algorithm allows the pharmacist to serve customers faster, more accurate, and easier to use.

To improve the searching algorithm, It is by making it work even with a large number of records as the pharmacy grows. The researcher's plan is to have another improvement. It is to add features that suggest the correct name of the medicine when spelling is wrong, In order not to give a wrong medicine and to avoid such accidents. These improvements will make our system more accurate, reliable, and even user-friendly.

Wireframe System



Appendices

https://github.com/yurachi-prog/Pharmacy-Prescription-Transaction-Manager.git

References

- Desai, V., Basargekar, V., Thorbole, S., Uttekar, S., Rai, S., Shingade, P., & Dhamale, Y. (2024). Pharmacy Management System. International Journal of Scientific Research & Engineering Trends, 10(2).
  URL:

  https://ijsret.com/wp-content/uploads/2024/03/IJSRET_V10_issue2_229.pdf

- Pharmacy Management System. (2025). International Journal for Research and Applications in Science, Engineering & Technology.
  URL: https://www.ijraset.com/research-paper/pharmacy-management-system

- Survey on Pharmacy Management System. (2024). Journal of Pharmacy Technology and Clinical Practice.
  URL: https://jptcp.com/index.php/jptcp/article/view/10060

- Osplabs. (2025). Pharmacy Management System Guide.
  URL: https://www.osplabs.com/pharmacy-management-system/

- AltexSoft. (2021). Pharmacy Management System: Benefits, Features, and Future Trends.
  URL: https://www.altexsoft.com/blog/pharmacy-management-system/