**PROJECT PROPOSAL**

Supervised Independent Study

(Capstone Project)

MIT 593-5

**Pharmacy Management System**

Student Registration No – UWU/PGU/MIT/23/112

Master of Information Technology

Postgraduate Unit / Board of Study-Computing and Informatics

Uva Wellassa University of Sri Lanka

1st Intake 2023/2024

## DECLARATION

I hereby declare that this project will be developed by us and will be my original work. Any sources used will be properly cited. This project is being conducted under the supervision of Dr. H.M.S.N. Ariyadasa, Senior Lecturer, Department of Computer Science and Informatics, Uva Wellassa University. This Project Proposal is submitted in partial fulfillment of the requirements for the course unit MIT 259, Supervised Independent Study (Capstone Project) for the Master of Information Technology (MIT) program.

Student Name – N.W.K.K Indika

Student Registration No – UWU/PGU/MIT/23/112

Supervisor Details:

| Name of the Supervisor | E-mail | Contact Number |
| --- | --- | --- |
| **Dr. H.M.S.N. Ariyadasa** | **subhash@uwu.ac.lk** | **+94 77 280 7582** |

Project Approval:

Signature of the Supervisor

Name of the Supervisor

Date

Contents

[DECLARATION 2](#_heading=h.1fob9te)

[Glossary of Terms 4](#_heading=h.3znysh7)

[1.](#_heading=h.2et92p0) Introduction 5

[1.1.](#_heading=h.tyjcwt) Project Title 5

[1.2.](#_heading=h.3dy6vkm) Project Description 5

[1.3.](#_heading=h.1t3h5sf) Background 5

[1.4.](#_heading=h.4d34og8) Motivation 5

[1.5.](#_heading=h.2s8eyo1) Problem in Brief 5

[1.6.](#_heading=h.17dp8vu) Proposed Solution 6

[1.7.](#_heading=h.3rdcrjn) Project Aim and Objectives 6

[2.](#_heading=h.26in1rg) Methodology 7

[2.1.](#_heading=h.lnxbz9) Requirement Gathering and Analysis 7

[2.2.](#_heading=h.35nkun2) System Design 7

[2.3.](#_heading=h.1ksv4uv) Implementation 8

[2.4.](#_heading=h.44sinio) Testing 9

[2.5.](#_heading=h.2jxsxqh) Deployment 10

[2.6.](#_heading=h.z337ya) Maintenance 10

[3.](#_heading=h.3j2qqm3) Requirements Identification 12

[3.1.](#_heading=h.1y810tw) Functional and Non-functional requirements 12

[3.1.1.](#_heading=h.4i7ojhp) Functional Requirements 12

[3.1.2.](#_heading=h.2xcytpi) Non-Functional Requirements 13

[3.2.](#_heading=h.1ci93xb) System Requirements 13

[3.2.1.](#_heading=h.3whwml4) Hardware Requirements 13

[3.2.2.](#_heading=h.2bn6wsx) Software Requirements 13

[3.3.](#_heading=h.qsh70q) User Roles 13

[3.4.](#_heading=h.3as4poj) User Levels 14

[4.](#_heading=h.1pxezwc) Project Plan 16

[5.](#_heading=h.49x2ik5) References 17

[6.](#_heading=h.2p2csry) Appendixes 18

## Glossary of Terms

Abbreviation – Meaning

PMS – Pharmacy Management System

MIT – Master of Information Technology

POS – Point of Sale

Rx – Prescription

## Introduction

### Project Title

Pharmacy Management System

### Project Description

The Pharmacy Management System (PMS) is designed to streamline pharmacy operations, focusing on efficient inventory management, prescription tracking, and seamless sales processes. By securely storing prescription data and ensuring the confidentiality and integrity of patient information, the system contributes to patient-centric care. PMS facilitates smooth transactions for customers and maintains an accessible prescription history, emphasizing reliability, availability, robustness, scalability, usability, and efficiency.

### Background

The pharmacy industry faces numerous challenges, including managing inventory, tracking prescriptions, and handling sales efficiently. Traditional methods are often manual, time-consuming, and prone to errors. This project aims to address these issues by providing a comprehensive software solution.

### Motivation

My motivation for developing a PMS from a desire to streamline pharmacy operations, enhance prescription accuracy, improve inventory management, and elevate patient service. By leveraging modern software development techniques, this system aims to automate and optimize pharmacy workflows, ensuring a higher standard of service and operational efficiency.

### Problem in Brief

Pharmacies often struggle with managing inventory accurately, tracking prescriptions effectively, and processing sales efficiently. Manual methods can lead to errors, stockouts, overstocks, and a lack of timely information, affecting overall operational efficiency and customer satisfaction.

### Proposed Solution

The Pharmacy Management System will automate and streamline pharmacy operations. Key functionalities include efficient inventory management, organized prescription tracking, and seamless sales processes. The system will also focus on data security, ensuring the confidentiality and integrity of patient information.

### Project Aim and Objectives

The primary aim of this project is to develop a robust and efficient Pharmacy Management System (PMS) backend utilizing a micro-service architecture.

This system will ensure high availability and scalability, accommodating the dynamic needs of modern pharmacies. By implementing a micro-service platform, we aim to enhance the system’s modularity, allowing independent development, deployment, and scaling of various services such as inventory management, prescription tracking, and sales management. This approach not only improves system performance and reliability but also facilitates easier maintenance and integration of new features in the future.

1. Provide a comprehensive solution for managing pharmacy operations.
2. Automate inventory management to reduce errors and enhance efficiency.
3. Implement a robust prescription tracking system to ensure patient-centric care.
4. Facilitate smooth sales processes to improve customer experience.
5. Ensure data security and integrity.
6. Develop the system using modern software engineering practices.

## Methodology

PMS will be developed using the waterfall methodology, which is suitable for the proposed projects with clear and well-defined requirements. This methodology follows a sequential design process, making it easier to manage and execute each project phase methodically. The following sections detail the steps involved in the project implementation.

### Requirement Gathering and Analysis

**Objective**

Understand and document the functional and non-functional requirements of the PMS.

**Steps**

1. Stakeholder Interviews – Conduct interviews with pharmacists, store managers, and other stakeholders to gather detailed requirements.
2. Surveys and Questionnaires – Distribute surveys to collect input on current issues and desired features.
3. Document Review – Analyze existing documentation and reports related to the current pharmacy processes.
4. Requirement Specification – Create a detailed Software Requirement Specification (SRS) document.

**Tools**

1. Google Forms (for surveys),
2. Microsoft Word (for documentation and requirement tracking)

### System Design

**Objective**

Design the architecture and components of the PMS based on the gathered requirements.

**Steps**

1. High-Level Design – Define the overall system architecture, including modules for inventory management, prescription tracking, and sales management.
2. Low-Level Design – Detail the design of each module, including database schema, user interfaces, and system interfaces.
3. Design Review – Conduct reviews with stakeholders to validate the design.

**Tools**

1. Microsoft Visio (for creating design diagrams)
2. MySQL Workbench (for database design)

### Implementation

**Objective**

Develop the PMS based on the approved design.

**Steps**

1. Coding Standards – Establish coding conventions to ensure consistency and readability (e.g., Java coding standards, JavaScript conventions).
2. Module Development – Implement each module (inventory management, prescription tracking, sales management) using Java for backend logic and React/JavaScript for the frontend.
3. Version Control – Use GitHub for version control to manage code changes and collaboration.

**Tools**

1. IntelliJ IDEA (for Java development)
2. Visual Studio Code (for React/JavaScript development)
3. GitHub (for version control).

### Testing

**Objective**

Ensure the PMS is functional, reliable, and meets the specified requirements.

**Steps**

1. Unit Testing – Write and execute unit tests for individual components using JUnit (for Java) and Jest (for JavaScript).
2. Integration Testing – Test the interactions between different modules to ensure they work together seamlessly.
3. System Testing – Perform end-to-end testing of the entire system to verify it meets all functional and non-functional requirements.
4. User Acceptance Testing (UAT) – Conduct testing sessions with end-users to validate the system in real-world scenarios.

**Tools**

1. JUnit (for Java unit testing)
2. Jest (for JavaScript unit testing)
3. Postman (for API testing).

### Deployment

**Objective**

Deploy the PMS to the production environment

**Steps**

1. Deployment Preparation – Prepare the deployment environment, including server setup and configuration.
2. Database – Implement database
3. Application Deployment – Deploy the application.
4. Verification – Conduct post-deployment verification to ensure the system is working correctly.

**Tools**

1. Docker (for containerization)
2. MySQL (for database management)

### Maintenance

**Objective**

Ensure the PMS remains functional, up-to-date, and meets evolving user needs.

**Steps**

1. Bug Fixes – Address any bugs or issues reported by users or identified through monitoring.
2. Updates – Implement regular updates to improve functionality, security, and performance.
3. User Support – Provide ongoing support to users for troubleshooting and assistance.

## Requirements Identification

This chapter outlines the requirements for the PMS, including functional and non-functional requirements, system requirements, user roles, and user levels. These requirements are essential for ensuring the system meets the needs of its users and operates efficiently and effectively.

### Functional and Non-functional requirements

### Functional Requirements

**Inventory Management**

1. Input Details – The system must allow input of product details such as name, quantity, price, and expiration date.
2. Update and Delete Information – Users must be able to update and delete product information as necessary.
3. Validate and Update Database – The system must validate data and update the inventory database accordingly.

**Prescription Tracking**

1. Record Patient Information – The system should record patient information and prescribed medications.
2. Secure Storage – Prescription data must be stored securely.
3. Accessible History – Provide an accessible history of prescriptions for patients.

**Sales Management**

1. Process Purchases – The system should process customer purchases, including product details and payment information.
2. Calculate Total Cost – The system must calculate the total cost of purchases, update the inventory, and generate sales receipts

### Non-Functional Requirements

1. Reliability – Ensure consistent software performance with minimal failures.
2. Availability – Ensure high system operational time and user accessibility.
3. Robustness – Ensure the system handles errors gracefully without failing.
4. Scalability – Ensure the software adapts to various system configurations.
5. Usability – Ensure a user-friendly interface and navigation.
6. Efficiency – Optimize resource utilization for effective software performance.

### System Requirements

### Hardware Requirements

1. PC Specifications – A personal computer with a minimum of 4GB RAM.
2. Storage – At least 50GB of storage space.

### Software Requirements

1. Operating System – Windows 10 or later.
2. Database – MySQL for database management.
3. Programming Language – Java for backend development, React and JavaScript for frontend development

### User Roles

The Pharmacy Management System will have multiple user roles, each with specific responsibilities and permissions:

1. Administrator – Manages system settings, user accounts, and has full access to all system features.
2. Pharmacist – Manages inventory, processes prescriptions, and handles sales transactions.
3. Store Manager – Oversees inventory management and sales, generates reports, and ensures smooth operation of the pharmacy.

### User Levels

User levels define the hierarchy and permissions associated with each role in the system.

**Level 1 – Administrator**

Full system access, including user management, system configuration, and access to all data and reports.

**Level 2 – Store Manager**

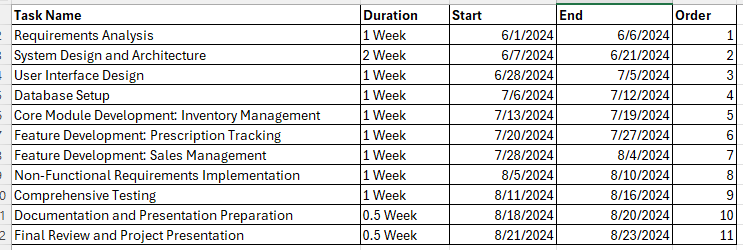
Access to inventory management, sales reports, and prescription tracking, with some administrative permissions.

**Level 3 – Pharmacist**

Access to prescription management, sales processing, and inventory updates.

This structured approach ensures that all aspects of the Pharmacy Management System are carefully planned and implemented, meeting the needs of various users while maintaining high standards of performance, security, and usability.

## Project Plan



## References

- Pressman, R. S., & Maxim, B. R. (2014). Software Engineering: A Practitioner's Approach (8th ed.). McGraw-Hill Education.

- Sommerville, I. (2015). Software Engineering (10th ed.). Pearson.

- Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1994). Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley Professional.

- Fowler, M. (2002). Patterns of Enterprise Application Architecture. Addison-Wesley Professional.

- Freeman, E., & Freeman, E. (2004). Head First Design Patterns. O'Reilly Media.

- Martin, R. C. (2008). Clean Code: A Handbook of Agile Software Craftsmanship. Prentice Hall.

- Kent Beck, (2002). Test-Driven Development: By Example. Addison-Wesley Professional.

- Docker Inc. (2020). Docker Documentation. Retrieved from https://docs.docker.com/

- MySQL Documentation. (2020). Retrieved from https://dev.mysql.com/doc/

- Postman API Platform. (2020). Retrieved from https://www.postman.com/

- GitHub Guides. (2020). Retrieved from https://guides.github.com/

## Appendixes