**Pharmacy Management System**

**Interim Report**

**ICT2233 - Software Engineering**

**Submitted by:**

1. TG/2020/674 – Chandimal Priyamantha
2. TG/2020/707 – K.G.L.Nethsara Kiringoda
3. TG/2020/694 – H.A.R.N.Karunarathna
4. TG/2020/706 – K.G.S. Dissanayake
5. TG/2020/684 – M.A.D.A.M.Arachchi
6. TG/2020/747 – H.Tirasha Dinuki
7. TG/2020/676 – A.Asama
8. TG/2020/704 – T.Madhusha

**Lecturer in charge:** Miss. Chanduni Gamage

**Submission Date:** 2023/10/06

Bachelor of Information and Communication Technology

Department of Information and Communication Technology

Faculty of Technology

University of Ruhuna.



**Declaration**

“We certify that this project does not incorporate without acknowledgment, any material previously submitted for an assignment in any institution, and to the best of my knowledge and belief, it does not contain any material previously published or written by another person or myself except where due reference is made in the text. I also hereby give consent for my dissertation, if accepted, to be made available for photocopying and for interlibrary loans, and for the title and summary to be made available to outside organizations”.

Name Index No Signature

1. Chandimal Priyamantha TG/2020/674 …………………………
2. K.G.L.Nethsara Kiringoda TG/2020/707 …………………………
3. H.A.R.N.Karunarathna TG/2020/694 …………………………
4. K.G.S. Dissanayake TG/2020/706 …………………………
5. M.A.D.A.M.Arachchi TG/2020/684 …………………………
6. H.Tirasha Dinuki TG/2020/747 …………………………
7. A.Asama TG/2020/676 …………………………
8. T.Madhusha TG/2020/704 …………………………

**Preamble**

**Abstract**

The Pharmacy Shop Management Application proposed in this project aims to solve pressing problems faced by small and medium-sized pharmacies. Designed to cater to both small and large pharmacies, this application provides a comprehensive solution for streamlining pharmacy operations.

The primary objectives of this project are as managing personnel information, managing customers, managing inventory, managing financials, report generating, formal communication, and remote pharmaceutical distribution.

By addressing these core aspects of pharmacy management, our Pharmacy Shop Management Application intends to provide an accessible, user-friendly, and efficient solution that empowers pharmacies to enhance their operational workflows and deliver superior customer service. This application promises to bridge the gap between manual processes and modern, streamlined pharmacy management practices, ultimately improving overall business efficiency and financial stability.

**Acknowledgement**

We would like to express our gratitude to all those who have contributed to the successful completion of this endeavor. Their unwavering support, guidance, and encouragement have been invaluable throughout this journey.

First and foremost, we extend our deepest appreciation to our head of the department for their guidance and constant supervision of Ms. Chanduni Gamage, whose expertise, patience, and dedication have been instrumental in shaping this project.

We wish to extend our thanks to all colleagues and friends who have been a constant source of motivation and camaraderie. Your discussions, brainstorming sessions, and moral support have been integral to this undertaking.

**List of figures**

[Figure 1 - Gantt Chart 10](#_Toc146900274)

[Figure 2 – Use case diagram 12](file:///C:\wamp\www\Pharmacy_Shop_Management_System\Doc\report\interm%20report.docx#_Toc146900275)

[Figure 3 – Log In Activity Diagram 21](#_Toc146900276)

[Figure 4- Dashboard Activity Diagram 22](#_Toc146900277)

[Figure 5- Add information Activity Diagram 23](#_Toc146900277)

[Figure 6 – Delete Information Activity Diagram 24](#_Toc146900278)

[Figure 7– Direct Customer Management Activity Diagram 25](#_Toc146900279)

[Figure 8– Formal Communication Activity Diagram 26](#_Toc146900280)

[Figure 9– Medicine Management Activity Diagram 27](#_Toc146900281)

[Figure 10 – Report Generating Activity Diagram 28](#_Toc146900282)

[Figure 11- Remote Order Activity Diagram 29](#_Toc146900283)

[Figure 12- Remote Payment Activity Diagram 30](#_Toc146900277)

[Figure 13 – Reset Password Activity Diagram 31](#_Toc146900284)

[Figure 14– Update Information Activity Diagram 32](#_Toc146900285)

[Figure 15– User Management Activity Diagram 33](#_Toc146900286)

[Figure 16 - Log Out Activity Diagram 34](#_Toc146900287)

[Figure 17 – Log In Sequence Diagram 35](#_Toc146900288)

[Figure 18 - Dashboard Sequence Diagram 36](#_Toc146900289)

[Figure 19 –Manage Direct Customer Order Sequence Diagram 37](#_Toc146900290)

[Figure 20 - Inventory Management Sequence Diagram 38](#_Toc146900291)

[Figure 21 – Report generating Sequence Diagram 39](#_Toc146900292)

[Figure 22 – Remote Order Sequence Diagram 40](#_Toc146900293)

[Figure 23 – Remote payment Sequence Diagram 41](#_Toc146900294)

[Figure 24 – Email Sequence Diagram 42](#_Toc146900295)

[Figure 25 – Manage Purchase Sequence Diagram 43](#_Toc146900296)

[Figure 26 – Logout Sequence Diagram](#_Toc146900297) 44

[Figure 27 – ER diagram 45](#_Toc146900298)

[Figure 28 – Architecture Design](#_Toc146900299) 48

[Figure 29 – Admin Registration Interface 49](#_Toc146900300)

[Figure 30 – Admin Log in 49](#_Toc146900301)

[Figure 31- Admin Manage Interface 50](#_Toc146900277)

[Figure 32- Admin Security Authorization 50](#_Toc146900277)

[Figure 33- User Log in 51](#_Toc146900277)

[Figure 34- Staff Manage 51](#_Toc146900277)

[Figure 35- Dashboard Interface 52](#_Toc146900277)

[Figure 36- Direct Customer Order 52](#_Toc146900277)

[Figure 37- Direct Customer Order Report 53](#_Toc146900277)

[Figure 38- Remote Customer Order 53](#_Toc146900277)

[Figure 39- Remote Customer Order Payment 54](#_Toc146900277)

[Figure 40-Remote Customer Order Report 54](#_Toc146900277)

[Figure 41- Place Purchase 55](#_Toc146900277)

[Figure 42-Complete Purchase 55](#_Toc146900277)

[Figure 43- Purchase Report 56](#_Toc146900277)

[Figure 44- Inventory Interface 56](#_Toc146900277)

[Figure 45- Email Interface 57](#_Toc146900277)

**CHAPTERS**

Chapter 1: Introduction

1.1 Statement of the Problem

1.2 Aims and Objectives

1.3 Scope of the Project

1.4 Gantt Chart

Chapter 2: System Analysis & Design

2.1 Use case Diagram

2.2 Use case narratives

2.3 Activity Diagrams

2.4 Sequence Diagrams

2.5 ER diagram

2.6 Software Requirement Specification

Chapter 3: Proposed System Design

3.1 Architecture Design

3.2 Proposed System Interfaces

3.3 Report Design

3.4 Test plan for the system

3.5 Proposed Software Engineering Methodology

Conclusion

References

****

**Pharmacy Management System**

**Chapter 1: Introduction**

We propose to design and build a thorough pharmacy shop management application that will streamline business processes and ensure effective staff management, customer engagement, medication inventory control, precise financial tracking, formal communication facilities (email), remote medication distribution, and strong reporting capabilities. The app will target both small and large pharmacies and be user-friendly and accessible.

* 1. **Statement of the problem**

Due to manual processes, many small- and medium-sized pharmacies find it difficult to operate their business effectively, which causes mistakes in the management of inventory, customers, and finances.

Inefficiencies and significant revenue losses result from the lack of a centralized system for managing the inventory of medicines, customer information, and employee data.

* 1. **Aims and Objectives**

The project's goal is to provide a user-friendly, effective Pharmacy Shop Management Application that improves the workflow of pharmacy operations as a whole. The precise goals consist of:

1. Managing personnel Information: Create a module that makes it simple to handle personnel details such identities, responsibilities, and schedules.

2. Managing Customers: Create a customer management module that simplifies order history, communication, and customer registration.

3. Managing Medicine: Develop a module to check stock levels, expiration dates, and enable reordering alerts for medicines.

4. Managing Financials: Create a module for accounting and invoicing so that revenues, expenses, and client bills can all be tracked accurately.

5. Reporting: Create a reporting module that offers details on sales patterns, inventory turnover, profit margins, and other important performance metrics.

6. Formal Communication: Create a module that enables communication with clients and vendors.

7. Remote pharmaceutical Distribution: Create a pharmaceutical distribution module that offers consumers precise and efficient service.

* 1. **Scope of project**

1.Planning the project (Week 1):

- Specify the project's goals, objectives, and needs.

- Divide up team members' roles and tasks.

- Create lines of communication to facilitate productive teamwork.

|  |  |  |  |
| --- | --- | --- | --- |
| **Task** | **Duration (Days)** | **Start** | **End** |
| Planning the project | 07 | 19/08/2023 | 25/08/2023 |

2.Designing UI/UX (Week 2):

- Produce mockups and wireframes for the user interface of the program.

- Collect suggestions and improve the design for the best user experience.

|  |  |  |  |
| --- | --- | --- | --- |
| **Task** | **Duration (Days)** | **Start** | **End** |
| Designing UI | 07 | 26/08/2023 | 31/08/2023 |

3. Database design (Week 3):

- Create a database schema to contain information about the personnel, customers, medications, and finances.

- Ensure scalability and data integrity.

|  |  |  |  |
| --- | --- | --- | --- |
| **Task** | **Duration (Days)** | **Start** | **End** |
| Database Design | 07 | 26/08/2023 | 31/08/2023 |

4.Development on the front end (Week 4):

- Create user interfaces for different modules utilizing cutting-edge JavaFX technologies.

Develop a responsive design strategy.

|  |  |  |  |
| --- | --- | --- | --- |
| **Task** | **Duration (Days)** | **Start** | **End** |
| Development on the front end | 07 | 01/09/2023 | 07/09/2023 |

5.Development on the back end (Week 5):

- Create the essential logic and features of the program.

- Connect the front end to the back-end services and database.

|  |  |  |  |
| --- | --- | --- | --- |
| **Task** | **Duration (Days)** | **Start** | **End** |
| Development on the back end | 07 | 08/09/2023 | 14/09/2023 |

6. Quality assurance and testing (Week 6):

- Carry out exhaustive testing to find and address any flaws or problems.

- Ensure system stability and data security.

|  |  |  |  |
| --- | --- | --- | --- |
| **Task** | **Duration (Days)** | **Start** | **End** |
| Quality assurance and testing | 07 | 15/09/2023 | 21/09/2023 |

7. Documentation (Week 7):

- Produce thorough documentation for installation, upkeep, and user instructions.

|  |  |  |  |
| --- | --- | --- | --- |
| **Task** | **Duration (Days)** | **Start** | **End** |
| Documentation | 01 | 22/09/2023 | 29/09/2023 |

8.Deployment and Training (Week 8):

- Install the application on a client machine or hosting platform.

- Teach pharmacy personnel how to use the system efficiently.

|  |  |  |  |
| --- | --- | --- | --- |
| **Task** | **Duration (Days)** | **Start** | **End** |
| Select a client | 07 | 30/09/3023 | 06/10/2023 |

9.Finalisation and Presentation (Week 9):

- Check the application one last time to make sure all requirements have been completed.

- Create a presentation that highlights the features and functionality of the program.

|  |  |  |  |
| --- | --- | --- | --- |
| **Task** | **Duration (Days)** | **Start** | **End** |
| Select a client | 07 | 07/10/2023 | 13/10/2023 |

* 1. **Gantt Chart**

A graph with blue rectangular objects

Description automatically generated with medium confidence

Figure 1 - Gantt chart

**Chapter 2: System Analysis and Design**

**2.1 Use case diagram**

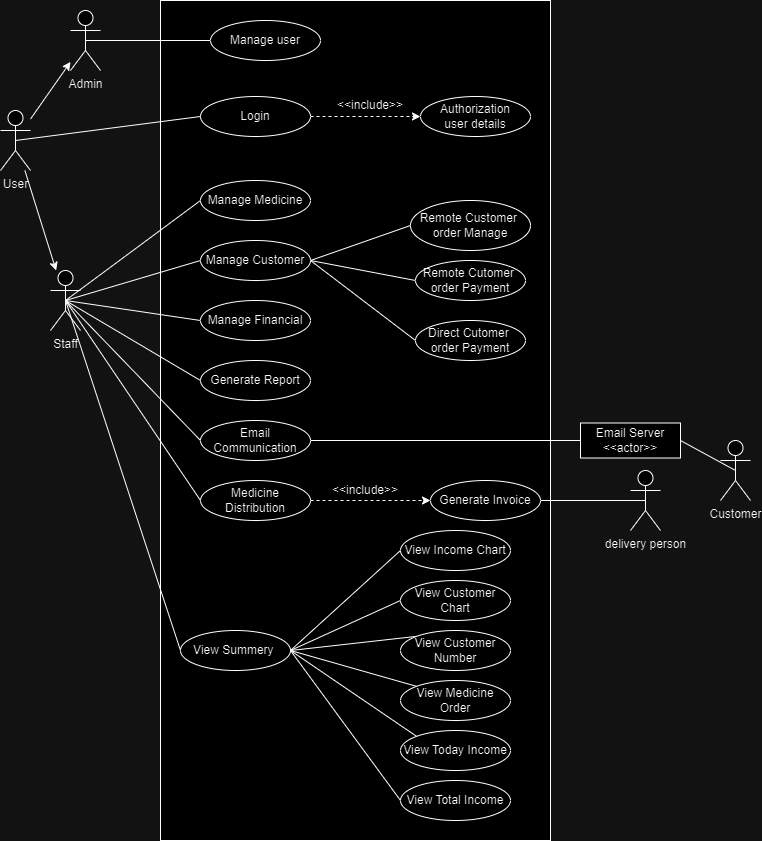
****

Figure – Use case diagram

**2.2 Use case narratives**

|  |  |
| --- | --- |
| **Use case ID** | 01 |
| **Name** | Login |
| **Participating actor** | User |
| **Description** | The user login to the system |
| **Entry condition** | The user should have a username and password |
| **Basic course** | 1. User Request login form.  2. The System returns login form.  3. User fill the form.  4. The system send validate response.  5. User request reset the password.  6. The system send password reset form.  7. User filled out the form. |
| **Alternative course** | 3 a. The user entered valid user credentials.  1.The system display “Login Successful”  b. The user entered invalid credentials.  1.The system send Error message “Invalid Credentials”  7 a. User filled out reset form.  2. The system send “Password updated successfully” message. |

|  |  |
| --- | --- |
| **Use case ID** | 02 |
| **Name** | Manage Medicine |
| **Participating actor** | User |
| **Description** | User Request to Add, Update, Delete Medicine details |
| **Entry condition** | User logging in the system |
| **Basic course** | 1. User request the system to add medicine.  2. The system send form to add medicine details.  3. User fill and submit the form.  4. User request Update the medicine details.  5. The system send medicine list.  6. User request particular medicine.  7. The system return update medicine form.  8. User fill and submit updated details.  9. User request delete the medicine details.  10. The system send medicine list.  11. User select the medicine to delete. |
| **Alternative course** | 3 a. User added medicine details correctly   1. The system display “Added successfully” message.   8 a. User updated medicine details correctly   1. The system display “update successfully” message.   11 a. User select a medicine to delete   1. The system display “Delete successfully” message. |

|  |  |
| --- | --- |
| **Use case ID** | 03 |
| **Name** | Remote orders |
| **Participating Actor** | User |
| **Description** | User can order the medicine using remote access |
| **Entry condition** | User login in to the system |
| **Basic course** | 1. User, request remote customer order manage dashboard.  2. The system return remote customer order manage dashboard.  3. User fill and submit customer details.  4. User fill and submit medicine details.  5. User clicks place order button.  6. The system generate invoice. |
| **Alternative course** | 3 a. User filled and submit valid customer details.   1. Then, system generate invoice and send “Successful message”   b. User filled and submit invalid customer details.  1. Then, system send “error message” |

|  |  |
| --- | --- |
| **Use case ID** | 04 |
| **Name** | Generate Report |
| **Participating Actor** | Staff |
| **Description** | Review the report |
| **Entry condition** | User logging in to the system |
| **Basic course** | 1 .Staff send, Request to generate the report.  2. Systems Send generated report.  3. Staff Request Soft copy of report.  4. System send soft copy to the staff.  5. Staff Request to print the report.  6. Staff got printed report.  7. Staff Request to close the report view |
| **Alternative course** |  |

|  |  |
| --- | --- |
| **Use case ID** | 05 |
| **Name** | Remote customer order payment |
| **Participating Actor** | Staff |
| **Description** | Remote customer paying their order |
| **Entry condition** | Staff logging in the system |
| **Basic course** | 1. Staff, request remote order payment dashboard.  2. System Load data to dashboard.  3.Staff Enter ID  4. System send data.  5.staff click verification checkbox  6. System display order details & enable pay button.  7. Staff click pay button. |
| **Alternative course** | 3 a. Staff entered invalid id  1.The system display “Error message”  7 a. Staff click pay button   1. The system display “successful message” |

|  |  |
| --- | --- |
| **Use case ID** | 06 |
| **Name** | Manage purchases |
| **Participating Actors** | Staff, Supplier |
| **Description** | Purchase medicine from suppliers |
| **Entry condition** | Staff logging in the system |
| **Basic course** | 1. Staff, request purchase order.  2. System provide purchase order form.  3. Staff submit purchase order form.  4. Supplier confirm purchase order by receipt.  5. Staff update purchase order status.  6. Supplier deliver supplies.  7. Staff record receive supplies.  8. Staff send invoice payment.  9. Supplier send payment confirmation to staff. |
| **Alternative course** |  |

|  |  |
| --- | --- |
| **Use case ID** | 07 |
| **Name** | Manage DCO |
| **Participating Actor** | Direct Customer, Staff |
| **Description** | Direct customer physically orders and get medicines. |
| **Entry condition** |  |
| **Basic course** | 1. Direct customer place order request.  2. Staff check product availability.  3. The system send availability status.  4. Staff send confirm product availability.  5. Direct customer confirm order.  6. Staff initiate payment process.  7. Staff send record order details.  8. Staff reserve ordered products.  9. Direct customer get order confirmation.  10. Direct customer pickup order.  11. Staff hand over order. |
| **Alternative course** | 6 a. Staff initiate payment process.  1.System send “successful payment transaction”  7 a. Staff send record order details.  1.System send “successfully record order details”  8 a. Staff reserve ordered products.   1. System send reservation confirmation message. |

|  |  |
| --- | --- |
| **Use case ID** | 08 |
| **Name** | Inventory management (medicine) |
| **Participating Actor** | Staff |
| **Description** | Staff Add and Edit medicine details |
| **Entry condition** | Staff logging in to the system |
| **Basic course** | 1. Staff request inventory page.  2. System send medicine data.  3. Staff request add or edit medicine.  4. System return form to staff.  5. Staff fill medicine details. |
| **Alternative course** | 5 a. Staff filled out valid details  1. The system display “Successful message”.  b. Staff filled out invalid details  1.The system display “Error message” |

|  |  |
| --- | --- |
| **Use case ID** | 09 |
| **Name** | Dashboard |
| **Participating Actor** | User |
| **Description** | User wants to get dashboard |
| **Entry condition** | User logging in to the system. |
| **Basic course** | 1. User request dashboard.  2. The system send data and load data to dashboard. |
| **Alternative course** |  |

**2.3 Activity diagrams**

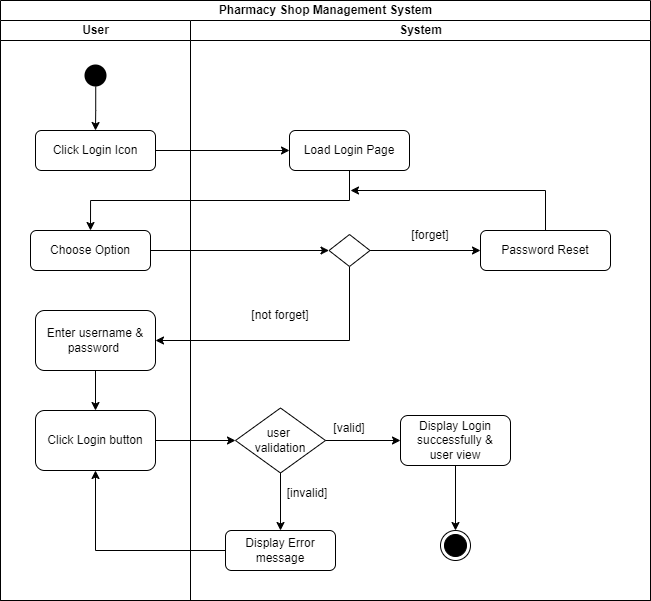
****

Figure 3– Log in Activity Diagram

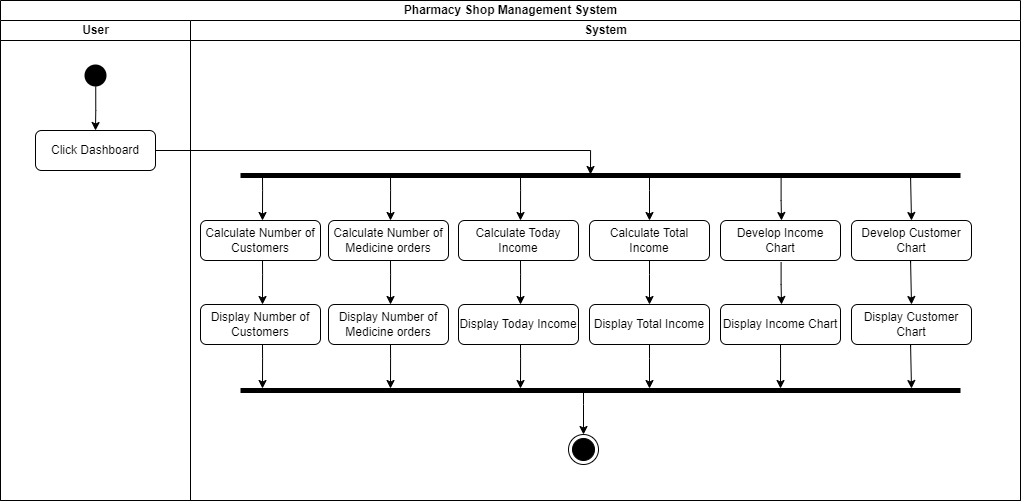


Figure 4– Dashboard Activity Diagram

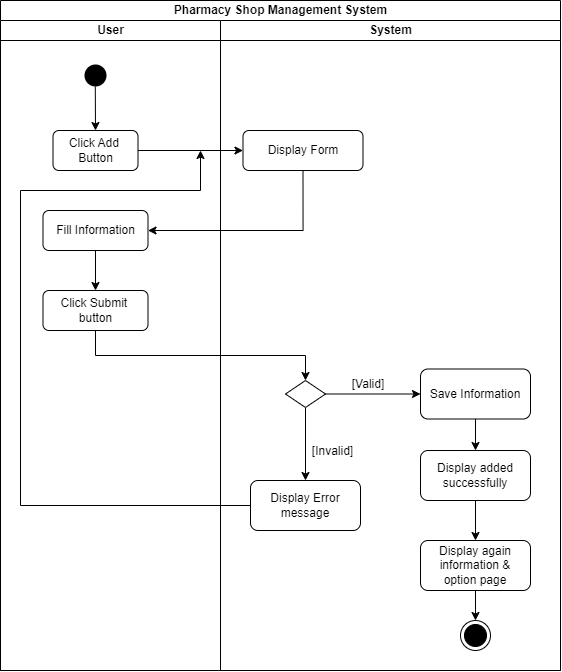
****

Figure 5- Add information Activity Diagram

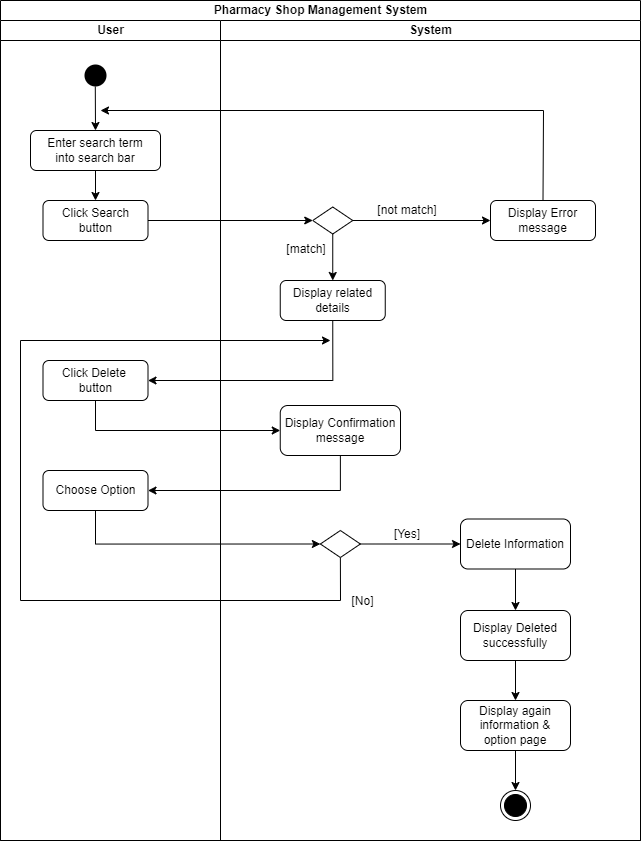
****

Figure 6 – Delete Information Activity Diagram

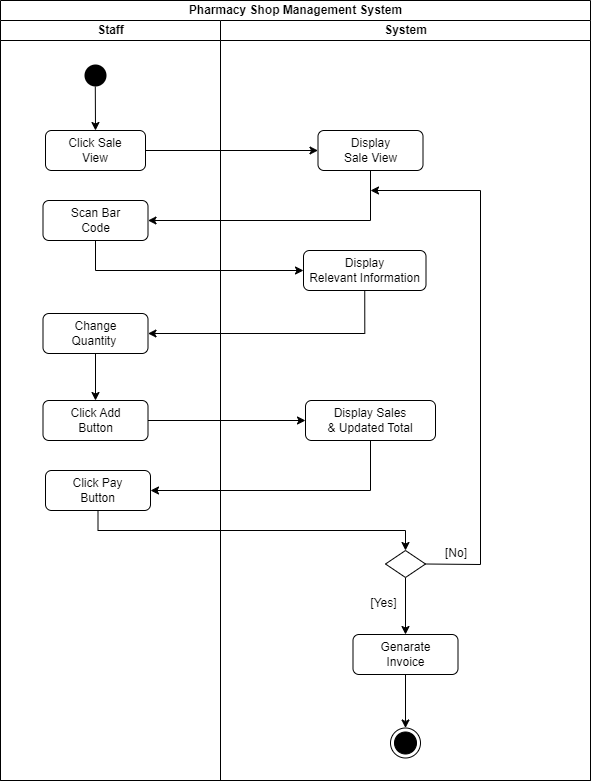
****

Figure 7 – Direct Customer Management Activity Diagram

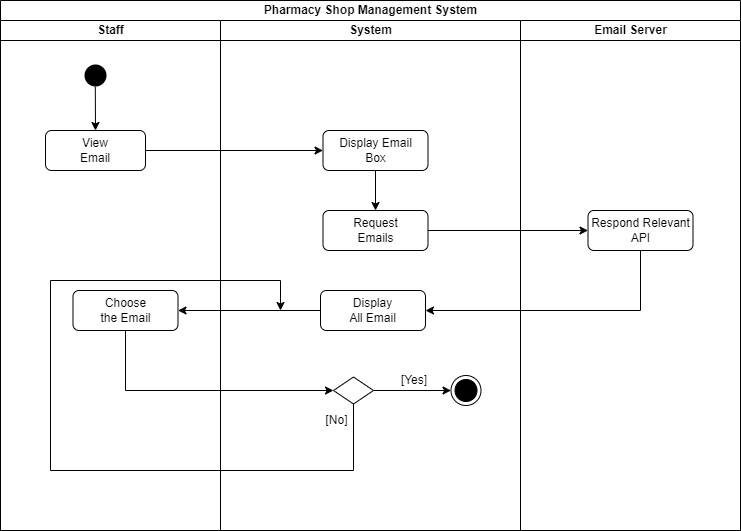
****

Figure 8 – Formal Communication Activity Diagram

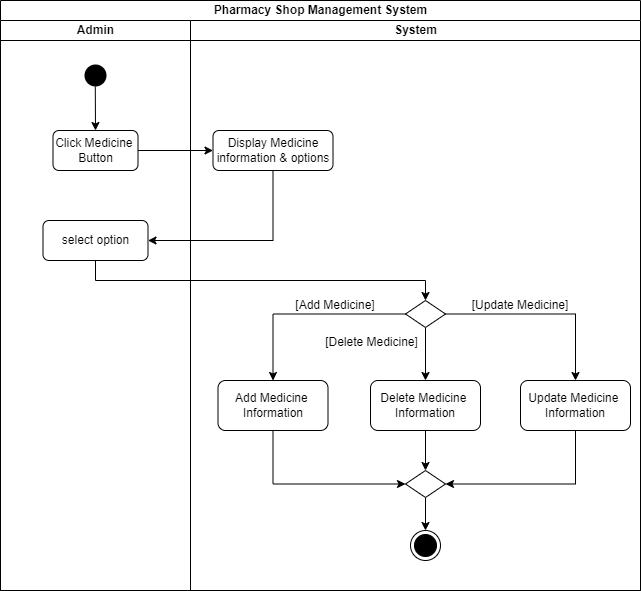
****

Figure 9 – Medicine Management Activity Diagram

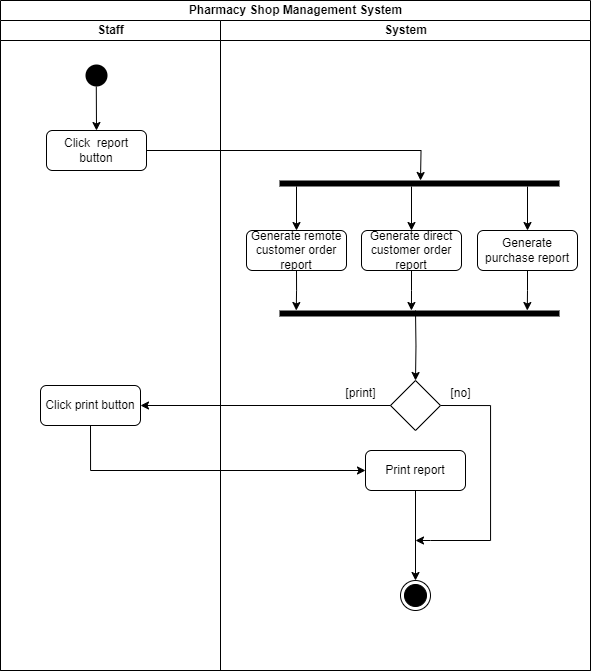
****

Figure 10 – Report Generating Activity Diagram

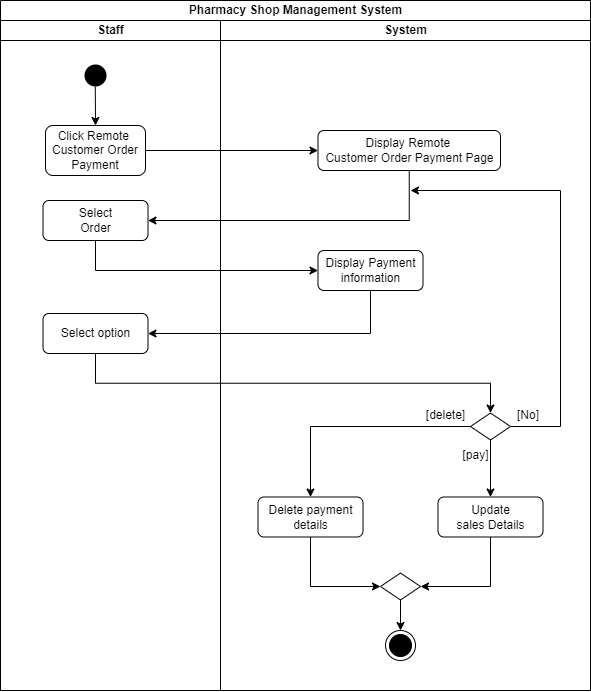


Figure 11 – Remote Payment Activity Diagram

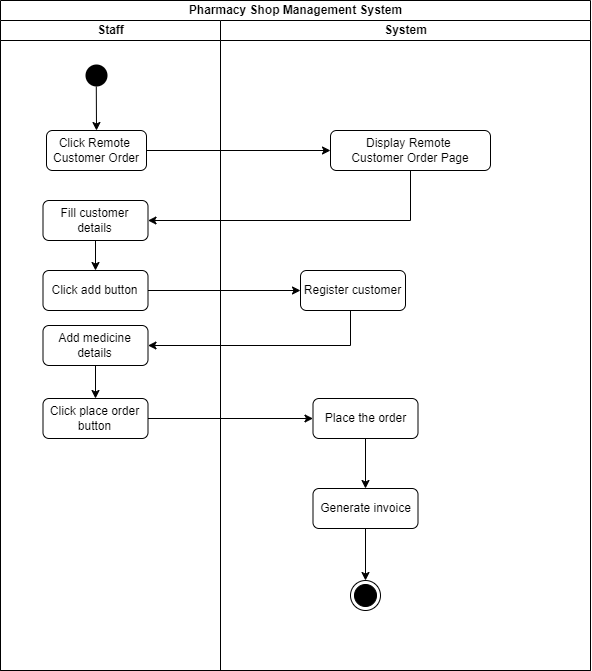


Figure 12 - Remote Order Activity Diagram

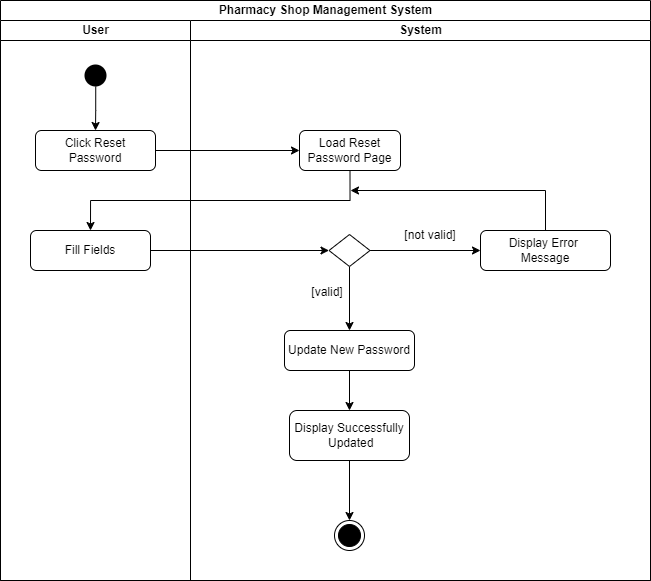
****

Figure 13 – Reset Password Activity Diagram

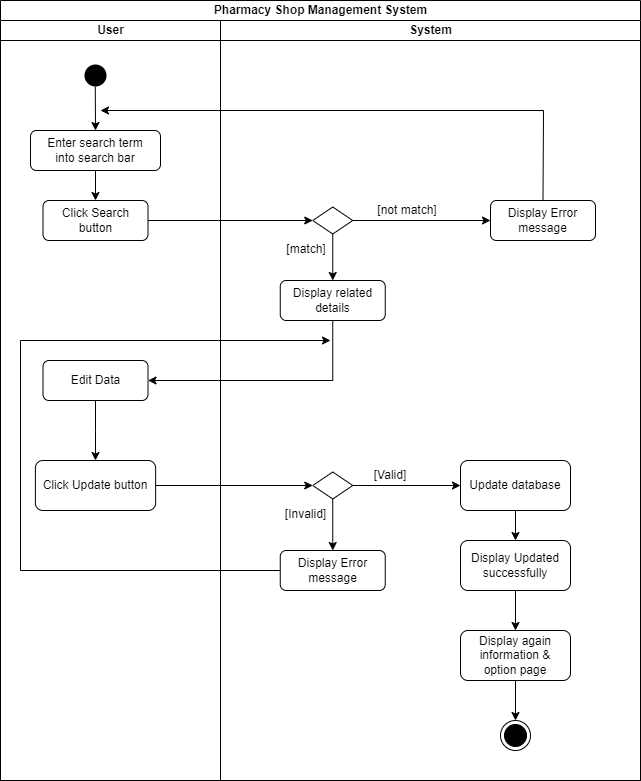
****

Figure 14 – Update Information Activity Diagram

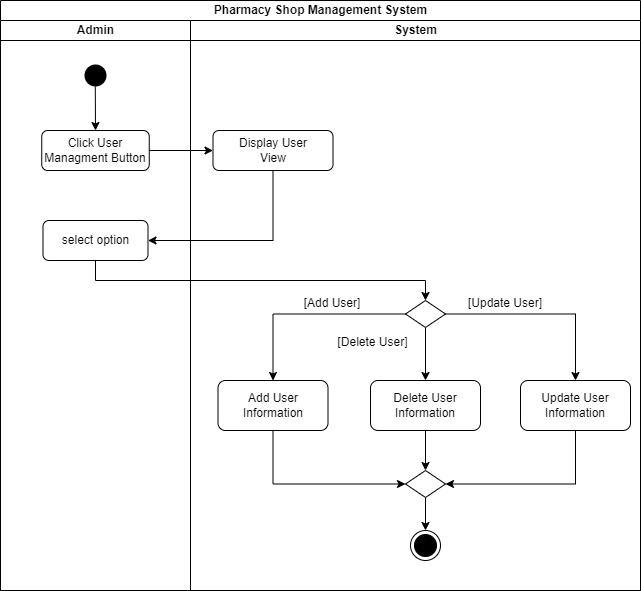
****

Figure 15 – User Management Activity Diagram

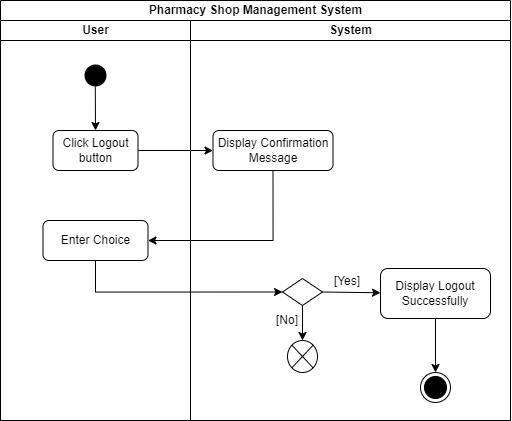
****

Figure 16- Log out Activity Diagram

**2.4 Sequence Diagrams**

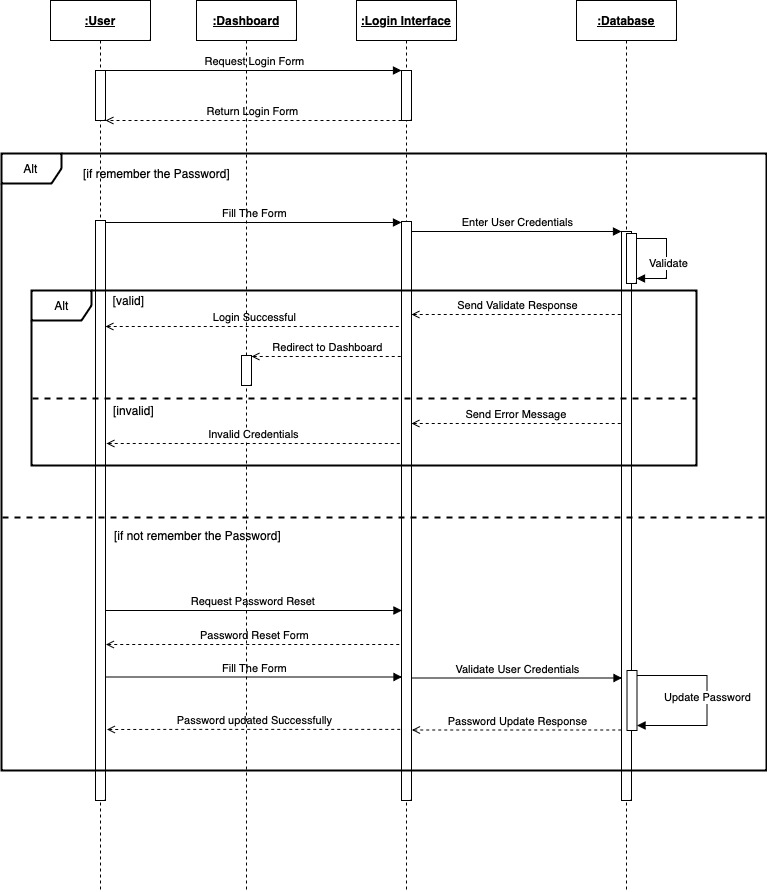
****

Figure 17 – Log in Sequence Diagram

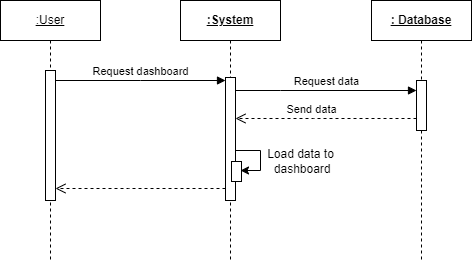
****

Figure 18 - Dashboard Sequence Diagram

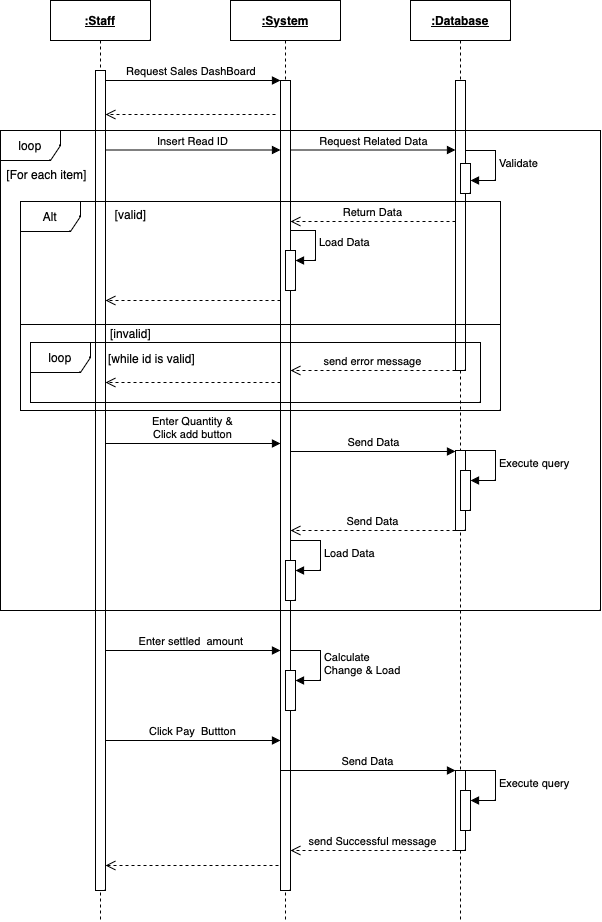
****

Figure 19 – Manage Direct Customer Order Sequence Diagram

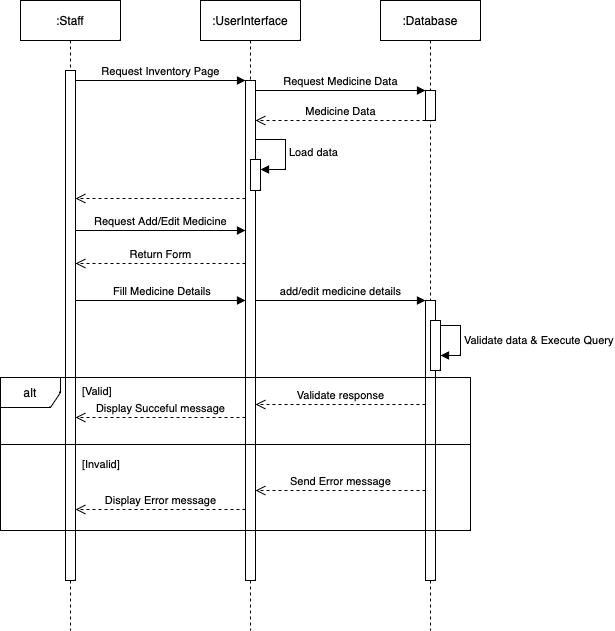
****

Figure 20 - Inventory Management Sequence Diagram

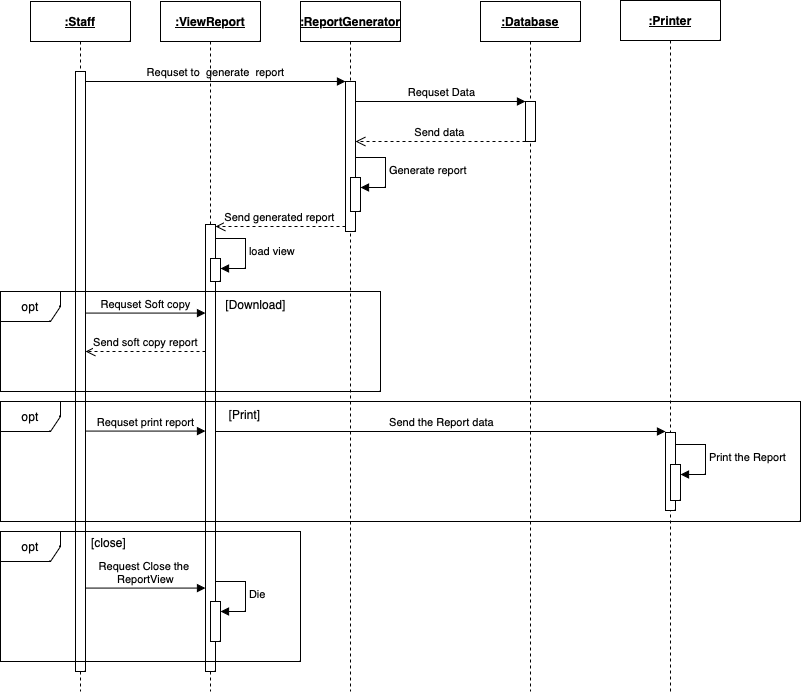


Figure 21 – Report generating Sequence Diagram

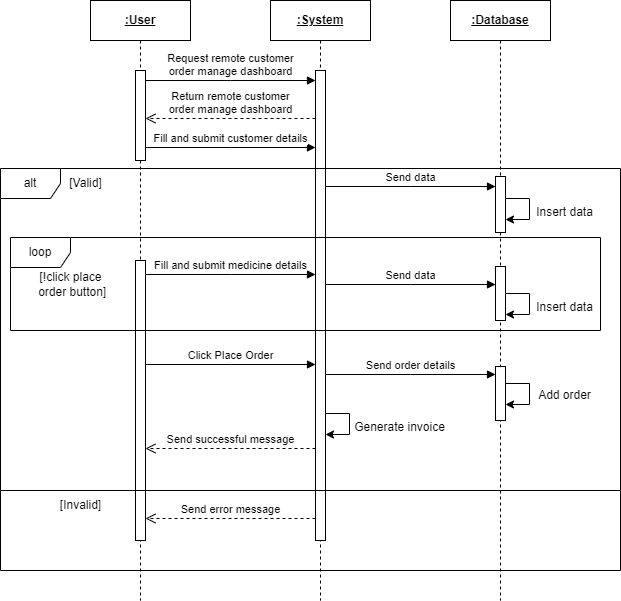


Figure 22 – Remote Order Sequence Diagram

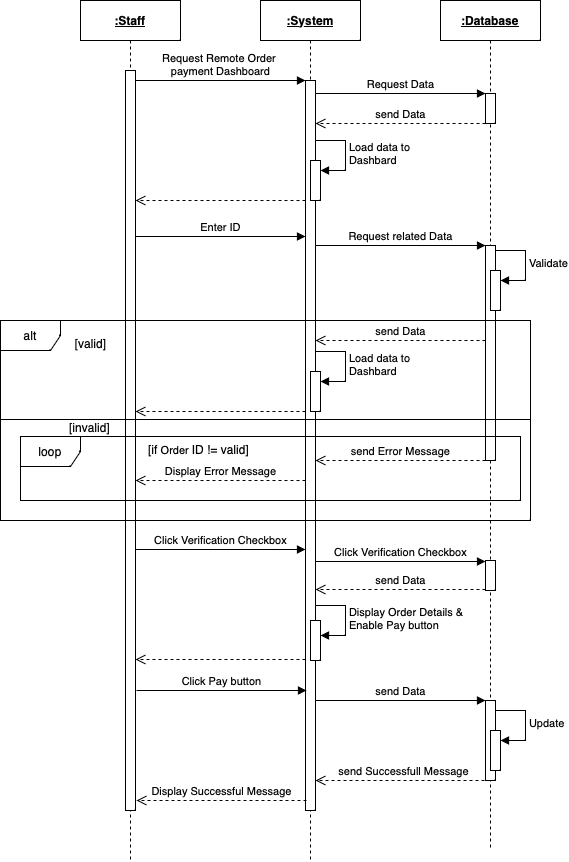


Figure 23 – Remote payment Sequence Diagram

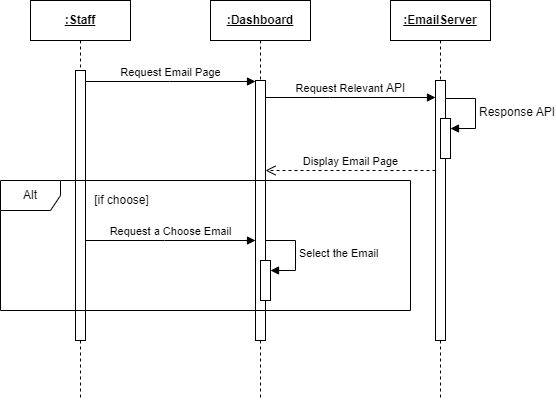
****

Figure 24 – Email Sequence Diagram

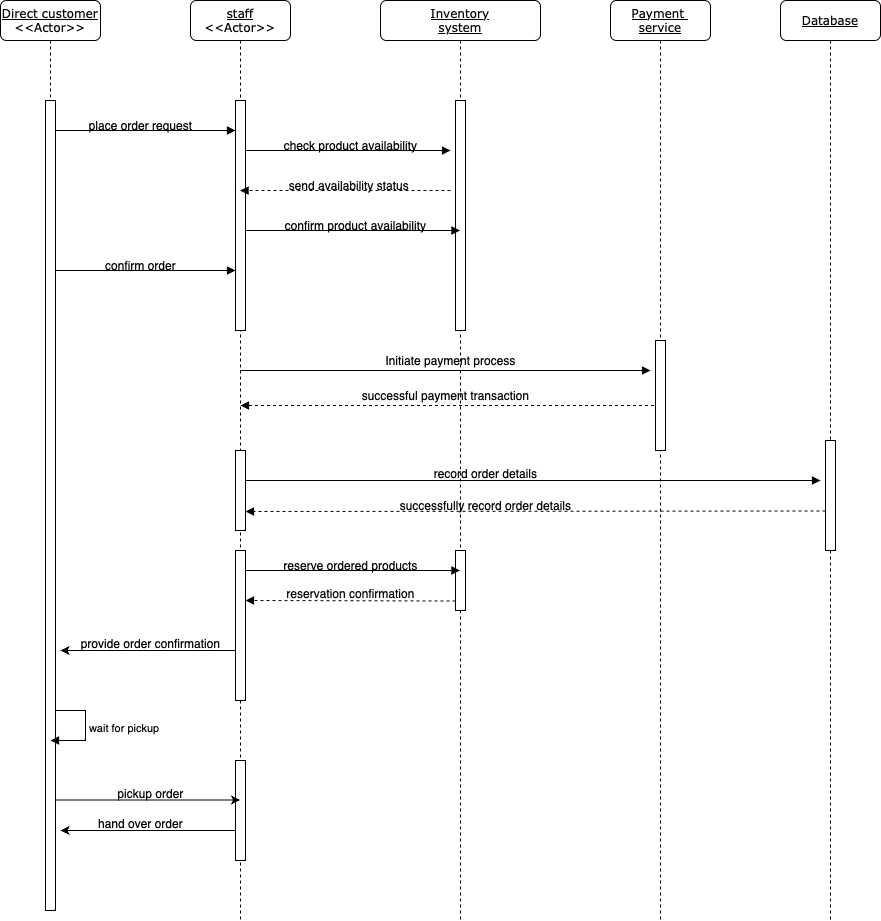
****

Figure 25 – Manage purchase Sequence Diagram

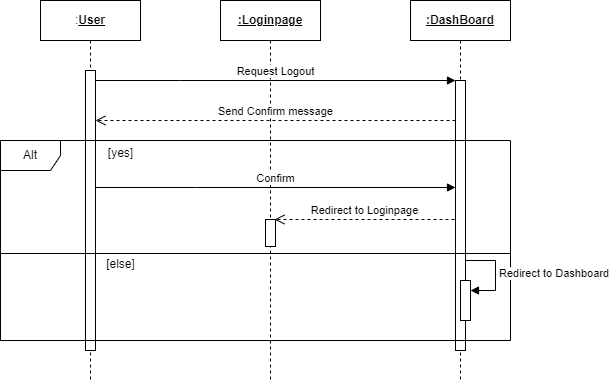
****

Figure 26 –Log out Sequence Diagram

**2.5 ER Diagram**

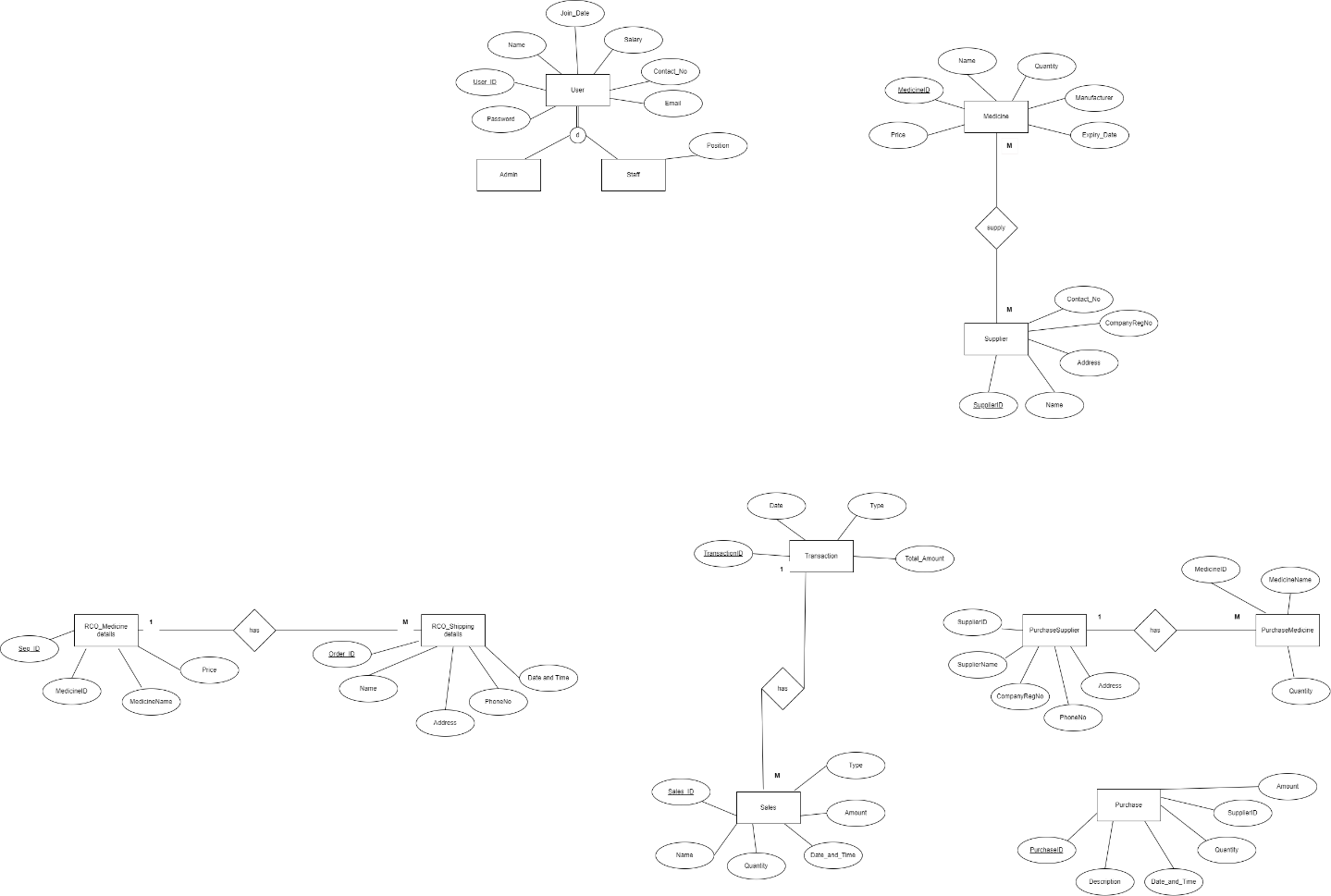
****

Figure 27 – ER Diagram

**2.6 Software Requirement Specification**

**Functional requirements**

A Functional Requirement is a document that describes the essential technical requirements of a system or feature, including the procedures by which it can be determined that requirements have been met. The following points represent the functional requirements of pharmacy management system.

|  |  |
| --- | --- |
| **ID** | **Description** |
| **Admin** | |
| 1 | Shall be able to register. |
| 2 | Shall be able to log in to the system. |
| 3 | Shall be able to authorize the system |
| 4 | Shall be able to manage user |
| 5 | Shall be able to generate and manage reports. |
| 6 | Shall be able to manage inventory |
| **Staff** | |
| 1 | Shall be able to log in to the system. |
| 2 | Shall be able to manage direct and remote customer orders |
| 3 | Shall be able to manage payments. |
| 4 | Shall be able to manage medicine purchases |
| 5 | Shall be able to manage sales. |
| 6 | Shall be able to view today’s summarization. |

**Non-functional requirements**

Non-functional requirements are the requirements that are not directly affect to the system. [1]

* Performance and scalability. How fast does the system return results? How much will this performance change with higher workloads?
* Portability and compatibility. Which hardware, operating systems, and browsers, along with their versions does the software run on? Does it conflict with other applications and processes within these environments?
* Reliability, maintainability, availability. How often does the system experience critical failures? How much time does it take to fix the issue when it arises? And how is user availability time compared to downtime?
* Security. How well are the system and its data protected against attacks?
* Localization. Is the system compatible with local specifics?
* Usability. How easy is it for a customer to use the system?

**Chapter 3: Proposed System Design**

**3.1 Architecture Design**

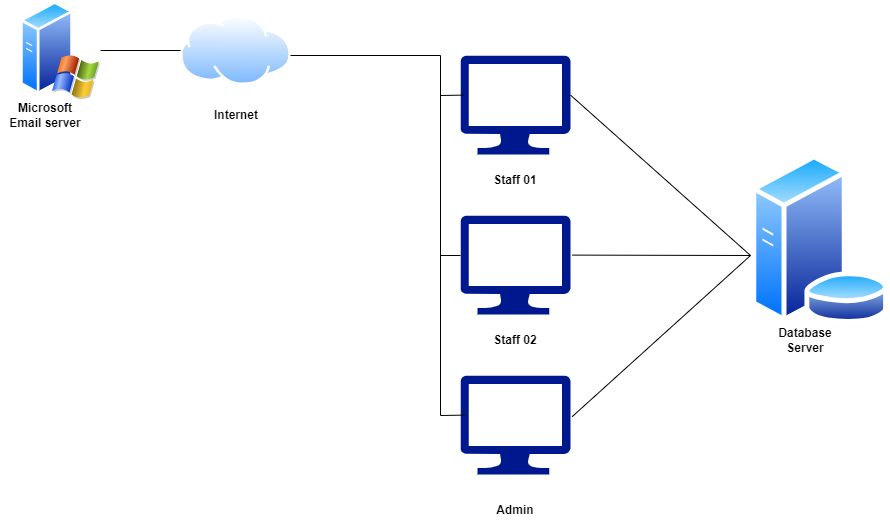


Figure 28 - Architectural Design

**3.2 Proposed System Interfaces**

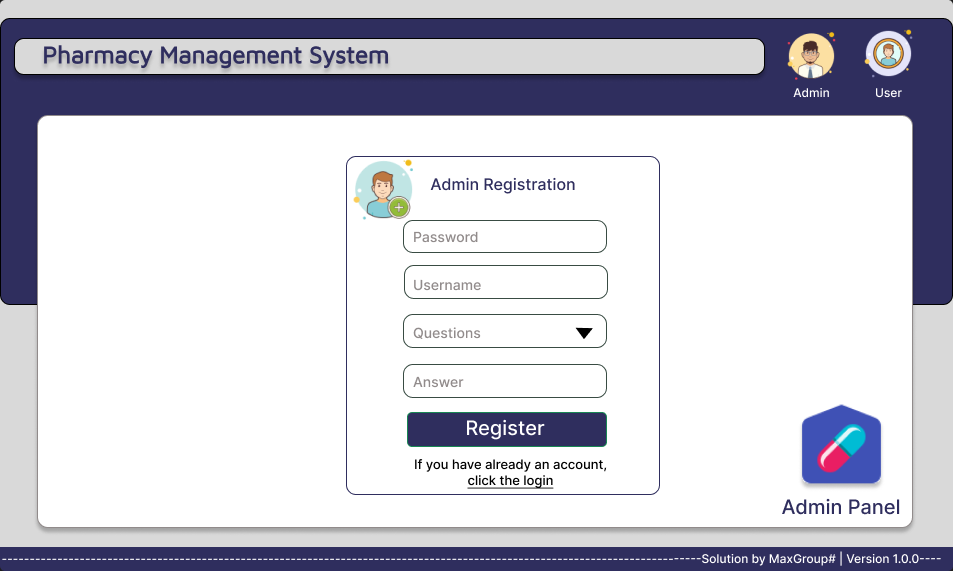


Figure 29 – Admin Register

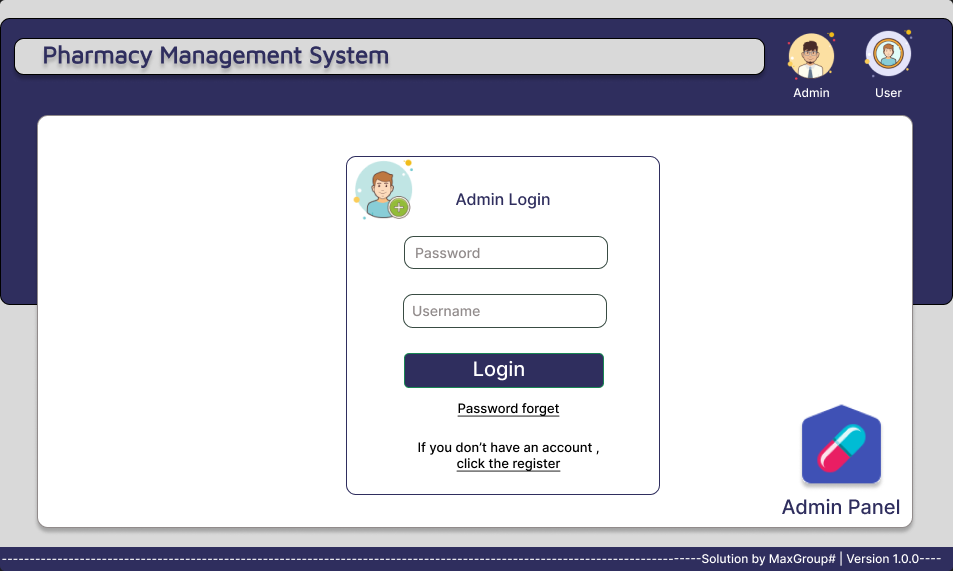


Figure 30 – Admin Log in

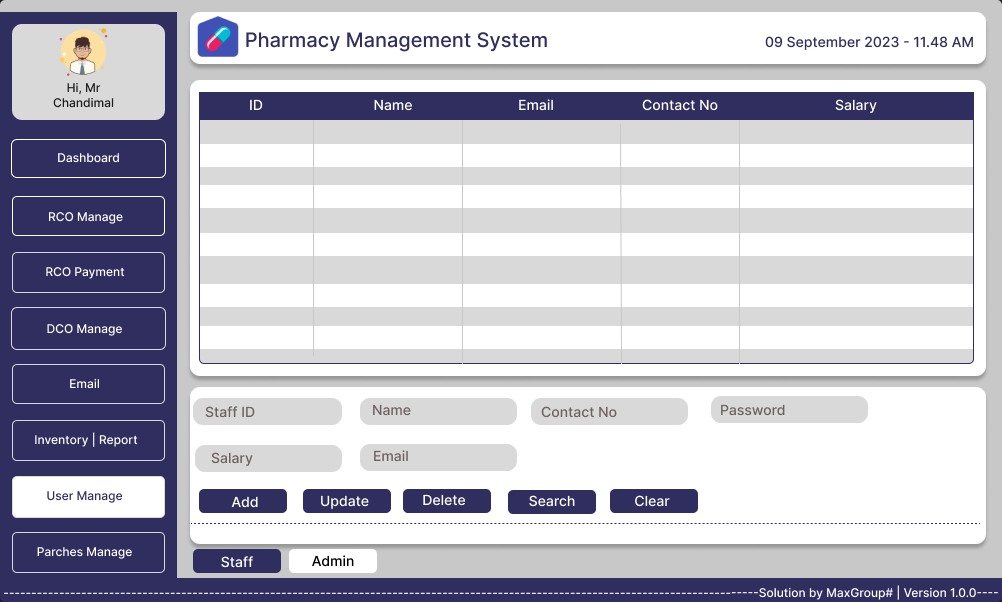


Figure 31 – Admin Manage

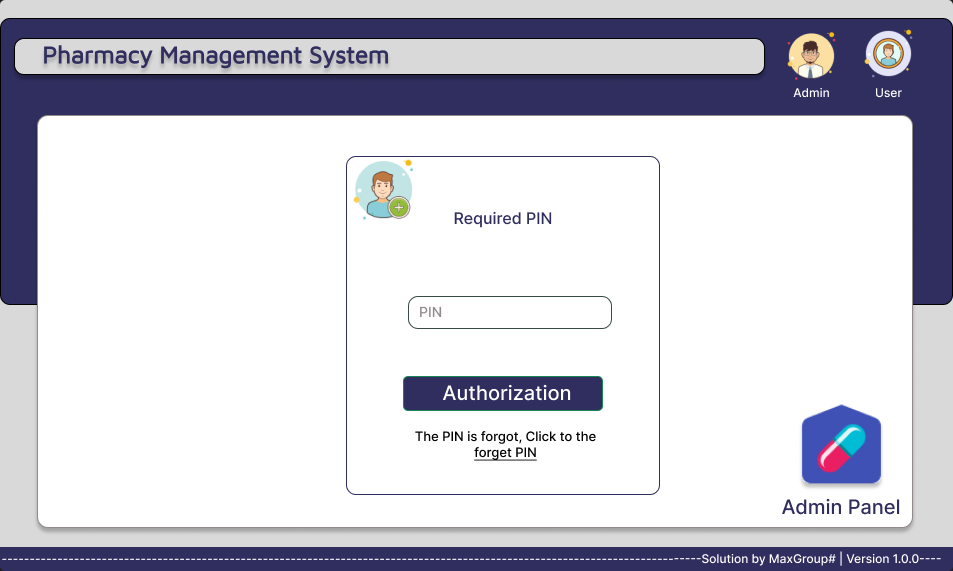


Figure 32 – Admin Security Authorization

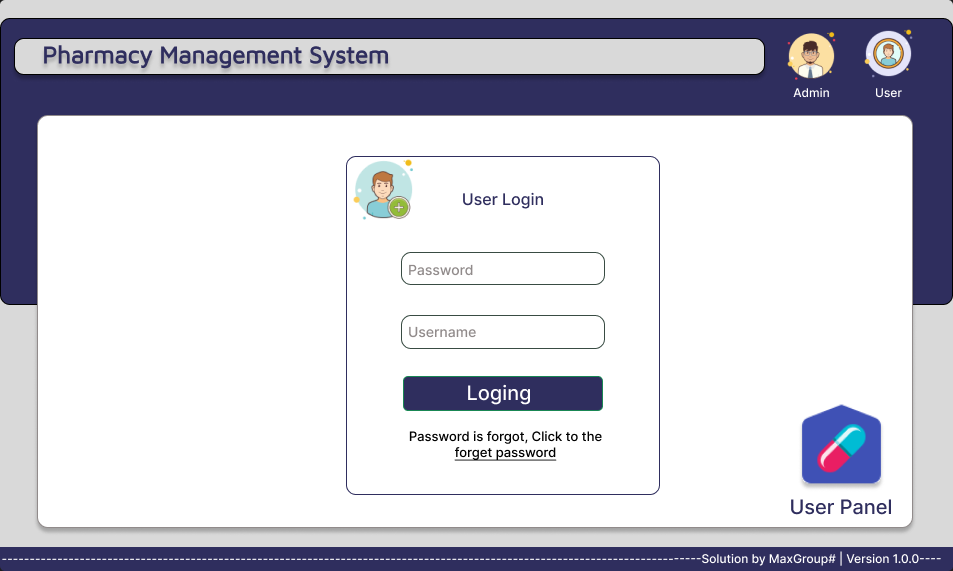


Figure 33 – User Log in

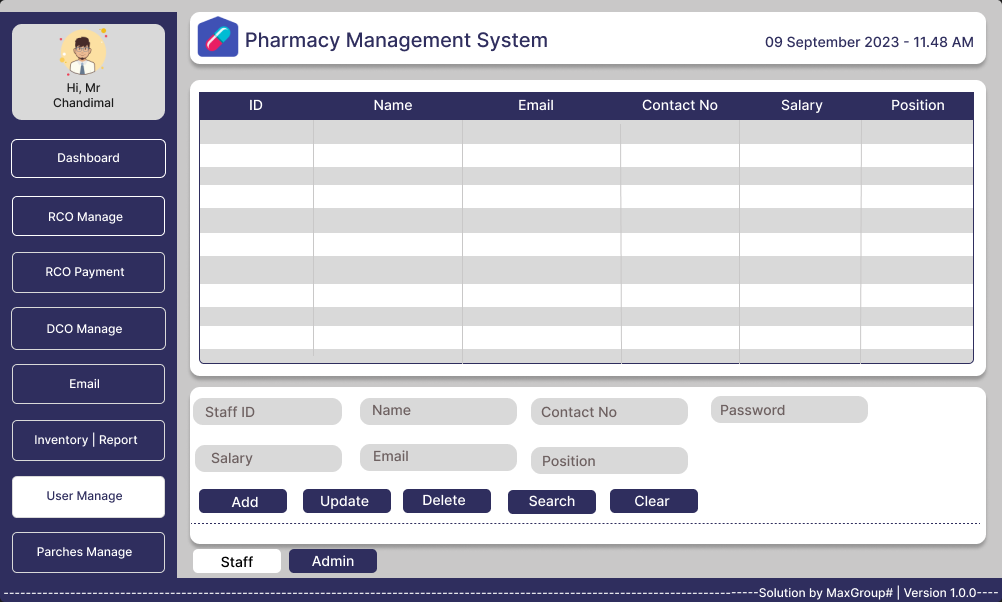


Figure 34 –Staff manage

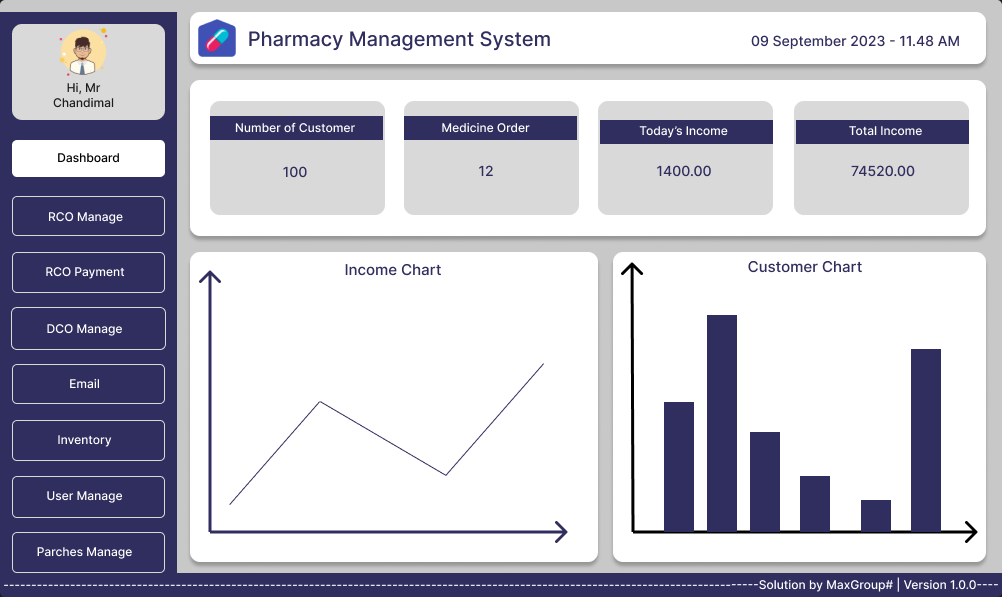


Figure 35 – Dashboard Interface

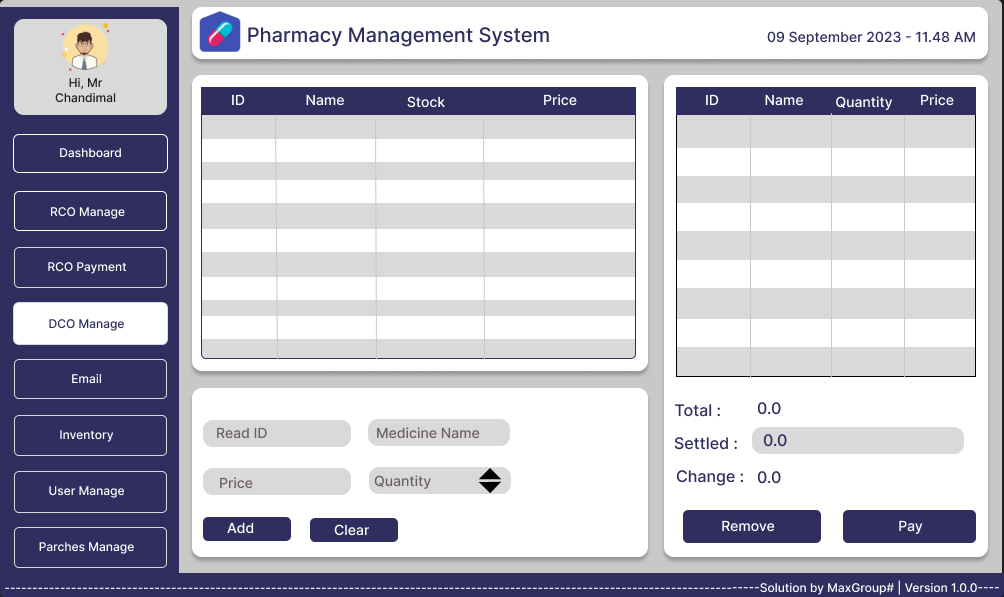


Figure 36 – Direct Customer Order

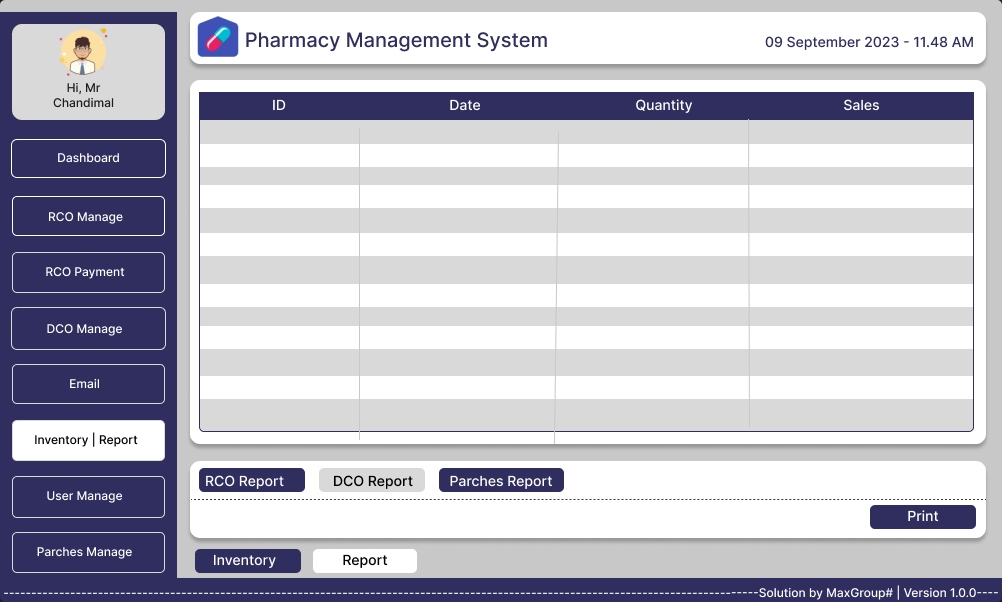
****

Figure 37 – Direct Customer Order Report

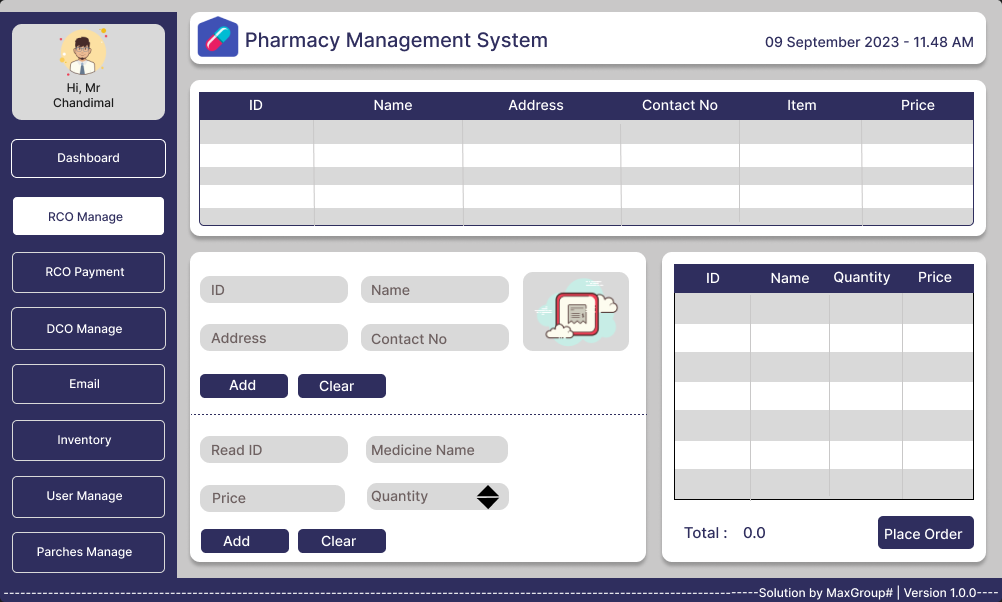


Figure 38 – Remote Customer Order

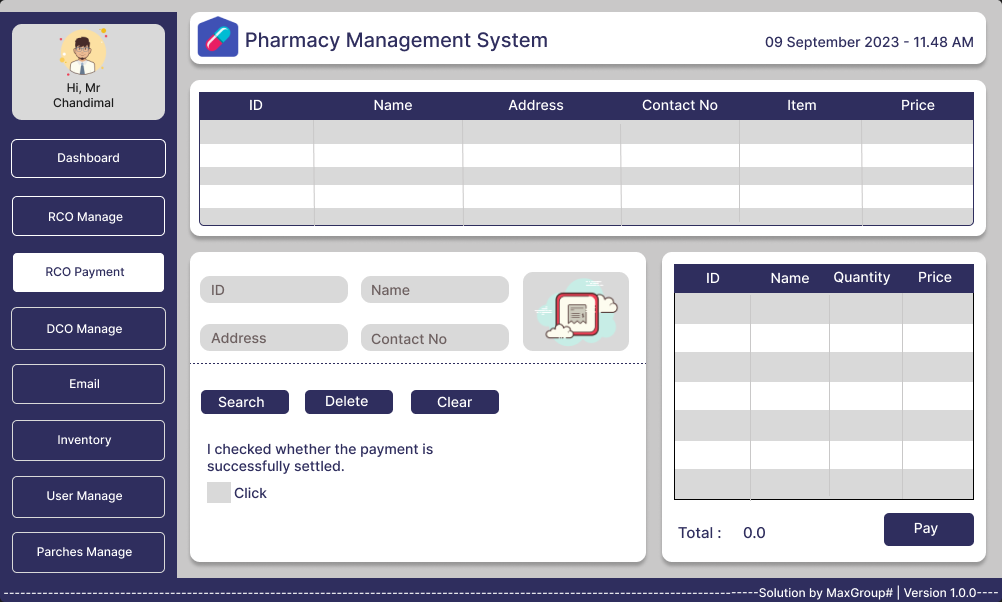


Figure 39 – Remote Customer Order Payment

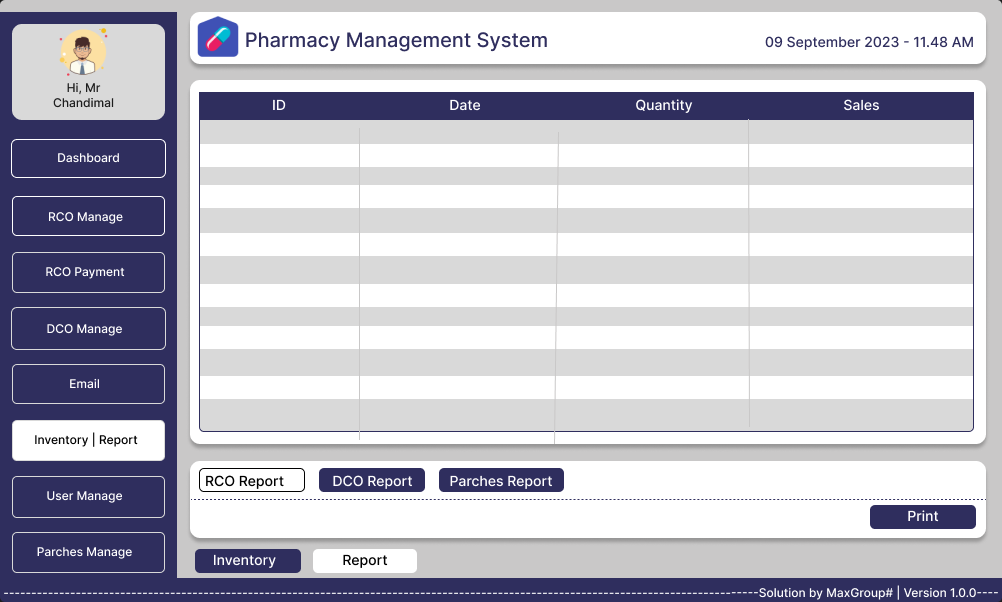


Figure 40 – Remote Customer Order Report

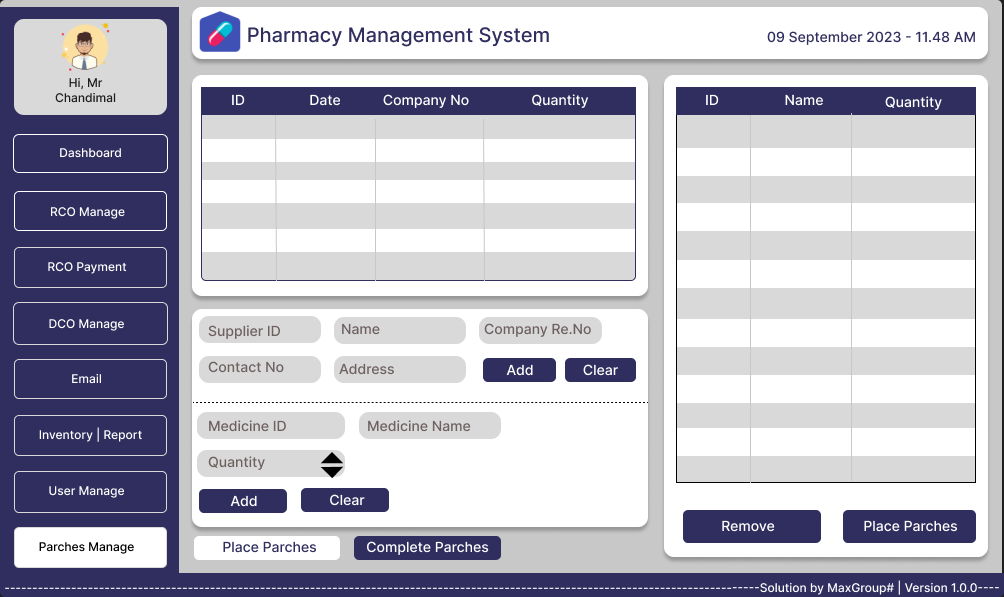


Figure 41 – Place Purchase

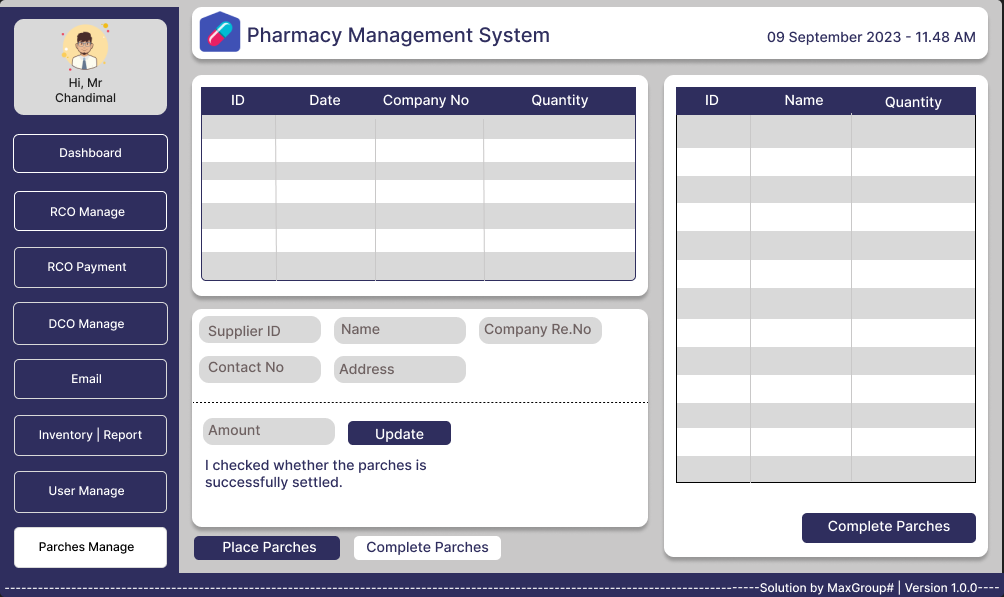


Figure 42 – Complete Purchase

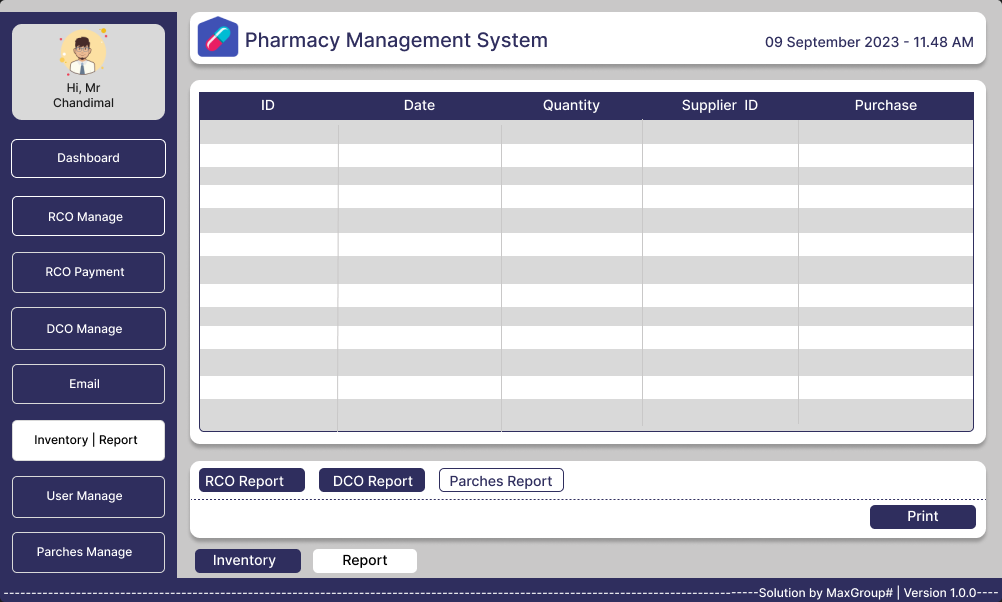


Figure 43 – Purchase Report

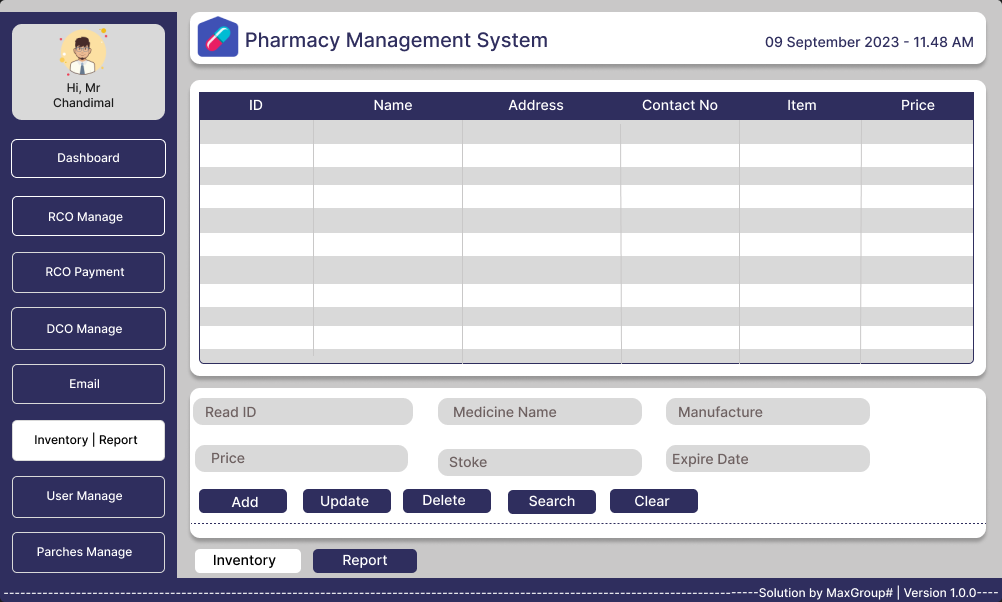


Figure 44 – Inventory

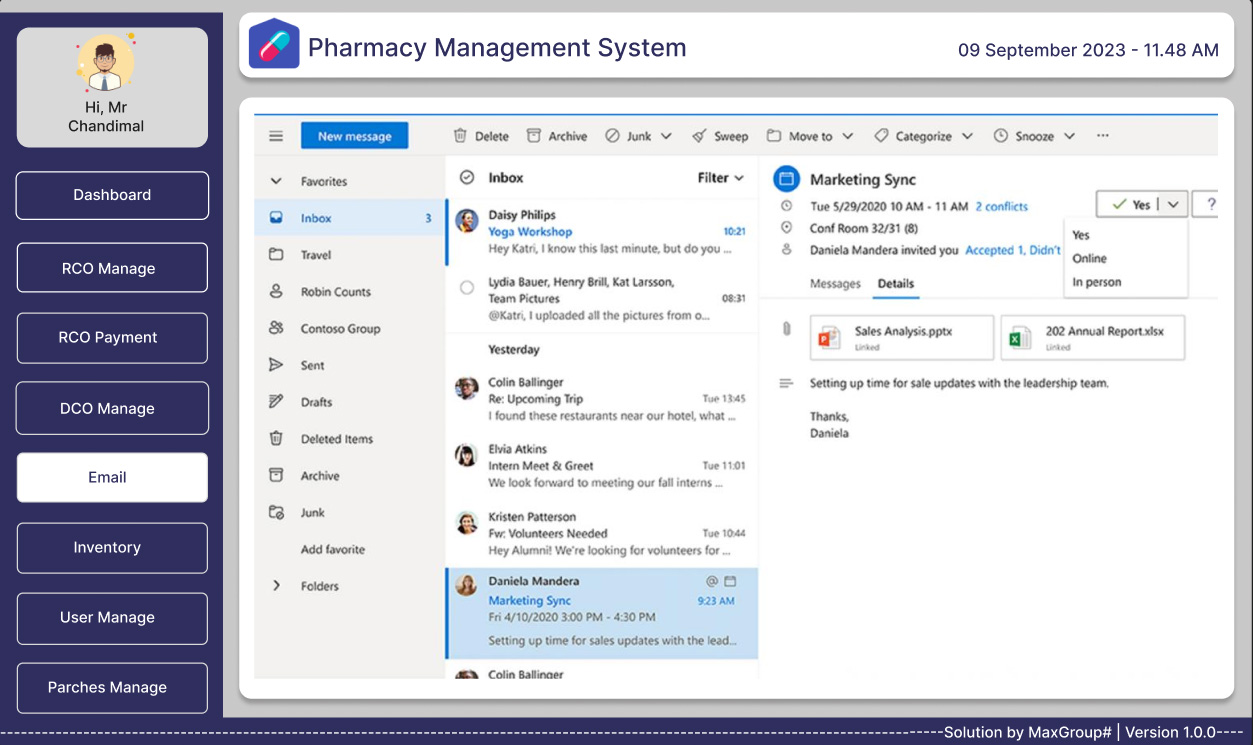


Figure 45 – Email

**3.3 Report Design**

**“Report-Design-Phase - Under Planning”**The Report Design phase is a pivotal component of our project's development process. During this phase, we will meticulously plan and define how the reports generated by our Pharmacy Shop Management Application will be structured, formatted, and presented to users. While this phase is currently in the planning stage, its significance cannot be overstated, as the quality and effectiveness of our reports directly impact the user experience and the application's overall utility.

**3.4 Test plan for the system**

**Introduction**

**Purpose**

The purpose of this test plan is to ensure the quality and reliability of the Pharmacy Management System during its development. It outlines the approach, objectives, and schedule for testing to identify and address defects and ensure that the system meets its requirements.

**Scope**

This test plan covers the testing of all modules and features of the Pharmacy Management System, including personnel management, customer management, medication inventory control, financial tracking, reporting, formal communication, and pharmaceutical distribution.

**Objectives for the test plan:**

* To verify that each module and feature of the Pharmacy Management System functions as per defined requirements.
* To identify and report defects, if any, and ensure they are addressed during development.
* To ensure that the system effectively manages staff, customers, medications, finances, and communication.

**Test Objectives**

* Verify the functionality of each module during development.
* Ensure that the system performs accurately and efficiently.
* Validate security measures to protect sensitive data.
* Ensure the system's user-friendliness and accessibility.

**Test Environment**

* Hardware Requirements

Specify hardware specifications required for development and testing.

* Software Requirements

Specify software components needed for development and testing.

* Test Data

Describe the test data sources and generation methods.

**Test Approach**

* Testing Levels

Define the levels of testing, such as unit testing, integration testing, system testing, and user acceptance testing (UAT).

* Testing Types

Specify the testing types, including functional testing, non-functional testing (performance, security, usability), and regression testing.

* Test Design Techniques

Explain the methods used to design test cases, such as equivalence partitioning, boundary value analysis, and use case-based testing.

* Test Execution Strategy

Describe the strategy for executing tests, including test cycles, iterations, and retesting.

Entry and Exit Criteria

Define criteria for entering and exiting each testing phase, including acceptance criteria for UAT.

* Test Deliverables

List the documents and artifacts to be delivered as part of testing, such as test scripts, test reports, and defect logs.

**Test Schedule**

Provide a timeline for testing, including milestones and deadlines for each testing phase.

**Test Cases**

List individual test cases for each module and feature, including detailed test steps, expected results, and pass/fail criteria.

**Risks and Contingencies**

Identify potential risks to testing and development, such as data security breaches or system performance issues, and outline mitigation strategies.

**Testing Responsibilities**

Define roles and responsibilities of the test team and stakeholders, including testers, developers, and business analysts.

**3.5 Proposed Software Engineering Methodologies**

**Introduction:**

We have chosen to adopt an agile software development methodology for the development of our Pharmacy Management System. Agile is a modern and flexible approach to software development that emphasizes collaboration, customer feedback, and incremental progress. It is particularly well-suited for projects like ours, which aim to address pressing problems and deliver user-centric solutions efficiently.

**Alignment with Agile Principles:**

The decision to use Agile is aligned with several key principles outlined in the abstract:

1. Customer-centric approach: Agile prioritizes understanding the needs of pharmacies and constantly considers customer suggestions in its development work.

2. Iterative and Incremental Development: Break the project into manageable chunks and develop incrementally in short iterations.

3. Flexibility with changing requirements: Agile adapts to changing requirements during development.

4. Efficiency and Simplification: The recurring deliveries ensure that pharmacies can realize profits faster and more smoothly.

5. User Engagement: Actively engage pharmacy staff and users to find user-friendly custom solutions. 6. Adaptive planning: Agile's dynamic plan adapts to the needs of pharmacies.

7. Quality Assurance and Continuous Improvement: Ensures quality updates and strives for continuous improvement.

In summary, our decision to use the Agile software development methodology reflects our commitment to customer satisfaction, flexibility in responding to changing needs, and a focus on delivering a user-friendly and efficient Pharmacy Management System that addresses real-world challenges faced by pharmacies. Agile's iterative and customer-centric approach is well-suited to our project's goals and objectives.

**Conclusion**

It can be stated that the introduction of the management system of online pharmacies brings many benefits to both the pharmacy and its patients. This system greatly improves operational efficiency by simplifying tasks such as inventory management, prescription processing and billing. The included automation reduces the risk of errors and ensures compliance with patient safety and medical standards. Pharmacists have access to comprehensive patient information and medication history, allowing them to provide more informed and individualized care. In addition, savings can be achieved through efficient inventory control and reduced paperwork. Regulatory compliance becomes easier and patients benefit from convenient online services such as filling prescriptions and obtaining drug information from the comfort of their homes. However, data security must be prioritized to protect sensitive patient information in this digital age. In essence, an online pharmacy management system is a game-changer that elevates pharmacy operations by improving patient experience and health outcomes.

**References**

[1]<https://www.altexsoft.com/blog/non-functional-requirements/#:~:text=Non%2Dfunctional%20requirements%20or%20NFRs,reliability%2C%20data%20integrity%2C%20etc>.