Vietnamese-German University

Faculty of Engineering

Team 3

Programming Exercise Course

Pharmacy Management System

June 2024

Nguyen Ngoc Truong	10421066
Mai Ngo Anh Khoa	10421027
Le Huu Trinh Kha	10421023
Le Thai Duong	10421012
Tran Viet Trung	10421065
Tran Cao Gia Bao	10421005
Van Anh Thu	10421105

Computer Science Program

Plagiarism Declaration

With the exception of any statement to the contrary, all the material presented in this report is the result of my own efforts. In addition, no parts of this report are copied from other sources. I understand that any evidence of plagiarism and/or the use of unacknowledged third party materials will be dealt with as a serious matter.

Signed

1. Tran Cao Gia Bao
2. Mai Ngo Anh Khoa
3. Le Huu Trinh Kha
4. Nguyen Ngoc Truong
5. Tran Viet Trung
6. Van Anh Thu
7. Le Thai Duong

Contents

List of Figures

Chapter 1

Introduction

Our Pharmacy Management System - Appotheke - is developed with the aim to provide a comprehensive application for the effective management of a pharmaceutical store. The system is designed to streamline and automate the various operations within a pharmacy, enabling pharmacists and administrators to efficiently manage the records of medicines, supplies, and personnel. The primary characteristics of the PMS comprise:

- **Inventory Management:** The system helps the pharmacist maintain detailed records of the medicines and supplies received from suppliers. This allows for efficient tracking of stock levels, reordering, and expiration dates.
- **Employee Management:** The administrative users can manage the records of all employees, including their login credentials (username and password). This ensures secure access and accountability within the organization.
- **Communication System:** The PMS incorporates a built-in messaging system, allowing employees to communicate with each other and share important information seamlessly.
- **Invoicing and Billing:** The system can generate invoices, bills, and receipts for customer transactions, streamlining the financial management of the pharmacy.

By leveraging the capabilities of the Pharmacy Management System, the pharmaceutical store can optimize its operations, improve customer satisfaction, and maintain compliance with industry regulations.

Chapter 2

Project Overview

2.1 Technology Stack

- **PHP:** The core of the Pharmacy Management System is built using the PHP programming language, a widely adopted and versatile server-side language. PHP was selected as the backend technology due to its strong suitability for web-based applications like the Pharmacy Management System. PHP provides a robust foundation for developing the necessary business logic and integrations required for the system. Additionally, the project team chose PHP because of its extensive ecosystem, built-in security functions, large and active community, and the availability of a wide range of libraries and frameworks that can accelerate the development process. The combination of PHP's widespread usage, rich feature set, and seamless integration with the other components of the technology stack made it the ideal choice for the Pharmacy Management System's backend implementation.
- **HTML, CSS, Bootstrap:** The user interface of the Pharmacy Management System is developed using HTML and CSS for structuring and styling the web pages. To enhance the user experience and ensure a consistent and responsive design, the project team has incorporated the popular Bootstrap front-end framework. Bootstrap provides a vast library of pre-built UI components, layout options, and responsive design capabilities, allowing for the rapid development of a visually appealing and user-friendly interface.
- **MySQL:** The Pharmacy Management System's data storage and retrieval needs are pow-

ered by the MySQL relational database management system (RDBMS). MySQL is a widely-used, open-source, and reliable database solution that offers robust features, scalability, and seamless integration with PHP-based applications. The project team has selected MySQL due to its widespread adoption, strong community support, and ability to handle the data requirements of the Pharmacy Management System.

- **XAMPP:** The development and testing of the Pharmacy Management System are carried out in the XAMPP environment, which is a popular cross-platform software stack package that includes the Apache web server, MySQL, PHP, and other necessary components for web application development.

2.2 Target Users

2.2.1 Pharmacy Administrators

Role and Responsibility

The Pharmacy Manager/Administrator is responsible for the overall management and operations of the pharmacy. They are tasked with inventory management, employee oversight, financial reporting, data security, and strategic decision-making. While moderately tech-savvy, they often struggle with difficulties in tracking inventory, monitoring staff productivity, generating reports, and ensuring data security.

User Need

The Pharmacy Manager/Administrator expects the Appotheke system to provide comprehensive tools for inventory control, employee management, communication, document generation, and data protection. They require streamlined inventory tracking and ordering processes, effective employee management features, efficient communication channels, automated generation of essential business documents, and robust security measures to protect sensitive customer and financial data.

2.2.2 Pharmacist

Role and Responsibility

The Pharmacist is responsible for dispensing medications, providing patient consultations, and managing the pharmacy floor. They have varying levels of comfort with technology, ranging from tech-savvy to more traditional. Pharmacists often face inefficient communication with other pharmacy roles, cumbersome record-keeping, and limited access to customer/patient information.

User Need

The Appotheke system should provide the Pharmacist with efficient medication management tools, seamless access to customer/patient records, effective communication channels to coordinate with other pharmacy staff, a user-friendly interface that integrates well with their daily workflow, and secure access to sensitive information while maintaining patient confidentiality.

2.3 Feature

2.3.1 Dashboard for Navigation

The dashboard will serve as the central hub for users to access and navigate the various functionalities of the Appotheke system. It will provide a comprehensive overview of key metrics, such as inventory levels and status, sales and revenue data, and human resource management. The dashboard will feature customizable widgets and visualization tools to enable users to quickly identify and act on important information. Intuitive navigation menus and shortcuts will allow users to seamlessly access the specific modules and tasks they need to perform.

2.3.2 User Authentication

The Appotheke system will implement a user authentication system to ensure secure access to the platform. Users with the role as Admin will be able to create the accounts for the Pharmacists, with options for different user roles (Pharmacy Manager (Admin) and Pharmacist). The login process will include password-based authentication, and user accounts will be managed centrally, allowing administrators to control access privileges and monitor user activities.

2.3.3 User Authorization

The Appotheke system will incorporate a role-based access control (RBAC) mechanism to manage user permissions and responsibilities. Different user roles (Pharmacy Manager and Pharmacist) will be defined, each with a specific set of permissions and access rights. This will ensure that users can only perform the tasks and access the data that are relevant to their assigned role, enhancing data security and preventing unauthorized actions.

2.3.4 Data Interaction Functions

- **CRUD data function:** The Appotheke system will provide users with a comprehensive CRUD (Create, Read, Update, Delete) functionality to manage various data entities within the application. This will include the ability to add new medication items to the inventory, including details such as drug id, name, price, type, quantity, expiry date, and supplier information. Users will be able to manage stock levels, track expiration dates, and generate alerts for low inventory or expiring medications. Additionally, the system will allow users to add and maintain information about the pharmacy's suppliers, including id, contact details, and name. The CRUD function will be tightly integrated with the inventory management system, streamlining the ordering and fulfillment process.
- **Chat:** The Appotheke system will incorporate a secure, internal chat feature to facilitate communication between different pharmacy roles (Pharmacy Manager and Pharmacist). Users will be able to send messages, and engage in real-time discussions to coordinate tasks, resolve issues, and provide updates, enabling efficient collaboration and information sharing within the pharmacy.

Chapter 3

Project Management

3.1 Team Roles and Responsibilities

Our project team consisted of the following members with their respective roles and responsibilities based on their expertise and the project's needs:

- Product Owner (Leader) - Nguyen Ngoc Truong (10421066): Primarily focused on the frontend development and serving as the project's Product Owner, leading the team and facilitating the project's direction.
- Database Administrator - Le Huu Trinh Kha (10421023) & Mai Ngo Anh Khoa (10421027): Responsible for the design and implementation of the project's database, including the data schema and modeling.
- Backend Developer - Le Thai Duong (10421012) & Tran Viet Trung (10421065) & Van Anh Thu (10421105): Responsible for the backend development, including the implementation of the core business logic, APIs, and integration with the frontend.
- Front End Developer - Tran Cao Gia Bao (10421005) & Nguyen Ngoc Truong (10421066): Responsible for the frontend development, collaborating with the Product Owner to ensure the user interface and experience meet the requirements.

3.2 Project Timeline

The project was executed over a 15-week timeline, with the following key milestones and activities:

- Week 1: Introduction, Formulate Team, Choose Project
- Week 2: Analyzing requirements and project proposal, Team Division, Requirement Engineering, Set timeline, List Use Case
- Week 3: Analyzing requirements and use case, Draw Use Case diagrams, Research
- Week 4: Draw Sequence diagram, Fabricate UI using Figma, Set the project's architecture
- Week 5: Build data schema, Modeling data, Initializing in building a database
- Week 6: Draw component and deployment diagrams, Start the documentation process
- Week 7: Initialize frontend Implementation, Choose Front-end Framework
- Week 8: Continuing Front-end development
- Week 9: Initialize backend Implementation, Choose backend framework
- Week 10: Login-signup system
- Week 11: Pending due to errors in the system
- Week 12: Authentication and authorization testing
- Week 13: Business logic - CRUD functionality, Chat system
- Week 14: Fixing errors, Starting report document
- Week 15: Fixing errors, Deployment

Chapter 4

Requirement Analysis

4.1 Functional Requirements

4.1.1 Management Role (Admin)

User Management

- Create, edit, and delete user accounts.
- Assign roles and permissions to users.

Inventory Management

- Add, edit and remove medicines/drugs and consumables.
- Track stock levels and set reorder thresholds.
- Generate and review inventory reports.

Supplier Management

- Add, edit and remove supplier information.
- Record and manage supplies sent in by suppliers.
- Generate and review supplier-related reports.

Employee Management

- Manage employee records(personal details).

- Track employee performance and attendance.
- Generate employee report.

Financial Management

- Generate and manage invoices, bills and receipts.
- Track sales and revenue.
- Generate financial reports.

Messaging System

- Send and receive messages to/from employees.
- Manage and archive messages.

4.1.2 Pharmacist Role

Inventory Viewing

- View medicines/drugs and consumables inventory.
- Check stock levels and availability.

Sales and Billing

- Generate invoices and bills for customers.

Messaging System

- Send and receive messages to/from management and other employees.

4.2 Non-Functional Requirements

Performance

- The system should handle multiple concurrent users without significant performance degradation.
- Inventory updates should reflect in real-time.

Security

- Implement authentication and authorization mechanisms to ensure only authorized access.
- Encrypt sensitive data such as passwords.
- Validate user's input data to prevent SQL injection.

Usability

- User-friendly interface with clear navigation and instructions.

Reliability

- Ensure the system is available 99.9 of the time.
- Implement failover mechanisms and data redundancy.

Scalability

- The system can be scaled to accommodate a growing number of users and inventory items.
- Support for future function enhancements and integration with other services.

Maintainability

- Code should be well-documented and modular to facilitate easy maintenance and updates.
- Provide logging and monitoring to detect and diagnose issues.

Compliance

- Ensure the system complies with relevant regulations and standards for pharmaceutical management and data protection.
- Regular audits to ensure compliance.

Backup and Recovery

- Implement automated backup procedures to prevent data loss.
- Provide a recovery plan in case of system failure.

Chapter 5

Software Design

5.1 System Architecture

The system architecture of the Pharmacy Management System is designed to ensure scalability, robustness, and ease of maintenance. The architecture follows a multi-tier approach, consisting of the following layers:

- **Presentation Layer:** This layer includes the user interface components, facilitating interaction between the users and the system. The technologies used in this layer include HTML, CSS, JavaScript, and a server-side language like PHP.
- **Application Layer:** This layer contains the business logic and functionalities of the system, implemented using PHP.
- **Data Layer:** This layer manages data storage and retrieval, using a relational database management system (RDBMS) like MySQL. It includes database schemas, tables, and relationships.

The system architecture follows a well-structured approach, with clear separation of concerns between the presentation, application, and data layers. This design ensures modularity, flexibility, and ease of maintenance, allowing the system to scale and adapt to future requirements.

5.2 System Interaction

5.2.1 Usecase diagrams

1. **General System:**

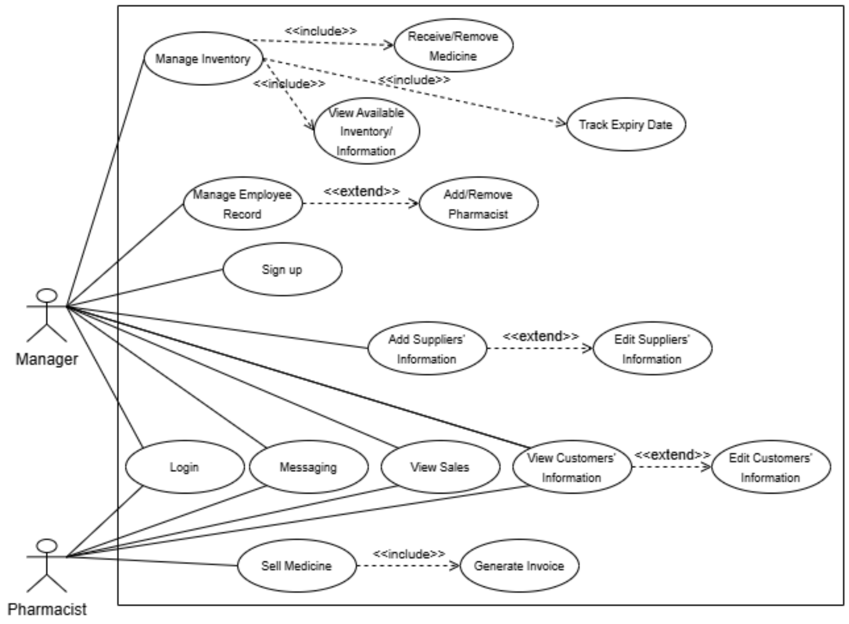


Figure 5.1: General System

The use case diagram illustrates the interactions and key roles within a Pharmacy Management System, describing the responsibilities of the Manager and Pharmacist actors.

The Manager has broad range of responsibilities within the system. These include managing inventory, maintaining employee and supplier records, and user management. The Manager can receive and remove medicine, track expiry dates, and view available inventory information, all of which are encapsulated within the "Manage Inventory" use case. Additionally, the Manager can sign up new users, and add or remove pharmacists from the system. When managing employee records, the Manager ensures that the staff information is up to date. The tasks "Add Suppliers' Information" and "Edit Suppliers' Information" allow the Manager to maintain accurate supplier records, which is essential for maintaining stock levels and ensuring the availability of medicines. The manager can also perform daily activities similar to pharmacists, such as logging in to the system, engaging in messaging, viewing sales data, and handling customer-related tasks.

The Pharmacist's role focuses more on day-to-day operations within the pharmacy. They can log in to the system, engage in messaging, view sales data, and handle customer-related tasks. The use case ”Sell Medicine” includes generating invoices for sales transactions, ensuring that each sale is accurately recorded and customers receive proper documentation for their purchases. Additionally, the Pharmacist can view and edit customer information, allowing for efficient customer service and personalized care. This includes the ability to update records, which may be necessary for maintaining accurate customer histories and ensuring compliance with legal requirements.

The system integrates various functionalities to support the roles of both the Manager and the Pharmacist, ensuring seamless operations within the pharmacy. The inclusion relationships (≪include≫) indicate that certain actions are part of broader tasks, highlighting the system's modularity. The extend relationships (≪extend≫) show optional but related activities that enhance primary functions.

The use case diagram effectively maps out the critical interactions and processes within a Pharmacy Management System, showcasing how Managers and Pharmacists engage with the system to ensure efficient and effective pharmacy operations. By detailing the specific tasks and their relationships, the diagram provides a clear overview of the system's functionality and the collaborative efforts required to maintain a well-functioning pharmacy.

2. **Create Account:**

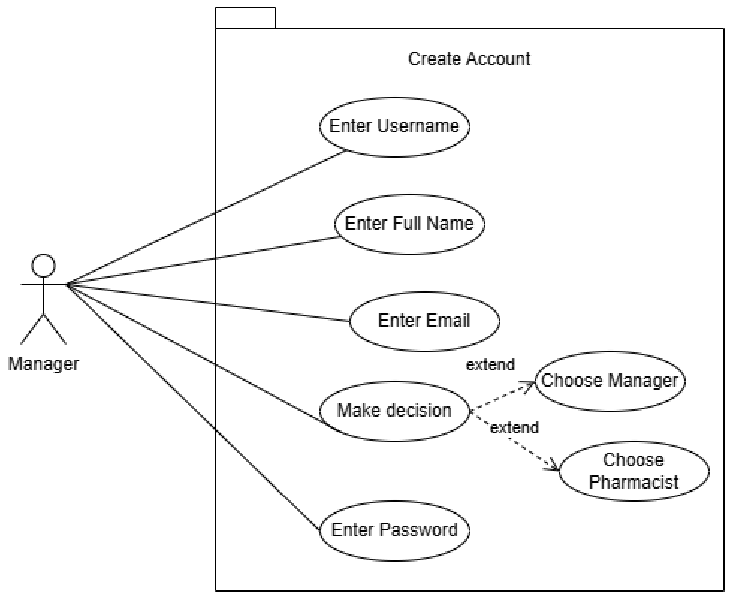


Figure 5.2: Create Account

The use case diagram for "Create Account" in the Pharmacy Management System outlines the detailed steps a Manager takes to register a new user. Initially, the Manager inputs essential information such as the username, full name, email, and password to establish the user's profile. Following this, the Manager must make a decision regarding the user's role, which extends to selecting either "Manager" or "Pharmacist." This role assignment is crucial for defining the user's access permissions and responsibilities within the system. By specifying these steps, the diagram emphasizes the structured approach to creating secure, role-specific user accounts, ensuring that each new user is appropriately authenticated and authorized for their designated tasks in the pharmacy management environment.

3. **Login:**

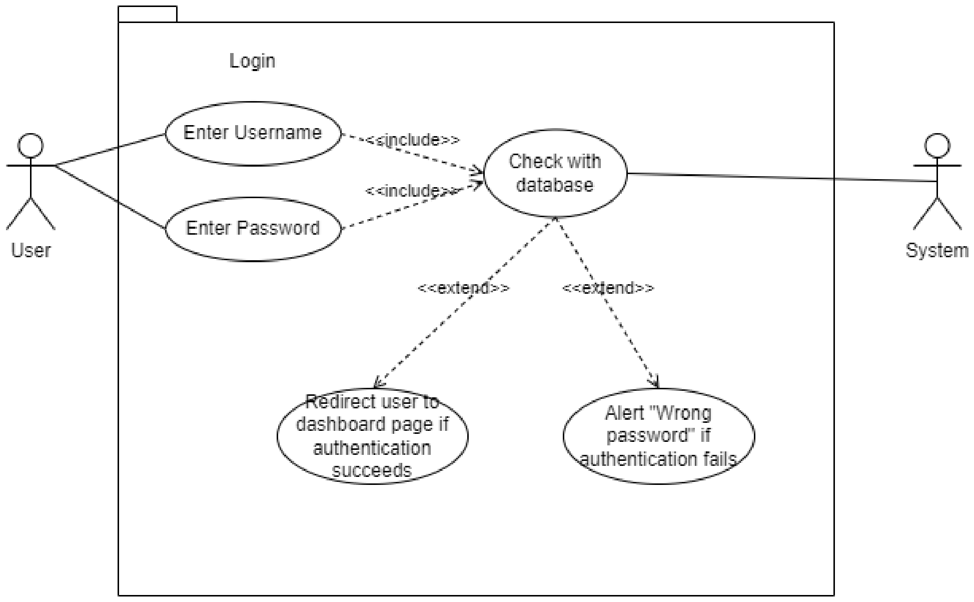

Login
Enter Username
Enter Password
<<include>>
<<include>>
Check with database
User
System
<<extend>>
<<extend>>
Redirect user to dashboard page if authentication succeeds
Alert "Wrong password" if authentication fails

Figure 5.3: Login

The use case diagram for the "Login" process in the Pharmacy Management System shows a user entering their username and password to gain access. This interaction is fundamental for system security, ensuring that only authorized users can log in. By requiring both credentials, the system enforces authentication, protecting sensitive information and maintaining the integrity of the system.

4. **Show Expired Medicine:**

The "Expired Medicine Tracker" use case diagram for the Pharmacy Management System outlines the interactions between a manager and the system for managing expired medicines. The manager is responsible for overseeing the entire process of tracking and

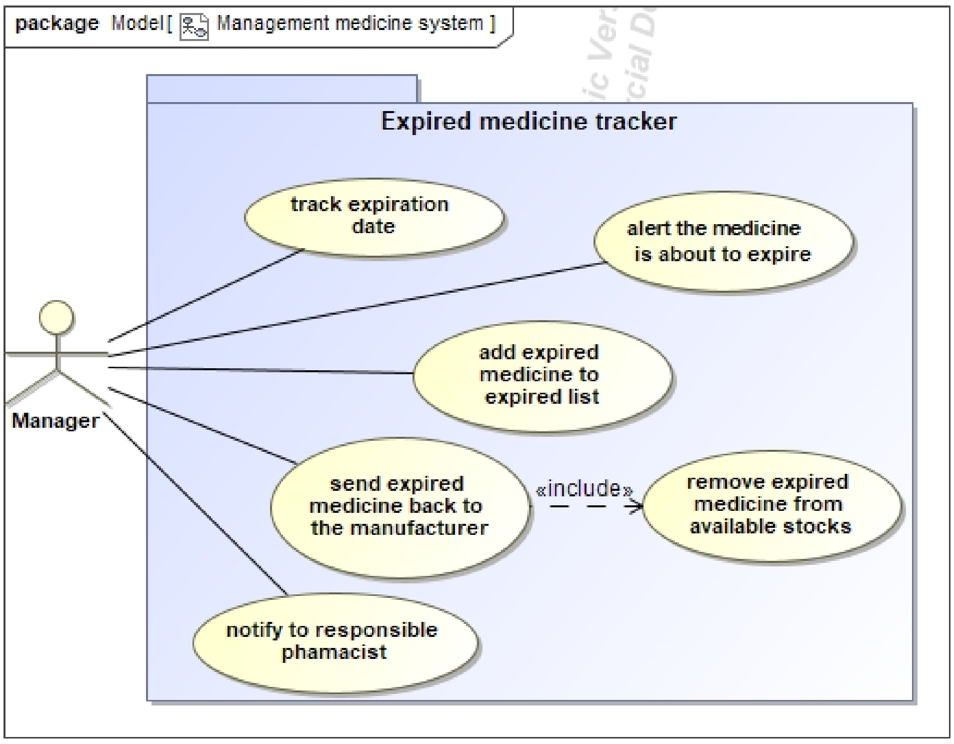

package Model [Management medicine system]
Expired medicine tracker
track expiration date
alert the medicine is about to expire
add expired medicine to expired list
send expired medicine back to the manufacturer
«include»
remove expired medicine from available stocks
notify to responsible phamacist
Manager

Figure 5.4: Show Expired Medicine

handling expired medicines to ensure that the pharmacy's stock remains up-to-date and safe for consumers.

The process begins with tracking the expiration dates of medicines. This proactive step allows the system to alert the manager when a medicine is approaching its expiration date. This alert is crucial for timely action to prevent the distribution of expired medicines.

When a medicine is identified as expired, it is added to the expired list. This action includes the subsequent step of removing the expired medicine from the available stocks, ensuring that it is no longer available for sale or distribution. The removal process is seamlessly integrated into the system's workflow, maintaining the accuracy of the pharmacy's inventory.

The system also facilitates the return of expired medicines to the manufacturer. This step is essential for proper disposal and potential credit or replacement from the manufacturer. Alongside this, the manager is responsible for notifying the responsible pharmacist about the expired medicines, ensuring that all relevant parties are aware and can take appropriate action.

In summary, the "Expired Medicine Tracker" use case diagram demonstrates a comprehensive approach to managing expired medicines within the Pharmacy Management System. By incorporating tracking, alerting, listing, removing, returning, and notifying processes, the system supports the manager in maintaining a safe and efficient pharmacy operation.

5. **Manage Medicine Inventory:**

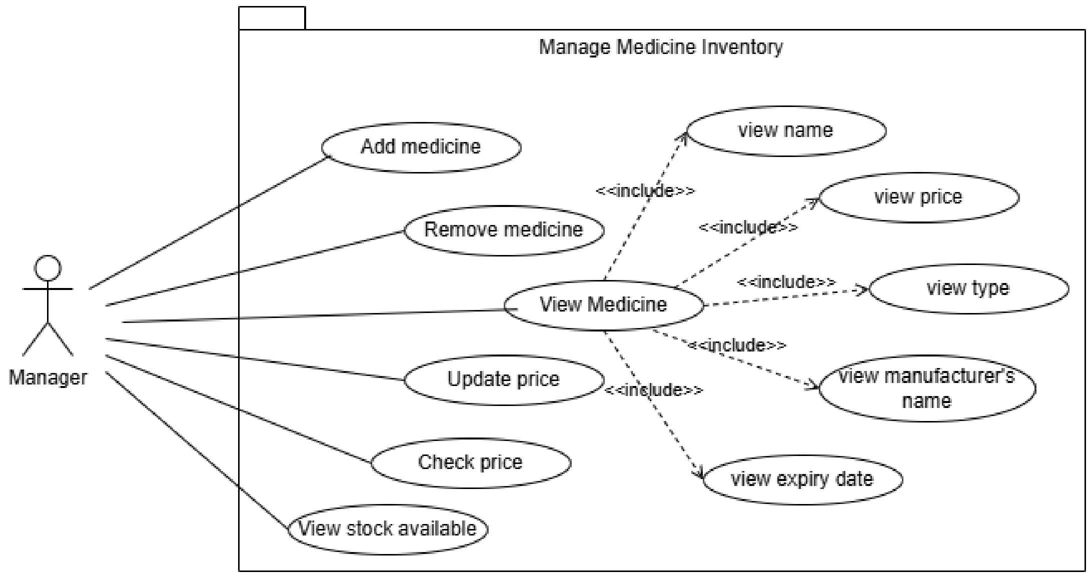


Figure 5.5: Manage Medicine Inventory

The use case diagram illustrates various interactions between the Manager and the system to manage the medicine inventory

The Manager has the capability to add new medicines to the inventory, ensuring that the latest and necessary medications are always available for customers. In addition, the Manager can remove outdated or unnecessary medicines, which helps in maintaining a streamlined and up-to-date inventory. Another crucial function is updating the prices of medicines, which is vital for reflecting current market rates and ensuring competitive pricing.

Central to the inventory management process is the 'View Medicine' use case. This allows the Manager to access detailed information about each medicine in the inventory. Through this function, the Manager can view the medicine's name, price, type, manufacturer's name, and expiry date. This ensures that the Manager has all the necessary information to make decisions about stocking and selling medicines.

Furthermore, the Manager can check the current prices of medicines and view the available stock levels. These functions are essential for maintaining an efficient inventory, preventing stockouts, and ensuring that customers have access to the medicines they need. The diagram effectively captures these interactions, illustrating how each action is interconnected and contributes to the overall management of the pharmacy's medicine inventory.

6. **View Medicine:**

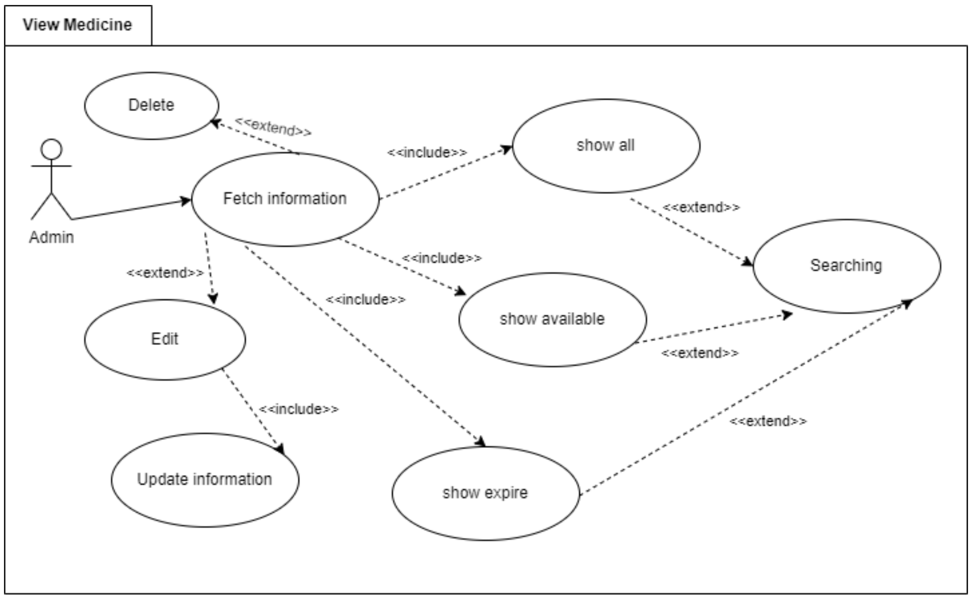


Figure 5.6: View Medicine

The "View Medicine" use case diagram outlines the interactions and functionalities available to the Admin for managing medicine information. The central use case is "Fetch Information," which acts as the primary function enabling the Admin to retrieve various data related to the medicines in the inventory.

From "Fetch Information," the Admin can extend its functionalities to include showing all medicines in the system. This comprehensive view helps the Admin get an overview of the entire inventory. Additionally, the Admin can perform searches, enhancing the system's usability by allowing specific queries based on different criteria.

Within the "Fetch Information" use case, there are several inclusions that detail the specific aspects of medicine data that can be retrieved. The Admin can "Show Available" medicines, providing a filtered view of only those medicines currently in stock. Similarly, the Admin can "Show Expire," which highlights the medicines that are nearing or have passed their expiration date, crucial for maintaining safe and effective inventory.

The diagram also includes the capability to "Update Information," ensuring that the medicine details are kept current. This functionality is critical for reflecting changes such as price updates or new stock information. The Admin can also ”Edit” existing entries, allowing for corrections or modifications to the medicine data. Extending from the Edit function, the Admin can ”Delete” outdated or incorrect entries, helping to maintain a clean and accurate database.

Overall, this use case diagram effectively captures the various operations available to the Admin for managing and viewing medicine information within the pharmacy's system. It shows the interconnectedness of these functions and how they contribute to efficient inventory management.

7. **Employees Management:**

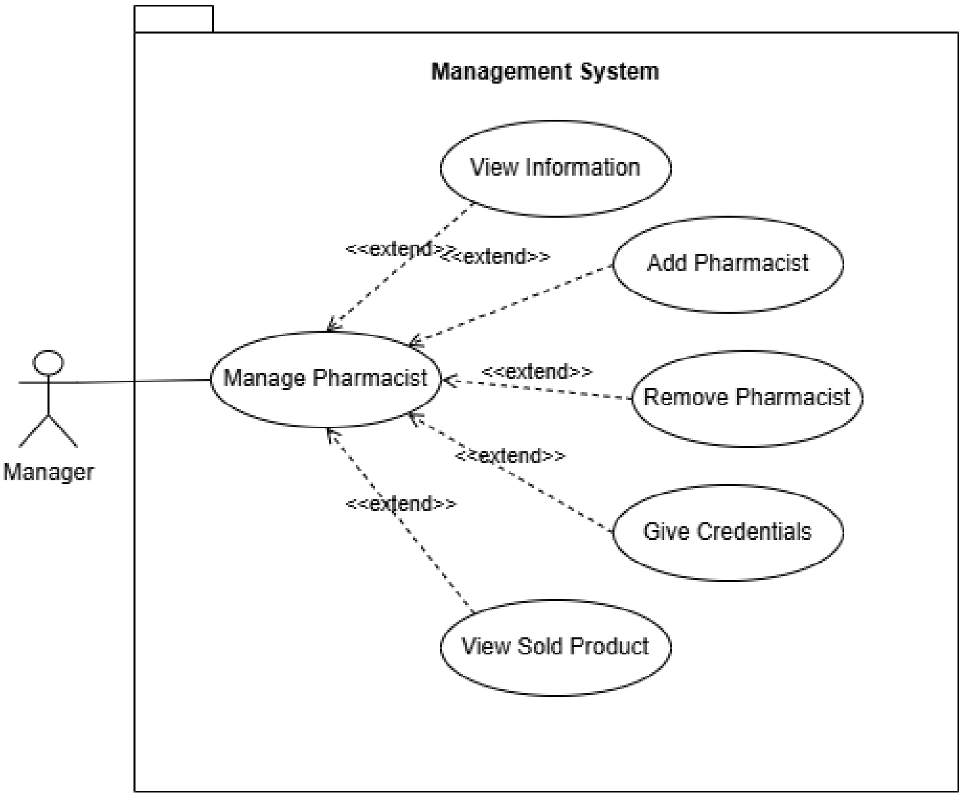


Figure 5.7: Employees Management

The use case diagram for the Pharmacy Management System highlights how the manager interacts with the system to manage pharmacists effectively. The central use case, ”Manage Pharmacist,” includes actions like adding or removing pharmacists and providing credentials. These actions ensure that only manager can manage pharmacy operations and maintain security by controlling access.

Additionally, the manager can view information about pharmacists and sold products, allowing for informed decision-making and efficient inventory management. Overall, the diagram illustrates the essential functions that the manager performs to ensure smooth pharmacy operations and data security.

8. **Client's Order and Invoice Generation:**

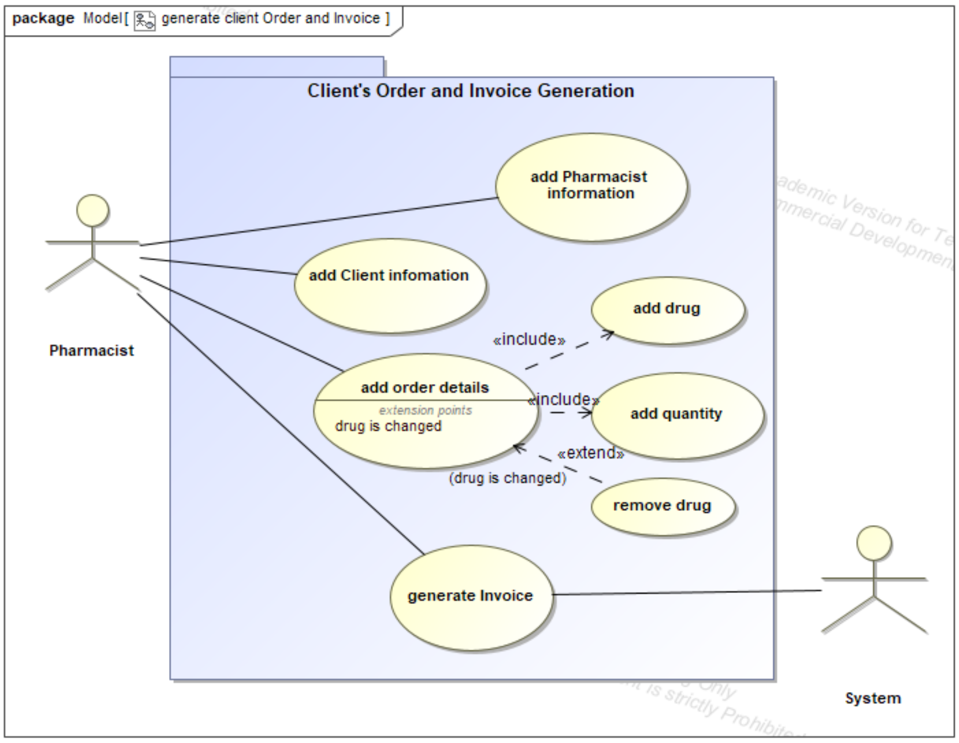


Figure 5.8: Client's Order and Invoice Generation

The use case diagram for "Client's Order and Invoice Generation" outlines the process through which a Pharmacist interacts with the Pharmacy Management System to manage client orders and generate invoices. This interaction begins with the Pharmacist adding client information, followed by entering order details, which include specific tasks like adding the pharmacist's information, adding drugs, and specifying the quantity. These tasks are interconnected through ≪include≫ relationships, indicating that each is a necessary component of completing the order details.

Additionally, the "add order details" use case includes an extension point for when a drug is changed, which extends to "remove drug" to handle modifications or corrections in the order. This ensures that any updates to the order are accurately reflected before proceeding further.

Once all necessary details have been entered and verified, the Pharmacist can proceed to "generate Invoice" which finalizes the transaction and creates a record for billing purposes. The System, represented as a secondary actor, interacts with the Pharmacist to ensure that all information is processed correctly, maintaining an accurate and efficient workflow from order placement to invoice generation. This diagram clearly illustrates the structured process of handling client orders and generating invoices, ensuring that all critical information is captured and managed effectively.

9. **Add Medicine:**

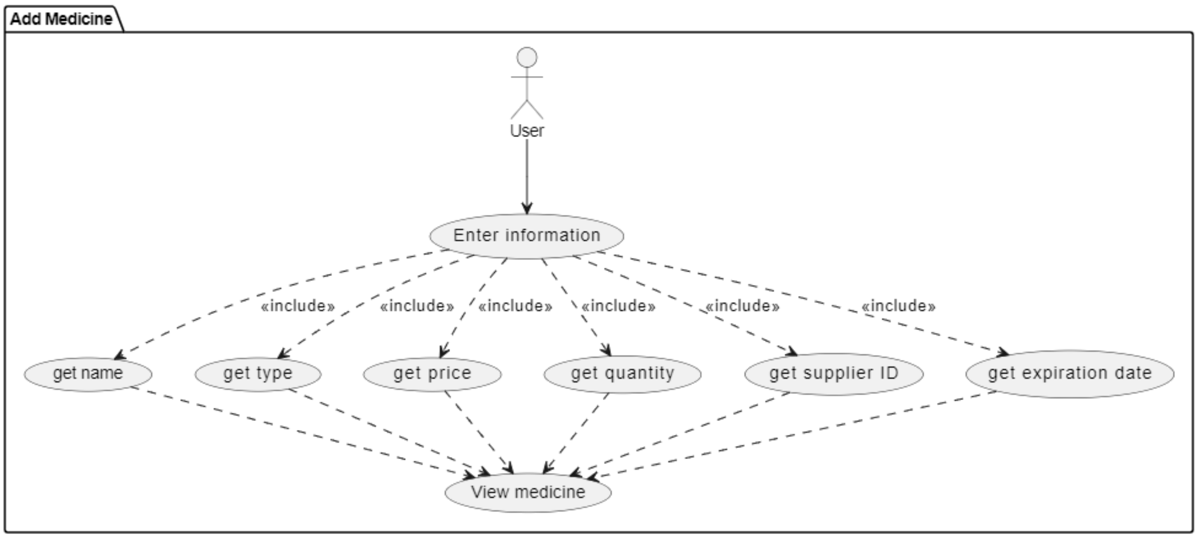


Figure 5.9: Add Medicine

The "Add Medicine" use case diagram illustrates how a user interacts with the system to enter detailed information about new medicines. The main action, "Enter Information" involves several steps to ensure that all relevant data is accurately recorded.

The user provides key details, such as the medicine's name, type, price, quantity, supplier ID, and expiration date. Each of these elements is crucial for maintaining accurate inventory and ensuring that the pharmacy operates efficiently.

Once the information is entered, the system includes it in the "View Medicine" section, allowing users to easily access and manage the medicine records. This structured approach ensures that the pharmacy's inventory is up-to-date, facilitating effective management and tracking of medications.

10. **Add Suppliers:**

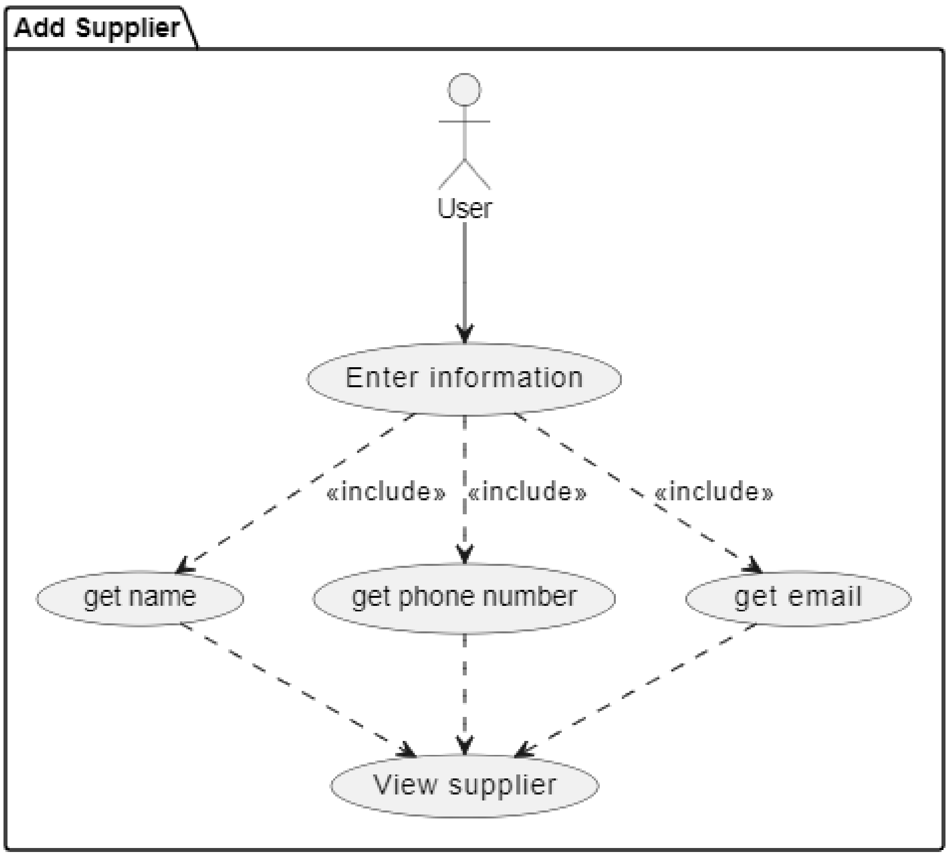


Figure 5.10: Add Supplier

The "Add Supplier" use case diagram illustrates the process by which a user adds new supplier information to the system. The primary action, "Enter Information" involves capturing essential details about the supplier, including name, phone number, and email.

Each of these details is included to ensure comprehensive supplier records, which are crucial for effective communication and inventory management. Once the information is entered, it can be viewed in the "View Supplier" section, allowing users to easily access and manage supplier data.

This process supports efficient supplier management, helping to maintain up-to-date records and fostering smooth business operations. Overall, the diagram highlights a straightforward approach to adding and managing supplier information within the system.

11. **View Suppliers**

The "View Supplier" use case diagram outlines how an admin manages supplier information. The central action, "Fetch Information" allows the admin to access supplier details. This is extended by actions such as "Delete" for removing outdated records; "Edit" for

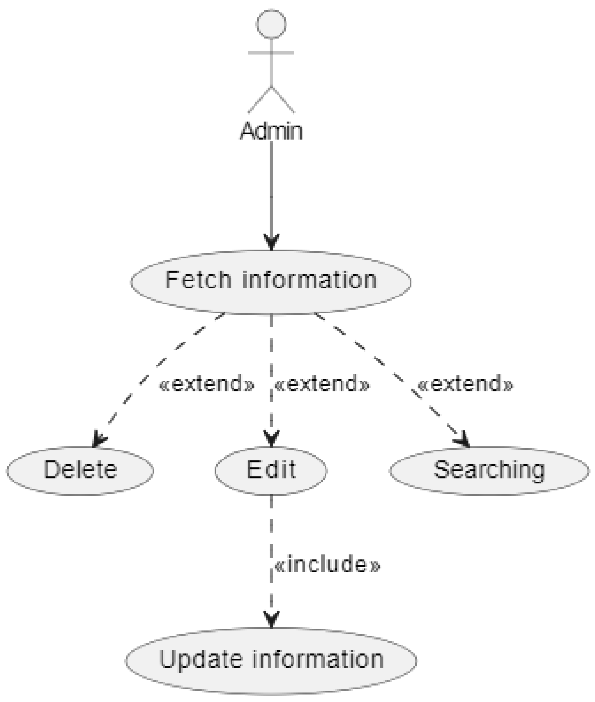


Figure 5.11: View Supplier

updating existing information; and "Searching" to quickly find specific suppliers. After editing, the "Update Information" step ensures that all changes are accurately saved. This process helps maintain up-to-date and accurate supplier records, supporting effective management and decision-making.

12. **Sale Report:**

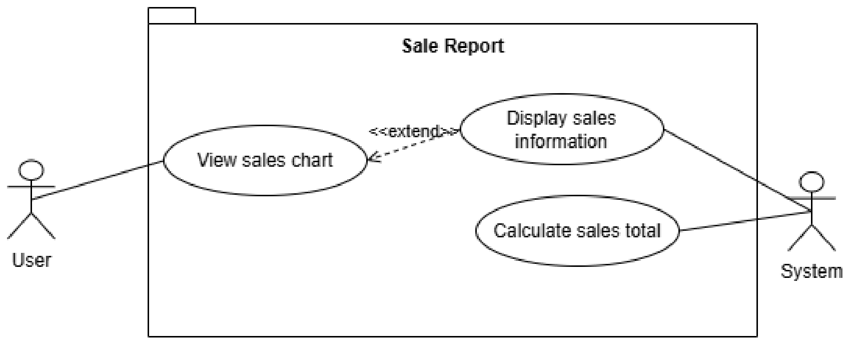


Figure 5.12: Sale Report

The "Sale Report" use case diagram outlines how users interact with the system to view and analyze sales data. The primary action, "View Sales Chart" allows users to access a visual representation of sales performance. This action is extended by "Display Sales Information" which provides detailed insights into specific sales figures.

The system plays a critical role by calculating the total sales, ensuring that users have accurate data to inform their decisions. This calculation is essential for generating comprehensive reports that reflect overall sales performance.

By displaying detailed sales information, users can assess trends, monitor performance, and make data-driven decisions to improve business operations.

13. **Messaging System:**

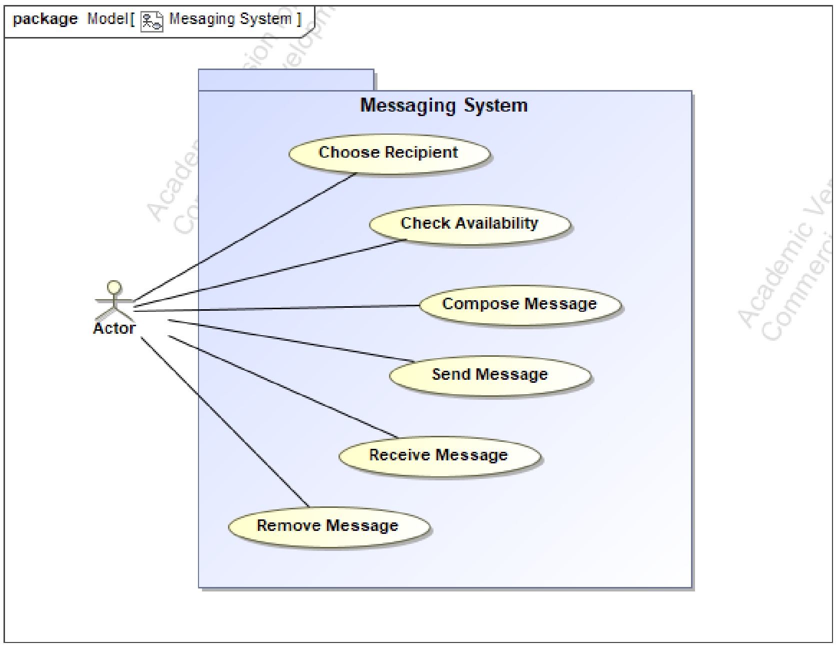


Figure 5.13: Messaging System

The use case diagram for the "Messaging System" outlines the various interactions an Actor (user) can have within the system, presenting a comprehensive set of functionalities to efficiently manage messaging needs.

Firstly, the "Choose Recipient" use case allows the Actor to select the recipient of the message, ensuring that the message is directed to the correct individual or group. This is followed by the "Check Availability" use case, where the Actor can determine if the recipient is available to receive messages.

The "Compose Message" use case involves the Actor writing the content of the message, including entering the subject, body, and any attachments or multimedia elements. Once the message is composed, the "Send Message" use case covers the actual transmission of the message through the messaging system.

On the receiving end, the ”Receive Message” functionality allows the Actor to retrieve messages sent by others, displaying them in the inbox and providing notifications to the Actor. Lastly, the ”Remove Message” use case enables the Actor to delete unwanted or obsolete messages from their inbox.

Overall, this diagram effectively encapsulates the core functions of a messaging system, emphasizing a user-centric approach to handling both sending and receiving messages. Each use case ensures that the messaging process is streamlined and user-friendly, including all the necessary steps from selecting recipients to managing received messages.

5.2.2 Sequence Diagrams

1. **Sign up:**

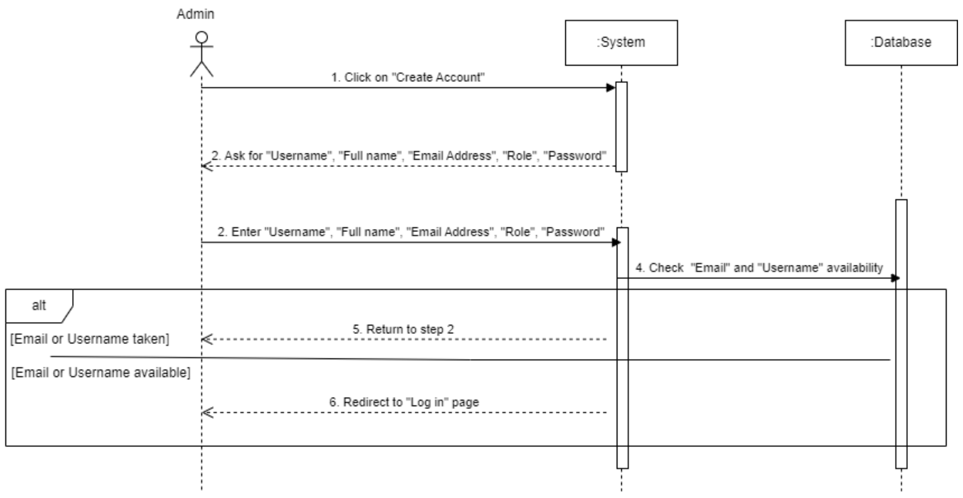

Admin
:System
:Database
1. Click on "Create Account"
2. Ask for "Username", "Full name", "Email Address", "Role", "Password"
2. Enter "Username", "Full name", "Email Address", "Role", "Password"
4. Check "Email" and "Username" availability
alt
[Email or Username taken]
5. Return to step 2
[Email or Username available]
6. Redirect to "Log in" page

Figure 5.14: Sign up

The sequence diagram illustrates the process for an Admin to create a new account in the system. The process begins when the Admin clicks on the ”Create Account” button. The System responds by asking for the required account details, including ”Username”, ”Full name”, ”Email Address”, ”Role” and ”Password”

The Admin then enters these details into the System. The System checks the ”Email” and ”Username” for availability by querying the Database. If either the email or username is already taken, the System prompts the Admin to return to the step of entering account details, allowing them to correct or choose different information.

If the email and username are available, the System redirects the Admin to the login page, completing the account creation process. This sequence ensures that new accounts are created with unique identifiers, maintaining the integrity of the user database and facilitating a smooth registration experience.

2. **Login:**

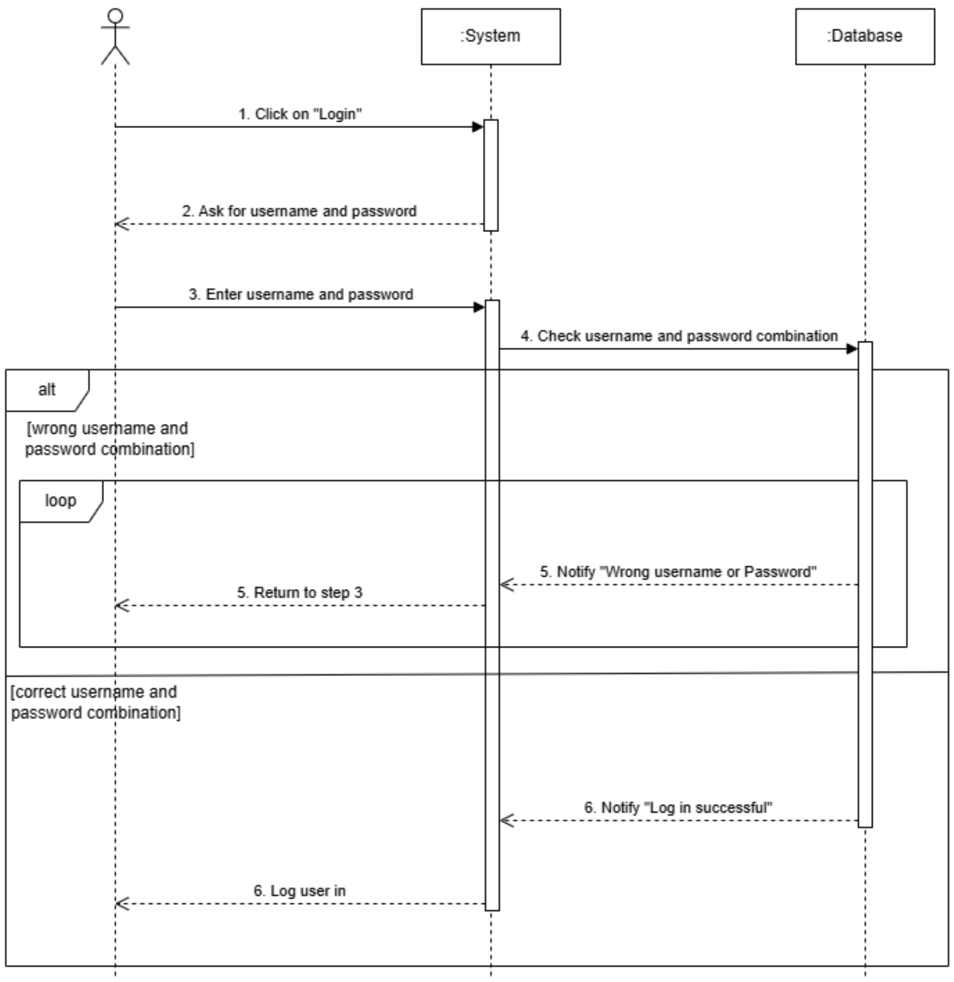

:System
:Database
1. Click on "Login"
2. Ask for username and password
3. Enter username and password
4. Check username and password combination
alt
[wrong username and password combination]
loop
5. Notify "Wrong username or Password"
5. Return to step 3
[correct username and password combination]
6. Notify "Log in successful"
6. Log user in

Figure 5.15: Login

The sequence diagram details the login process for a user in the system, ensures that users are properly authenticated before accessing the system. The interaction begins when the user clicks on the "Login" button. The System then prompts the user to enter their username and password. After the user provides their credentials, the System checks the username and password combination against the Database.

If the username and password combination is incorrect, the System notifies the user with a message indicating "Wrong username or password." This triggers a loop where the user is returned to the step of entering their username and password, allowing them to attempt login again.

If the username and password combination is correct, the System notifies the user that the login is successful. Subsequently, the user is logged into the system.

3. **Show Expired Medicine:**

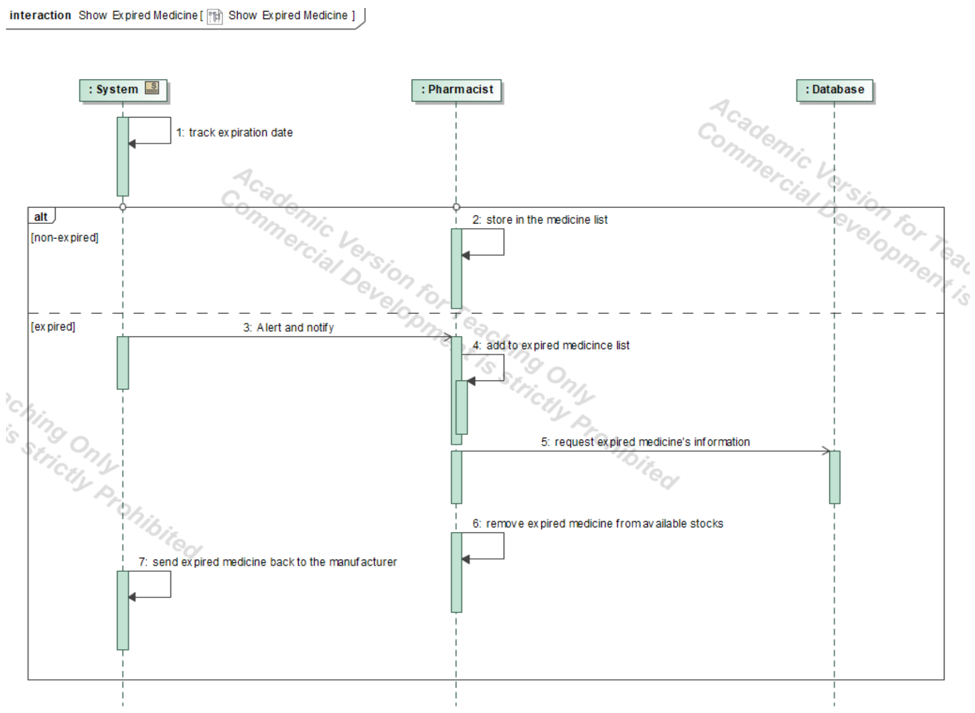

interaction Show Expired Medicine [Show Expired Medicine]
: System
: Pharmacist
: Database
1: track expiration date
alt
[non-expired]
2: store in the medicine list
[expired]
3: Alert and notify
4: add to expired medicine list
5: request expired medicine's information
6: remove expired medicine from available stocks
7: send expired medicine back to the manufacturer

Figure 5.16: Show Expired Medicine

The sequence diagram outlines the process for managing expired medicines within the system. The process starts with the System tracking the expiration date of the medicines. If the medicine is not expired, it is stored in the medicine list by the Pharmacist.

If the medicine is expired, the System alerts and notifies the Pharmacist. The Pharmacist then adds the medicine to the expired medicine list. Following this, the Pharmacist requests the expired medicine's information from the Database. Once the information is retrieved, the Pharmacist removes the expired medicine from the available stocks.

Finally, the System facilitates the process of sending the expired medicine back to the manufacturer. This sequence ensures that expired medicines are effectively identified, removed from inventory, and appropriately handled, maintaining the safety and efficacy of the available medicine stock.

4. **View Medicine:**

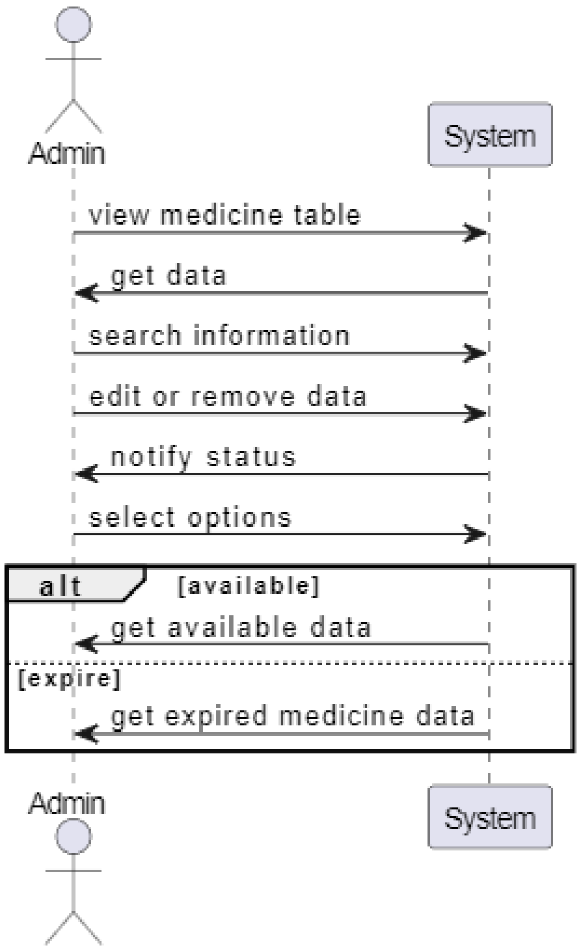


Figure 5.17: View Medicine

The sequence diagram illustrates the interaction between an Admin and the System for managing medicine data. The process begins with the Admin viewing the medicine table in the System. The System retrieves and provides the requested data to the Admin.

Next, the Admin searches for specific information within the medicine data. The System processes this request and returns the relevant information. The Admin can then choose to edit or remove the data as needed, with the System updating the medicine table accordingly and notifying the Admin of the status of the operation.

The Admin is presented with various options to select from for further actions. Based on the selected options, the process diverges into two alternate flows. If the Admin requests available data, the System retrieves and provides the available medicine data. Alternatively, if the Admin requests data on expired medicines, the System retrieves and provides the expired medicine data.

This sequence ensures that the Admin can efficiently manage and maintain accurate medicine records, including viewing, searching, editing, and removing data, while also handling specific requests for available or expired medicine information.

5. **Order and Invoice Generation:**

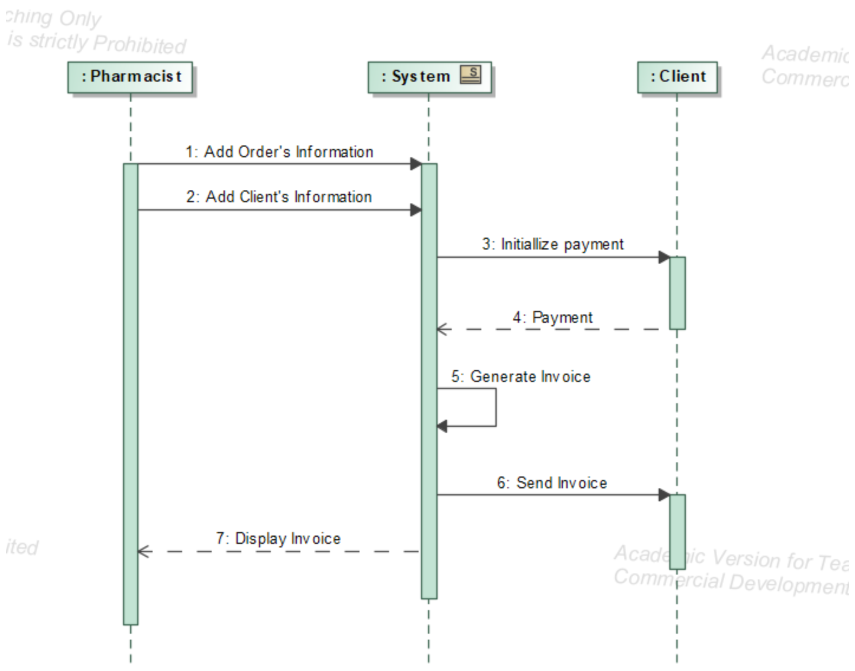


Figure 5.18: Order and Invoice Generation

The sequence diagram illustrates the interaction between the pharmacist, the system, and the client in the process of order and payment. Initially, the pharmacist inputs the order and client information into the system. Following this, the system initiates the payment process with the client, who then completes the payment. Once the payment is confirmed, the system generates an invoice and sends it to the client. Finally, the invoice is displayed to the pharmacist, completing the transaction. This sequence ensures a smooth flow of information and efficient transaction handling within the pharmacy management system.

6. **Add Medicine:**

The sequence diagram demonstrates the interaction between suppliers, users, and the system in the process of managing medicine deliveries. Initially, the supplier delivers medicine to the pharmacy. The user then enters the medicine's data into the system, prompting the system to retrieve relevant information. Once the necessary details are

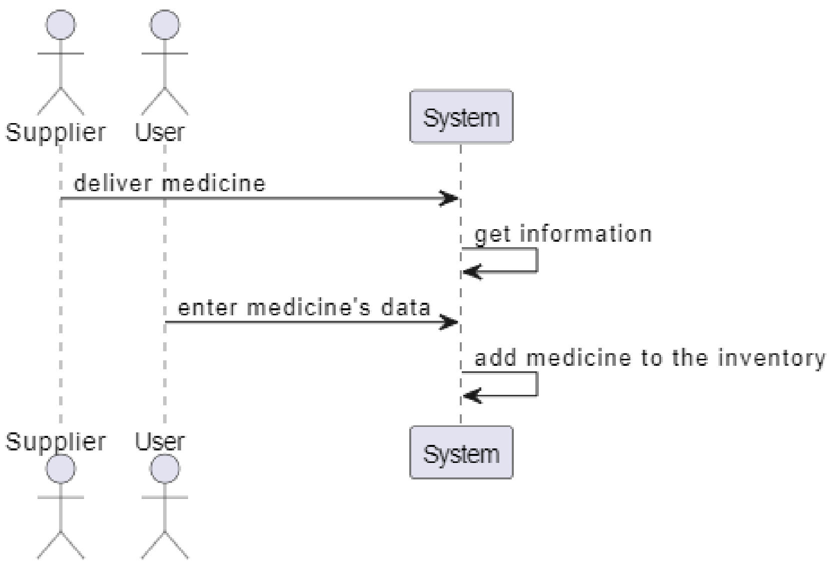


Figure 5.19: Add Medicine

confirmed, the system adds the medicine to the inventory. This process ensures that the inventory remains accurate and up-to-date.

7. **Add Suppliers:**

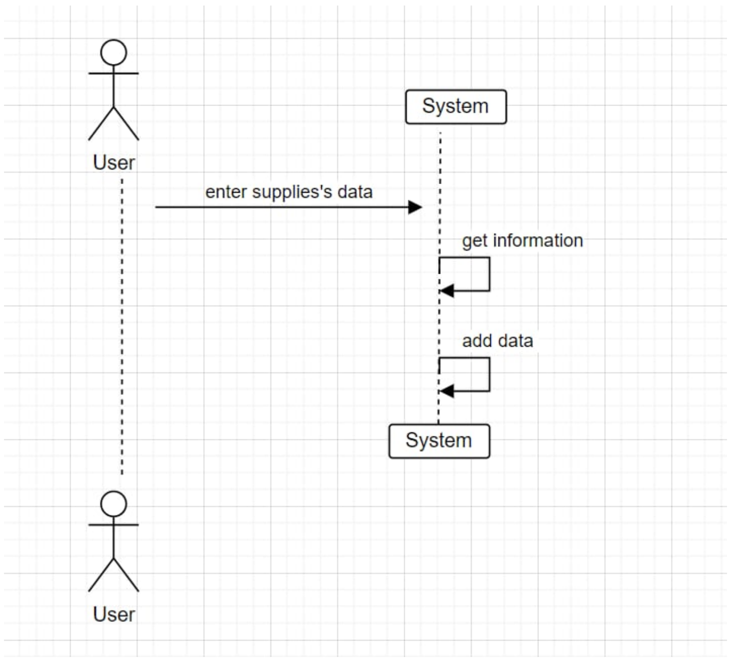


Figure 5.20: Add Suppliers

The sequence diagram describes the interaction between a user and a system. The user begins by entering supplies' data into the system. The system then retrieves the necessary information and subsequently adds the new data to its records. This process illustrates the steps the system takes to update its data repository with the supplies information provided by the user.

8. **View Suppliers:**

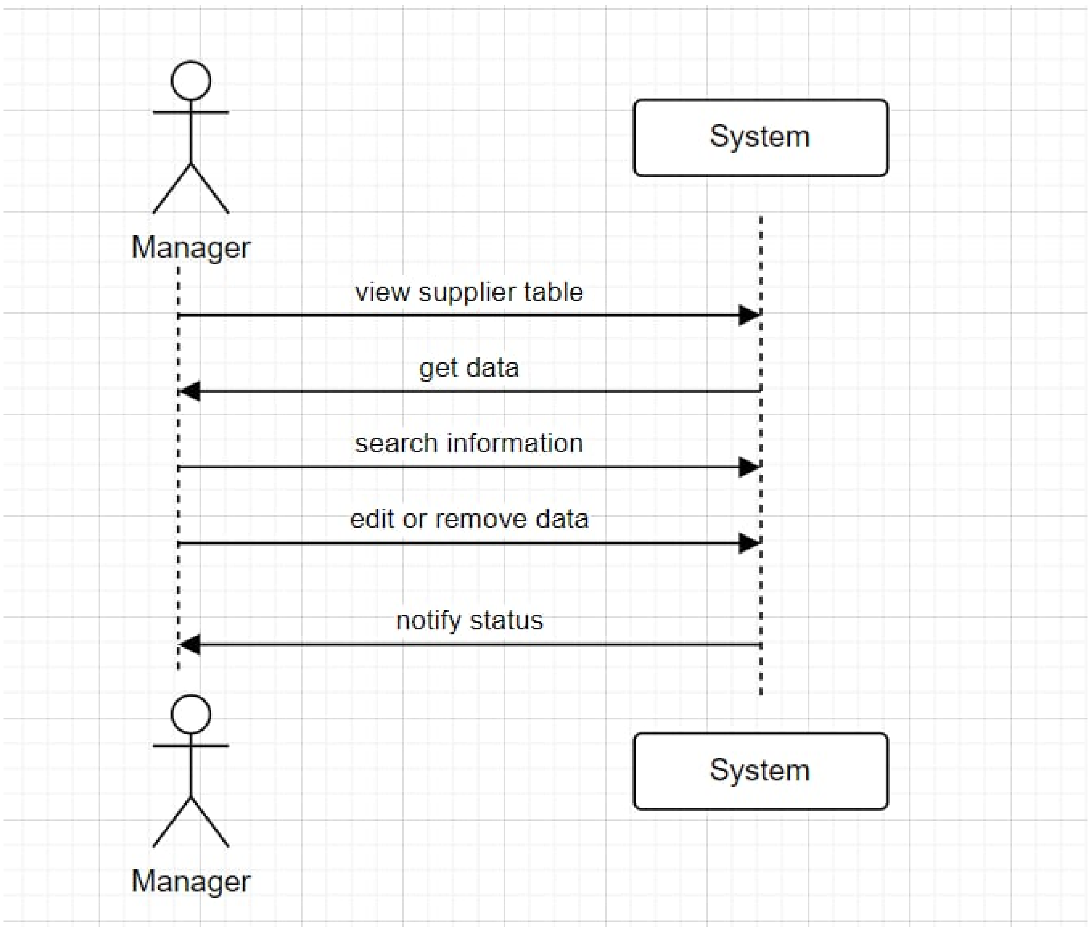


Figure 5.21: View Suppliers

The sequence diagram shows the interaction between a manager and a system. The manager initiates the process by viewing the supplier table in the system. This action prompts the system to provide the requested data. The manager then searches for specific information within the system. Following this, the manager can edit or remove data as necessary. Finally, the system notifies the manager of the status of these operations, completing the interaction cycle.

9. **Sale Report:**

The sequence diagram describes the interaction between a user, a system, and a database through three optional processes: viewing sales, viewing invoices, and viewing sales charts.

First, when the user selects ”View sales,” the system requests sales details from the

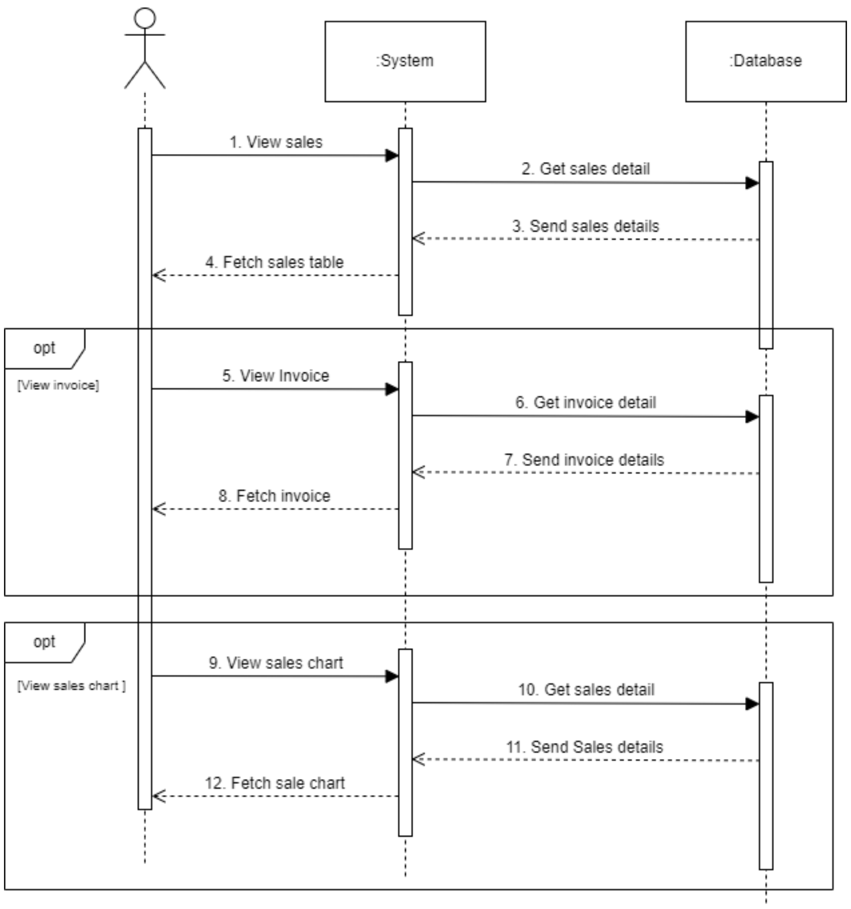


Figure 5.22: Sale Report

database. The database sends back the sales details, and the system displays the sales table to the user.

Next, the user may choose to ”View Invoice.” The system then requests invoice details from the database. Upon receiving the invoice details, the system fetches and displays the invoice to the user.

Lastly, the user can opt to ”View sales chart.” The system requests the relevant sales details from the database for the chart. The database responds with the sales details, which the system then uses to display the sales chart to the user.

Each process follows a consistent pattern where the user initiates a request, the system retrieves the necessary information from the database, and the system displays the information to the user.

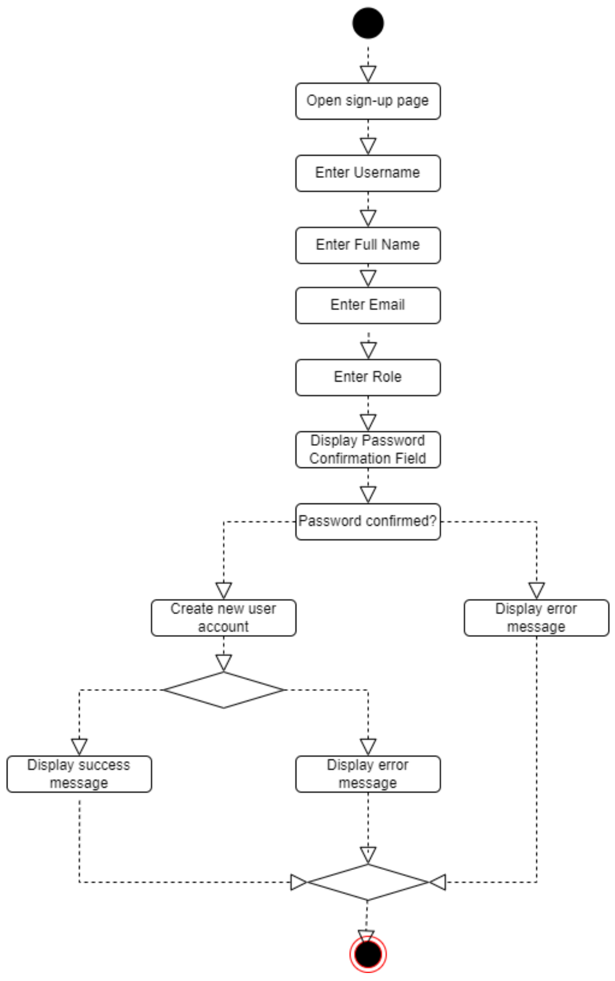


Figure 5.23: Sign Up

5.2.3 Activity Diagram

The activity diagram illustrates the user sign-up process in the system. It starts with the user opening the sign-up page and entering their Username, Full Name, Email, and Role. The system then displays a Password Confirmation Field. If the password confirmation matches, the system creates a new user account and displays a success message. If there is a mismatch or other error, the system displays an error message, prompting the user to correct the issue. This ensures a streamlined and user-friendly registration process by verifying all necessary information before account creation.

5.3 Database Design

The Entity-Relationship (ER) diagram for the pharmacy management system is designed to capture the various entities and their relationships within the system. At the core of the system are several key entities, including pharmacists, customers, sales, medicines, suppliers, conversations, chats, and invoice_medicines. Each entity is represented with its attributes and connected through various relationships to ensure a cohesive and comprehensive database structure.

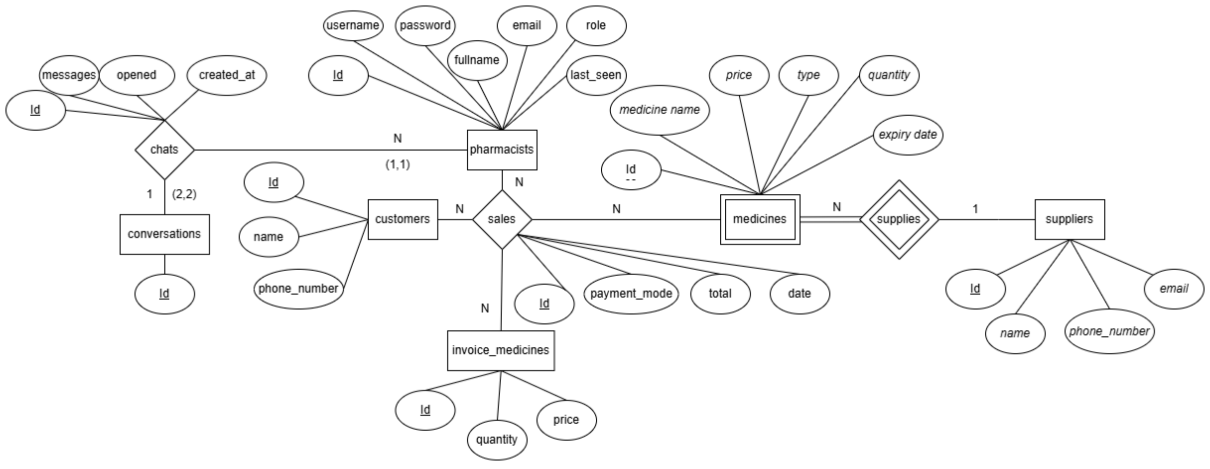


Figure 5.24: ER diagram

The pharmacists entity holds essential details about the pharmacy staff, such as id, username, password, email, fullname, role, and last_seen timestamp. Each pharmacist can be linked to multiple sales, indicating their involvement in processing transactions. Similarly, the customers entity captures customer information with attributes like id, name, and phone_number. Customers are associated with multiple sales, highlighting their purchasing activities.

Sales transactions are captured in the sales entity, which includes attributes like id, payment_mode, total amount, and date of the transaction. Each sale can consist of multiple medicines, and this relationship is managed through the invoice_medicines entity, which records details of the medicines included in each sale, such as quantity and price.

The medicines entity encompasses the details of the pharmacy's stock, including id, medicine name, price, type, quantity, and expiry date. Medicines are supplied by various suppliers, which are represented in the suppliers entity. Each supplier has unique attributes like id, name, phone_number, and email. The supply relationship between suppliers and medicines is managed through the supplies entity, linking suppliers to the medicines they provide.

Communication between pharmacists is facilitated through the conversations entity, which captures the overall discussion context. Each conversation can have multiple chats, recorded with attributes like id and created_at timestamp. The messages entity further breaks down the chats into individual messages, with each message having a unique id and an opened status.

In summary, the ER diagram outlines a robust database structure for a pharmacy management system, detailing the relationships and attributes of pharmacists, customers, sales, medicines, suppliers, and communication records. This design ensures efficient data management and seamless interaction among the various components of the system.

5.4 Security Design

The Pharmacy Management System places a strong emphasis on security, which is a critical aspect of the application. The key security measures implemented include:

Authentication: The system employs a secure login mechanism that utilizes hashed passwords. Additionally, the implementation of multi-factor authentication (MFA) is being considered to enhance the security of user access.

Authorization: The system incorporates role-based access control, ensuring that users can only access functionalities appropriate to their assigned roles. This helps to maintain the integrity of the system and prevent unauthorized access.

Data Encryption: Sensitive data is encrypted both at rest and in transit to protect the confidentiality of the information stored and transmitted within the system.

Input Validation: The application implements input validation procedures to prevent common security vulnerabilities, such as SQL injection and cross-site scripting (XSS), ensuring the overall security and integrity of the system.

RBAC prevent directory traversal attack: As all the file in the directory include a check for username and user role, the attacker can't access any files without proper authorization first. Ensuring that every user in the system is logged in and authorized.

In summary, the Pharmacy Management System has successfully implemented two key security measures: hashed passwords and prevention of SQL injection attacks. These security features, along with the other measures mentioned, are designed to provide a comprehensive security framework to protect the system and its sensitive data.

Chapter 6

Implementation

6.1 Code Structure

The Backend Implementation first start with open a connection to the MySQL database through an open connection with a database Handler through databaseHandler.inc.php with $pdo as the variable we use to interact with the database. First, we set up the connection to the database with mysql_connect command and set $pdo as a new PDO variable.

Next, we initiate a session cookie and create a _SESSION variable that store all the information in that session. We set the session's configuration through config_session.inc.php, in which we set the session's parameter such as lifetime, domain, path, etc... Then we start the session and regenerate the session based on if the user is logged in or not.

The above implementations are set in the include folder and alongside with login, signin and logout functionalities as they are heavily involved with the user's session. Beside that, all the system functionality are separated with each other as they are in their own designated folder. For example, all code base that involved with the creation, editing and viewing of invoice are stored in the invoice folder. The same are assumed with the other functionalities except stated otherwise.

6.2 Backend and database Interaction Functions

6.2.1 Create Account

Create account are only used by admin to create the accounts for the pharmacists.

The Create Account backend's code is implemented in the include folder, which is not in its own folder as we want all the code dealt with the session stay in the same place. The create account processing the session for error handling as we store the errors when signing up in the _SESSION variable.

The sign up page render a form and when the admin filled out the form. The form's data is sent to signup.inc.php through the the POST request method. Next, the requested information go through the error handler to see if there are any error. The code for the error handling are stored in signup_control.inc.php in which stored the error identifying logic. After error handling, if there are any invalid info we display the exist errors through sinup_view.inc.php, kill the current session and stored the signup info for the next session for the admin ease of use. Finally, we create an user instance in the database through the create_user() function in which located in sighup_mode.inc.php. Also in signup_control.inc.php are the code to get_username and get_email, the 2 mentioned functions are for error identifying function which implemented abode.

6.2.2 Login

Similar to create account, the backend code for login function are located in the include folder as their heavy involvement in working with session. In this use case, the login function store the logged in pharmacists' account information for this session for other applications to use.

Again, similar to the create account function, login.inc.php receives the login information through the POST request method from the login form. Then is the errors handling for the login information which the code logic stored in login_control.inc.php. Then if there are any error, the error are rendered which the the code located in login_view.inc.php, the session login info are stored for the next session and the session is killed. After the errors handling, the user is successfully logged in, transfer to the dashboard with the logged in information stored in the _SESSION variable.

6.2.3 Role Division

The role-based access control (RBAC) mechanism is implemented through the _SESSION variable. After a user is logged in, the login system stored the user role variable and at the begining of every page, the system check the user role, and rendered the page according to their role, the pages themselves also check the role of user, if the role is not relevant to the page function, the user is also redirected to the dashboard in which if the user is not logged in, they are redirected to the login page.

6.2.4 Medicine

In the medicine management application, each medicine record consists of several attributes that are crucial for ensuring comprehensive and accurate management of pharmaceutical data. These attributes are integrated into the CRUD (Create, Read, Update, Delete) operations to maintain the integrity and usability of the data.

Attributes

The primary attributes of a medicine record typically include:

- **Medicine ID:** A unique identifier for each medicine record, usually an auto-incremented integer. This attribute is essential for distinguishing between different records and is used in all CRUD operations to identify specific entries.
- **Name:** The name of the medicine. This attribute is critical for users to identify and search for specific medicines. It is a key field in both the creation and update processes and is prominently displayed in the reading operations.
- **Type:** The category or classification of the medicine (e.g., antibiotic, analgesic). This attribute helps in organizing and filtering medicines based on their usage and is used during the creation and update processes.
- **Supplier's ID:** A reference to the supplier providing the medicine. This attribute links the medicine to its supplier, facilitating supply chain management. It is used during the creation and updating of records.
- **Price:** The cost of the medicine. This attribute is important for inventory and sales management. It is part of the creation, update, and read operations to ensure accurate pricing information is maintained.
- **Expiry Date:** The date when the medicine expires. This attribute is essential for inventory management to prevent the use of expired medicines. It is used in all CRUD operations to ensure that the medicine records are up-to-date.

- **Stock Quantity:** The amount of the medicine available in stock. This attribute is crucial for inventory control and is regularly updated during the CRUD operations to reflect current stock levels.

CRUD operations

- **Create:** The `add` function in the medicine management application is responsible for both adding new medicine records and updating existing ones in the database. It first checks if the form submission method is POST using `$_SERVER['REQUEST_METHOD']`. If the form is submitted, the function sanitizes the input values using the `validate()` to prevent SQL injection and other malicious inputs. For adding a new medicine record, it employs the `insert()`, which constructs and executes an SQL INSERT query. For updating an existing record, the function uses the `update()`, which builds and runs an SQL UPDATE query based on the provided medicine ID. Upon successful addition or update, it uses the `redirect()` to navigate users back to the medicine list page with appropriate status messages.
- **Read:** The `view` function displays a comprehensive list of medicines stored in the database. It allows users to filter and search through the records by handling `$_GET` parameters such as search keywords and filter criteria. These inputs are used to construct a dynamic SQL SELECT query to fetch the filtered results. The `view` function also implements pagination to manage the display of records, calculating the total number of pages based on the total record count and the number of records displayed per page. It generates pagination links to navigate between pages. The medicines are displayed in an HTML table, with each row representing a medicine record. Action buttons for editing and deleting records are included in each row.
- **Update:** The `edit` function facilitates the modification of existing medicine records. It retrieves the current data of the medicine to be edited based on the provided ID using the `getById()`, which executes an SQL SELECT query to fetch the record. This data is then used to prepopulate the form fields, allowing the user to make necessary changes. Upon form submission, the `edit` function handles the data using `$_POST` and updates the record in the database using the `update()`, similar to the `add` function.
- **Delete:** The `delete` function is responsible for removing medicine records from the

database. It validates the medicine ID parameter using the `checkParamId()` function to ensure the ID is valid. Then it retrieves the current data of the medicine to be deleted using the `getById()` function. Once confirmed, it uses the `delete()` function to execute an SQL DELETE query, removing the record from the database.

Common elements across these functions include session management, achieved by including `config_session.inc.php` to handle user sessions, ensuring that only logged-in users with the necessary permissions can access these functionalities. The functions utilize the `alertMessage()` to display feedback messages to users regarding the success or failure of their actions. Additionally, `header.html` and `footer.html` are included in these functions to maintain a consistent layout and navigation structure across all pages in the application. These functions collectively provide a robust system for administrators to efficiently manage medicine records, ensuring data validation, secure session handling, and user-friendly feedback mechanisms.

6.2.5 Supplier

In the supplier management application, each supplier record consists of several attributes that are essential for maintaining a reliable and efficient supply chain. These attributes are incorporated into the CRUD (Create, Read, Update, Delete) operations to ensure accurate and consistent management of supplier information. Here is an overview of these attributes and their roles in CRUD operations:

Attributes

The key attributes of a supplier record include:

- **Supplier ID:** A unique identifier for each supplier record, typically an auto-incremented integer. This attribute is crucial for identifying specific records and is used in all CRUD operations to manage individual entries.
- **Name:** The name of the supplier. This attribute is vital for recognizing and searching for specific suppliers. It is a core field in the creation and update processes and is prominently displayed in the read operations.
- **Phone Number:** The contact number of the supplier. This attribute is important for communication and coordination with suppliers. It is utilized during the creation and update processes and is validated to ensure uniqueness.

- **Email:** The email address of the supplier. This attribute is essential for digital communication and correspondence. It is used in the creation and update processes and is validated to prevent duplicates.

CRUD Operations

- **Create:** During the creation of a new supplier record, all the aforementioned attributes must be provided to ensure that the record is comprehensive and useful. The `add` function is responsible for capturing these attributes through a form and validating the input before inserting the data into the database. Unique constraints on the phone number and email ensure no duplicate records are created.
- **Read:** In the read operation, such as in `view` function, the attributes are retrieved from the database and displayed to the user. The Supplier ID is used to fetch specific records, while attributes like Name, Phone Number, and Email provide detailed information about each supplier. This operation often includes search and filter functionalities that leverage these attributes.
- **Update:** The update operation, handled by `edit` function, allows users to modify the existing attributes of a supplier record. The Supplier ID is used to locate the specific record to be updated. The form is pre-populated with current attribute values, and the user can update any of the attributes as needed. Upon submission, the updated attributes are validated and saved to the database, ensuring the updated information is accurate and unique.
- **Delete:** In the delete operation, managed by `delete` function, the Supplier ID is crucial for identifying the record to be deleted. This operation removes the entire record, including all its attributes, from the database, ensuring that obsolete or incorrect data is no longer available in the system.

Common Operations

Across all CRUD operations, session management, input validation, and feedback mechanisms are implemented to ensure secure and reliable data handling. These operations ensure that only authorized users can perform CRUD actions, input data is sanitized to prevent security vulnerabilities, and users receive appropriate feedback messages after each operation.

By integrating these attributes into the CRUD operations, the supplier management application ensures accurate, efficient, and secure handling of supplier data, enhancing the overall user experience and operational integrity.

6.2.6 Customer

The Add Customer feature in the Pharmacy Management System allows pharmacists to add new customers to the system. This feature includes a form for entering the customer's name and phone number and provides functionality to save this information to the database. Additionally, the feature supports updating existing customer information.

- The feature begins by validating the user session to ensure the pharmacist is logged in. It establishes a connection to the database to retrieve and manage customer data.
- The system queries the database to fetch customer records based on the current page and search criteria. It retrieves a subset of customer records to display on the current page.
- The customer records are displayed in a table format with columns for essential details. Each record includes actions such as edit and remove, allowing users to manage individual customer records.
- Pharmacists can view the customers existed on the database on the view customer page, in which there are edit and remove options.

6.2.7 Invoice

The Pharmacy Management System includes an invoicing feature designed to streamline the management of customer orders and the tracking of medicine inventory. The system uses sessions to store selected medicine items and their quantities temporarily. This allows for persistent tracking of items added to an invoice across multiple interactions.

- Pharmacists can add multiple medicine items to an invoice. Each item is checked against the available inventory to ensure sufficient stock. If the requested quantity exceeds available stock, the pharmacist is notified.
- Upon proceeding to place an order, the system checks if the customer exists in the database based on the provided phone number. If the customer exists, an invoice number is generated, and the customer's phone number and payment mode are stored in the session for

future reference. If a customer does not exist, the system allows for the creation of a new customer profile by capturing the customer's name and phone number.

- When the pharmacist decides to save the invoice, the system first verifies the pharmacist's credentials using the session data. The system then validates the existence of the customer and calculates the total amount for the invoice based on the selected medicine items.
- A new invoice is created in the database, and each selected medicine item is recorded as part of the invoice. The inventory is updated to reflect the reduced quantity of each medicine item sold. Finally, session variables related to the invoice process are cleared to prepare for the next transaction.

6.2.8 Sales

After creating an invoice, pharmacists can then go to the Sales section to view past orders.

- The system retrieves sales data by executing an SQL query that joins the sales, pharmacists, and customers tables. This query fetches all relevant sales information, including the pharmacist's name and the customer's phone number.
- The sales records are displayed in a table format with the following columns: Invoice ID, Date, Pharmacist ID, Pharmacist Name, Customer, Customer's Phone, Payment Mode, Total, Expiry Date, and Action. Each row represents an individual sale, and the action column contains a button to view detailed information about the invoice.
- The feature includes a footer in the table that calculates and displays the total revenue by summing up the total amounts of all sales records.
- If no sales records are found, the system displays a message indicating that no invoices were found. If the user is not logged in, the system redirects them to the login page.

6.2.9 Employee

In the employee management system, each employee record comprises several attributes crucial for efficient workforce management. These attributes are integral to the CRUD (Create, Read, Update, Delete) operations, ensuring accurate and consistent management of employee information. Here is an overview of these attributes and their roles in CRUD operations:

Attributes

The primary attributes of an employee record include:

- **ID:** A unique identifier for each employee, typically an auto-incremented integer. This attribute is fundamental for identifying specific records and is used in all CRUD operations to manage individual entries.
- **User Name:** The username used for employee login purposes. This attribute serves as a unique identifier for authentication and authorization, essential for secure system access. It is prominently used in read, update, and delete operations to manage user accounts.
- **Full Name:** The full name of the employee. This attribute is essential for identifying individuals within the organization. It is prominently displayed in read operations and used in update operations to ensure accurate employee details.
- **Email:** The email address of the employee. This attribute is crucial for communication and correspondence within the organization. It is used in all CRUD operations, especially during creation and update processes, and validated to ensure uniqueness.
- **Role:** The role or position held by the employee within the organization (e.g., manager, developer, analyst). This attribute defines the permissions and responsibilities assigned to each employee. It is used in read and update operations to manage access rights and organizational hierarchy effectively.
- **Password:** The secure password associated with the employee's account. This attribute is crucial for user authentication and access control. While not displayed, it is managed securely in the system and updated through specific CRUD operations to ensure account security.

CRUD Operations

- **Create:** During the creation of a new employee record, all the aforementioned attributes must be provided to ensure comprehensive employee information. The `signup` function captures these attributes through a form, validating input and ensuring unique constraints on username and email addresses to prevent duplicates in the database.
- **Read:** In the read operation, such as in `view` function, attributes are retrieved from the

database and displayed to authorized users. The Employee ID is used to fetch specific records, while attributes like User Name, Full Name, Email, and Role provide detailed information about each employee. This operation often includes search and filter functionalities based on these attributes.

- **Update:** The update operation, managed by `edit` function, allows administrators to modify existing attributes of an employee record. The Employee ID identifies the record to be updated. The form is pre-populated with current attribute values, enabling updates to User Name, Full Name, Email, Password, and Role as needed. Input validation ensures updated information is accurate and adheres to unique constraints.
- **Delete:** In the delete operation, handled by `delete` function, the Employee ID plays a crucial role in identifying and removing specific employee records from the system. This operation ensures that obsolete or incorrect employee data is effectively removed from the database, maintaining data integrity.

Common Operations

Across all CRUD operations, session management, input validation, and feedback mechanisms are implemented to ensure secure and reliable data handling. These operations enforce user permissions, sanitize input data to prevent security vulnerabilities, and provide appropriate feedback messages after each operation.

By integrating these attributes into the CRUD operations, the employee management system ensures efficient management of workforce data, facilitating organizational processes and enhancing operational integrity.

6.2.10 Chat Page

The chat page are the most different from the other page, and interact with 3 set of database table, pharmacists, chats and conversations. As chat function are more different and complex than the other, its architecture is more complicated. As an example, we use ajax (Asynchronous JavaScript and XML) from jquery to to handle concurrent multiple users interactions, ensure real-time communication and smooth user experience.

The chat application start with home.php as its landing page. In which we can see our user information, a search box, and our past conversations. To accessing such page, we first check users' logged in status, then get user's info and their past conversation through getConversation() function located in helpers folder. Next you can search for the user you want to chat with and chat with them through the chat.php file, the file that handling sending and messages and displaying information about the conversation like messages sent and the other users last seen time. Once you compose your messages, and you send them, the system post the messages to insert.php in ajax folder to insert your messages to the chat database.

Inside the chat folder, there are additional folders contain the code that handle the chat apps logic, including database interactions, session management, and business logic.

One folder that handle database interaction and utilizing ajax's functionalities are ajax folder, more specifiaclly the files in it. First we have search.php and update_last_seen.php which is used in home.php. The update_last_seen.php file update the last seen time of an user to the database. Next, the search.php control the searching other user functionality, it fetch user's input keys from the POST request method and query similar name in the database and then display them in home.php. Beside them, we have getMessage.php and insert.php which handle the sending and receiving functionality of the chat.php file feature. getMessage.php fetches all the messages from the user and the user they chatted with and displaying them. Meanwhile the insert.php file insert the sent messages into the chats table in the database.

In addition to ajax folder, we have the codes in helpers folder which contains various utility function. The user.php file contain the getUser() function to return the user information from the database, the timeAgo.php contain the function that return the time the last time certain user seen the message, opened.php change the status of the opened status of the chat, last_chat.php return the last message between the user and the other user, chat.php return all the chat between those 2 users. Finally, conversations.php contain the code to return the information of the chats sessions known in the database as the conversation table.

Chapter 7

Deployment

7.1 Introduction

Deploying robust and scalable web applications involves careful used of server-side language like PHP and database management systems like MySQL. The below documents the deployment process of integrating PHP, MySQL, and phpMyAdmin using 2 different approach which are traditional web hosting adn Docker containerization.

The deployment strategies discussed aim to provide a documented process from our team's developers trying to establish a stable and scalable environment for our web applications. By leveraging web hosting services and Docker technology, this report explores different methods of deployment workflows while ensuring accessibility, reliability, and efficient management of both application code and database assets.

From our steps and insights, this report not only outlines the setup and configuration of PHP, MySQL, and phpMyAdmin but also addresses potential challenges and considerations in each deployment approach. Whether deploying on a shared hosting platform or through Docker containers, understanding these methodologies equips developers with the tools necessary to effectively manage and scale their web applications in diverse operational environments.

7.2 Preparation

7.2.1 Tools and Technologies

For Web Hosting Provider

- Web Hosting provider
- Git repository (optional)

For Dockerization

- Docker Desktop

7.2.2 Prerequisite

For Web Hosting Provider

- Set up an account
- Prepare your git repository
- Prepare your folder

For Dockerization

- Download Docker Desktop using this link https://www.docker.com/products/docker-desktop/

7.3 Deployment using Web Hosting

7.3.1 Hosting Setup

For this option, we decides to choose 000webhost.com as our choice for its free hosting, wide range of services and streamline hosting process.

- Account Creation
- Domain Configuration
- File Upload

7.3.2 Database Setup

- Creating a MySQL Database

- Database Configuration

7.3.3 phpMyAdmin Setup

- Configuration
- Run SQL commands;

7.4 Deployment using Docker Image

7.4.1 Creating Docker Images

For this web app to run, we create multiple images to run different services. 1 Image for PHP, 1 for MySQL and 1 for phpMyAdmin.

- Creating image for PHP
- Creating image for MySQL
- Creating image for phpMyAdmin

7.4.2 Docker Compose

Compose File:

Building and Running Containers: After setup Dockerfile and Compose file, we run the following command to set the Container up and running. First, we run, `docker-compose build` and then run `docker-compose up`, and with just that, we have an up and running service.

If you want to change anything in the image set up for debugging and fixing configuration. We stop the process by `Ctrl + C` and we take the images down with `Docker-compose down`.

7.5 Troubleshooting and Debugging

7.5.1 Common Issues

Web hosting

When setting web host, the encounter problem that involving database connections as is not apparent what is the correct credential at first. Another problem is that the session configuration is not controlled by us but the server that host our web and we have to fix our code accordingly.

Set up Docker

Some of the common issues that we encounter would be database connection problem as the php image we have doesn't have all the necessary libraries installed as first.

Another issues we encounter are versions conflict and we have to try several different version for the docker image have the same functionalities as we developed on our local environment.

7.5.2 Debugging

We have a hard time debugging all the errors in the debugging process while deploying our project on web as the debugging process is not really streamline, we employ several different techniques as inspecting the console log, echo variables, and more simple find the error warning and research how to find it.

For Dockerization process, as we are still inexperience, we encounter some of the issues we met while deploy on web host. The process is still not streamline, so we utilize technique like above for debugging.

7.6 Conclusion

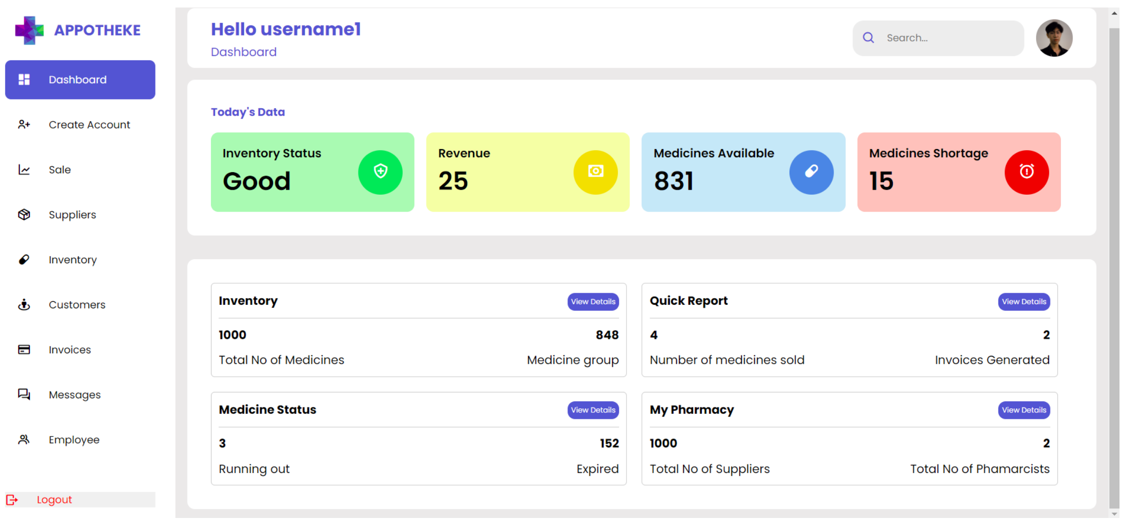
In conclusion, the overall process of deploying our web app is streamline and simple but we encountered huge problems with troubleshooting and debugging that lengthen our deploying process duration. We attribute such difficulties to our insufficient knowledge of both of the above deploying process. But at last we still overcome them with the help of numerous stack overflow articles.

Chapter 8

User Guide

8.1 Navigating The Dashboard

The dashboard of the pharmacy management system provides a comprehensive overview of the key operational metrics and data for the pharmacy. The "Today's Data" section displays high-level information on the current inventory status, revenue, available medicines, and medicines shortage, allowing the user to quickly assess the pharmacy's performance at a glance.
When the user clicks on the "View Details" button under the "Inventory" section, they are redirected to the dedicated "Inventory" page, which likely offers a more detailed view of the pharmacy's stock levels, product information, and other relevant inventory-related data.
Similarly, clicking on the "View Details" button under the "Quick Report" section would take the user to the "Sale" page, where they can access more detailed sales-related information, such as the number of medicines sold and the total invoices generated.
The "Medicine Status" section displays the current status of the pharmacy's medicines, including any running out or expired items. By clicking on the "View Details" button, the user is directed to the "Inventory" page, but with a focus on the expired medicines specifically.
Finally, the "My Pharmacy" section provides an overview of the pharmacy's key metrics, including the total number of suppliers and pharmacists. Clicking on the "View Details" button in this section would take the user to the "Employee" page, where they can likely manage the pharmacy's staff and personnel-related information.



Hello username1
Dashboard

Today's Data

Inventory Status Good
Revenue 25
Medicines Available 831
Medicines Shortage 15

Inventory
1000 Total No of Medicines
848 Medicine group

Quick Report
4 Number of medicines sold
2 Invoices Generated

Medicine Status
3 Running out
152 Expired

My Pharmacy
1000 Total No of Suppliers
2 Total No of Phamarcists

Figure 8.1: Dashboard Design

8.2 Create Account For Employee

To onboard new employees, navigate to the "Create Account" section. In the "Create Account" form, enter the required information for the new employee. Start by providing a unique username that the employee will use to log in to the system. Next, enter the employee's full legal name in the "Full Name" field. Input the employee's valid email address in the "Email Address". From the dropdown menu in the "Role" field, select the appropriate job title for the employee. Finally, set a secure password for the employee's account in the "Password" field. Once you have filled out all the necessary information, click the "Create Account" button to save the new employee's details. The employee can then use their assigned username and password to access the Pharmacy Management System.

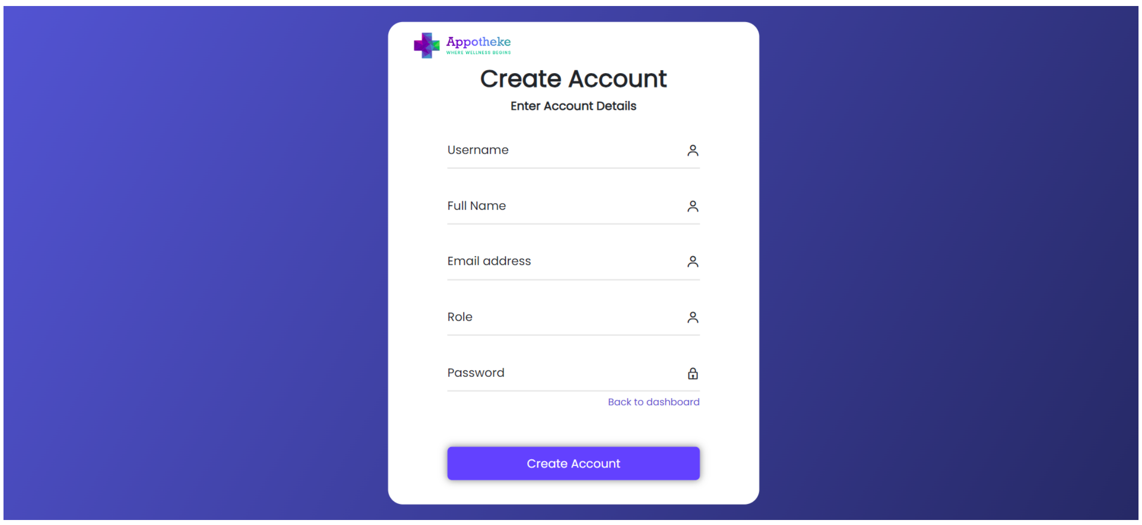
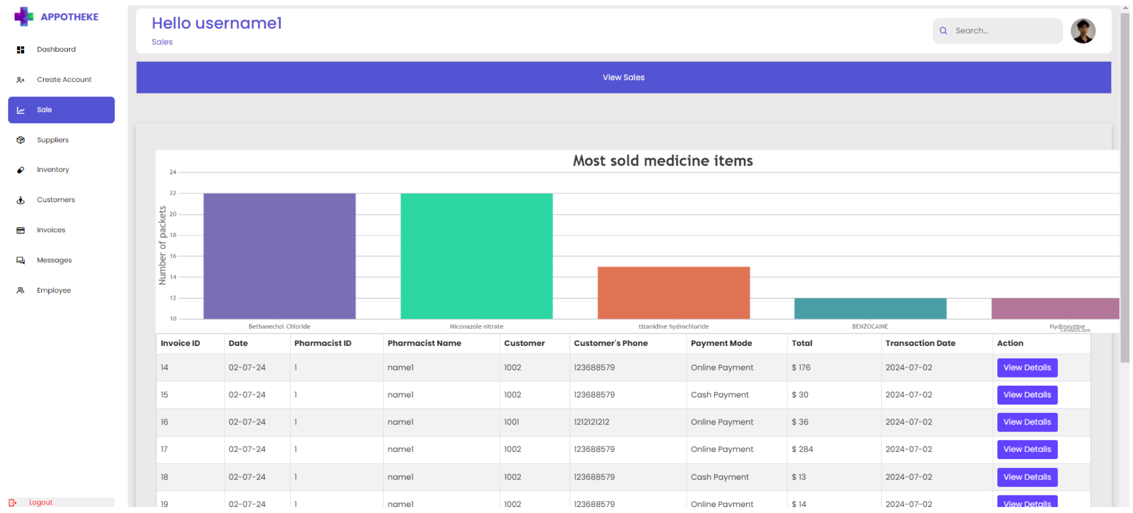


Figure 8.2: Create Account

8.3 Sales

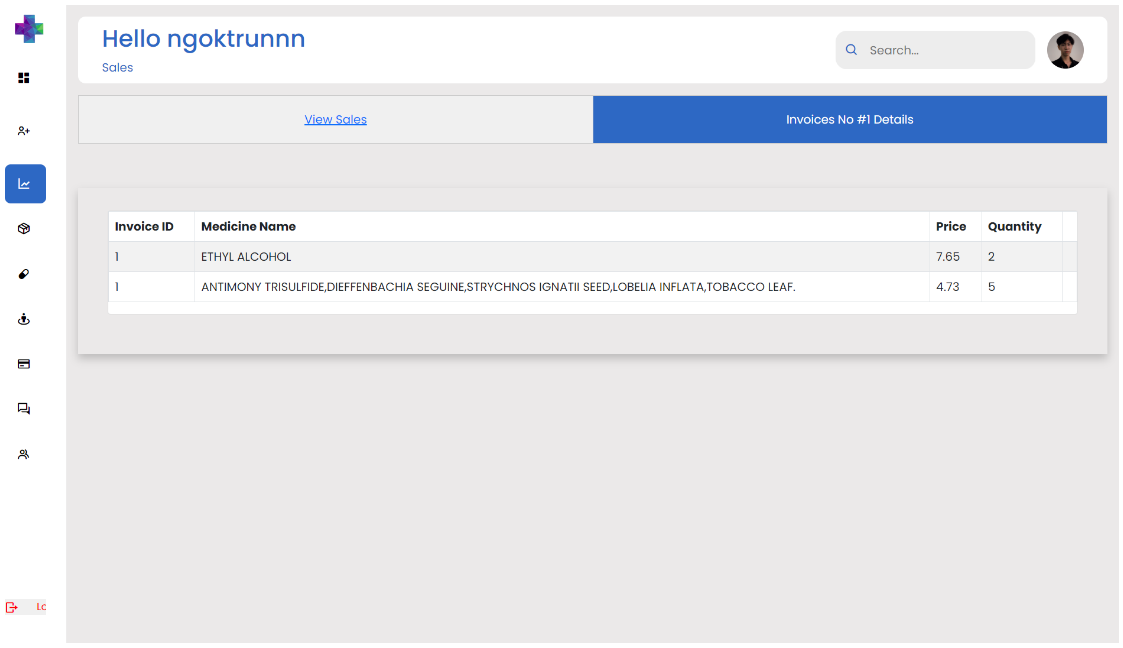
The ”Sales” section serves as the command center for accessing your pharmacy’s complete sales history. This comprehensive view presents a detailed record of every past transaction, providing key information to help you monitor trends, identify opportunities, and ensure accurate bookkeeping.When accessing the ”View Sales” page, users are presented with a comprehensive list of all past transactions. For each sale, this view displays key information such as the Invoice ID, the Date the order was placed, the Pharmacist ID and full Name of the employee who processed the transaction, the Customer’s Name and Phone number, the Payment Mode used, the Total amount charged, and the Expiry Data for any prescription medications included.



Invoice ID	Date	Pharmacist ID	Pharmacist Name	Customer	Customer's Phone	Payment Mode	Total	Transaction Date	Action
14	02-07-24	1	name1	1002	123688579	Online Payment	$ 176	2024-07-02	View Details
15	02-07-24	1	name1	1002	123688579	Cash Payment	$ 30	2024-07-02	View Details
16	02-07-24	1	name1	1001	1212121212	Online Payment	$ 36	2024-07-02	View Details
17	02-07-24	1	name1	1002	123688579	Online Payment	$ 284	2024-07-02	View Details
18	02-07-24	1	name1	1002	123688579	Cash Payment	$ 13	2024-07-02	View Details
19	02-07-24	1	name1	1002	123688579	Online Payment	$ 14	2024-07-02	View Details

Figure 8.3: View Sales

Alongside this detailed sales data, the ”View Sales” page also features a ”View Details” button at the end of each row. Clicking this button opens a new page that provides the full invoice breakdown, allowing users to review the individual medicines purchased, their Prices, and Quantities. This granular visibility into past orders empowers the pharmacy team to monitor trends, identify growth opportunities, and maintain meticulous record-keeping.



Invoice ID	Medicine Name	Price	Quantity
1	ETHYL ALCOHOL	7.65	2
1	ANTIMONY TRISULFIDE,DIEFFENBACHIA SEGUINE,STRYCHNOS IGNATII SEED,LOBELIA INFLATA,TOBACCO LEAF.	4.73	5

Figure 8.4: View Sales Details

8.4 Using The Suppliers Function

The ”Supplier” section allows you to manage your network of product vendors. The ”Add Suppliers” feature enables you to log new supplier details, including the company name, phone number, and email address. This centralized supplier database ensures you have a comprehensive record of all your approved and active partners for purchasing supplies and other pharmacy resources.

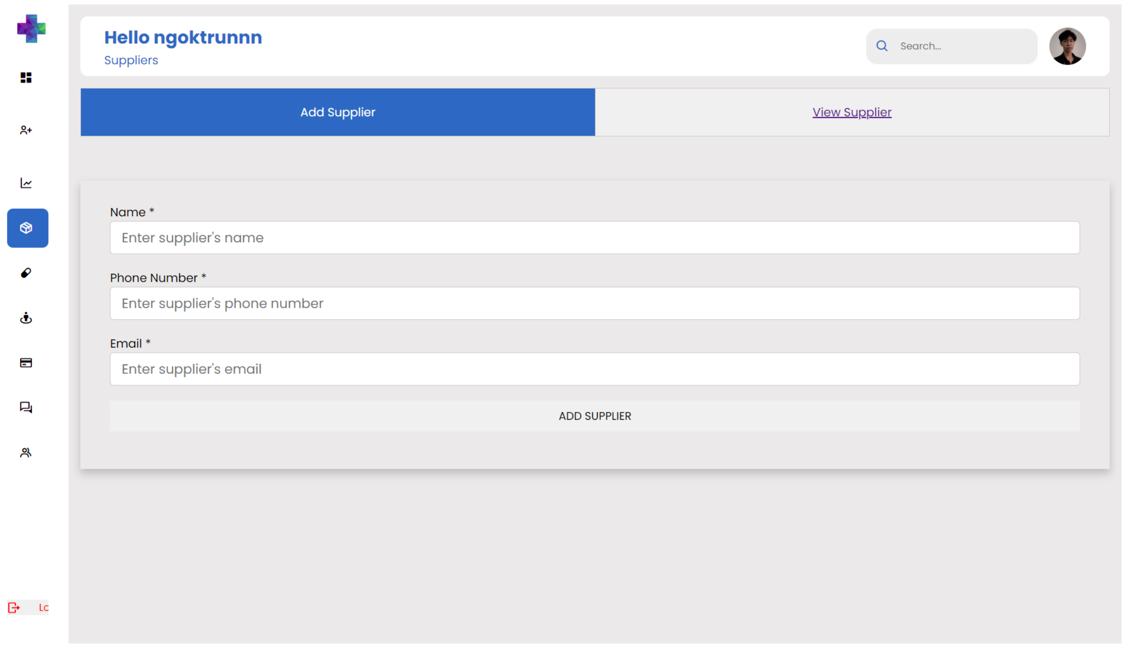
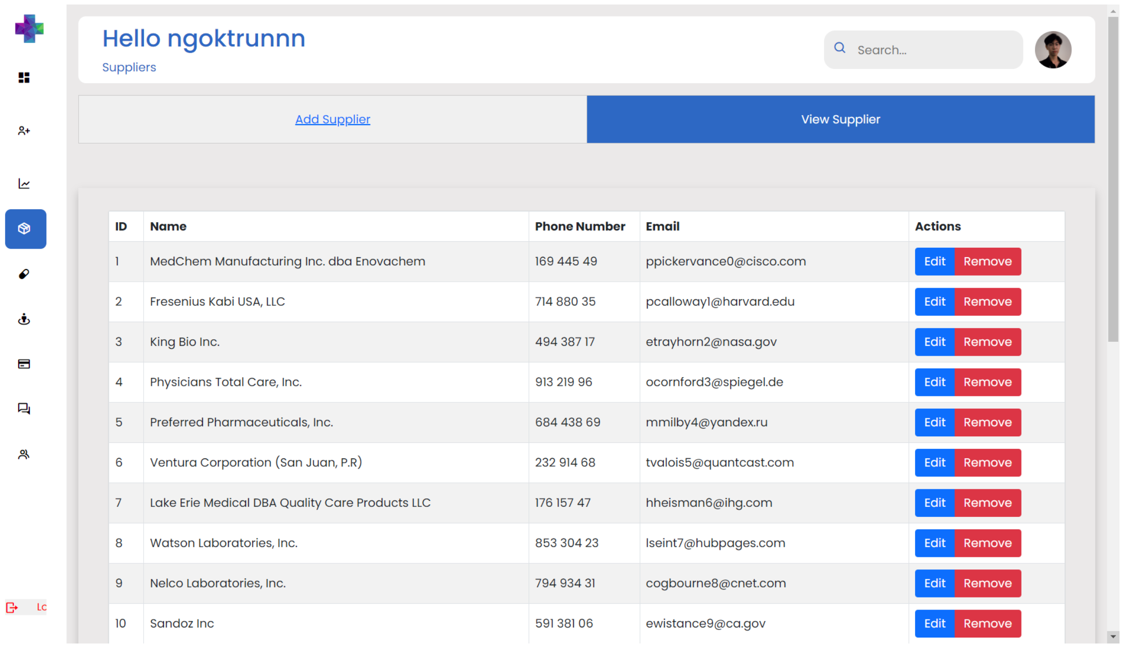


Figure 8.5: Add Suppliers

In the "View Suppliers" section, you can access a comprehensive list of all suppliers currently registered in the system. This view provides detailed information about each vendor, including their unique supplier ID, company name, contact phone number, and email address. From this "View Supplier" screen, you also have the ability to remove vendors if necessary. The robust functionality of this section empowers you to maintain an up-to-date and accurate supplier directory, allowing you to quickly reference key contact details and make informed choices when placing orders or evaluating potential new partners.

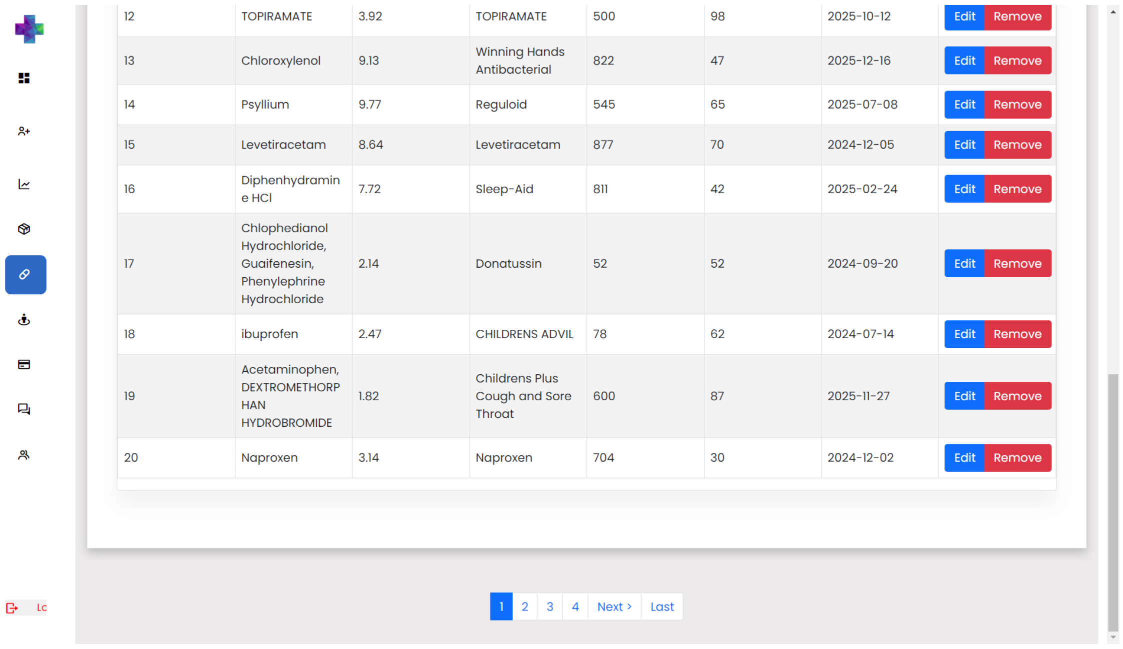
At the top right corner of the pharmacy management system's "View Suppliers" section, users will find a dedicated search field. This feature enables users to easily search and retrieve information about the various suppliers that the pharmacy works with.



ID	Name	Phone Number	Email	Actions
1	MedChem Manufacturing Inc. dba Enovachem	169 445 49	ppickervance0@cisco.com	Edit Remove
2	Fresenius Kabi USA, LLC	714 880 35	pcallowayl@harvard.edu	Edit Remove
3	King Bio Inc.	494 387 17	etrayhorn2@nasa.gov	Edit Remove
4	Physicians Total Care, Inc.	913 219 96	ocornford3@spiegel.de	Edit Remove
5	Preferred Pharmaceuticals, Inc.	684 438 69	mmilby4@yandex.ru	Edit Remove
6	Ventura Corporation (San Juan, P.R)	232 914 68	tvalois5@quantcast.com	Edit Remove
7	Lake Erie Medical DBA Quality Care Products LLC	176 157 47	hheisman6@ihg.com	Edit Remove
8	Watson Laboratories, Inc.	853 304 23	lseint7@hubpages.com	Edit Remove
9	Nelco Laboratories, Inc.	794 934 31	cogbourne8@cnet.com	Edit Remove
10	Sandoz Inc	591 381 06	ewistance9@ca.gov	Edit Remove

Figure 8.6: View Suppliers

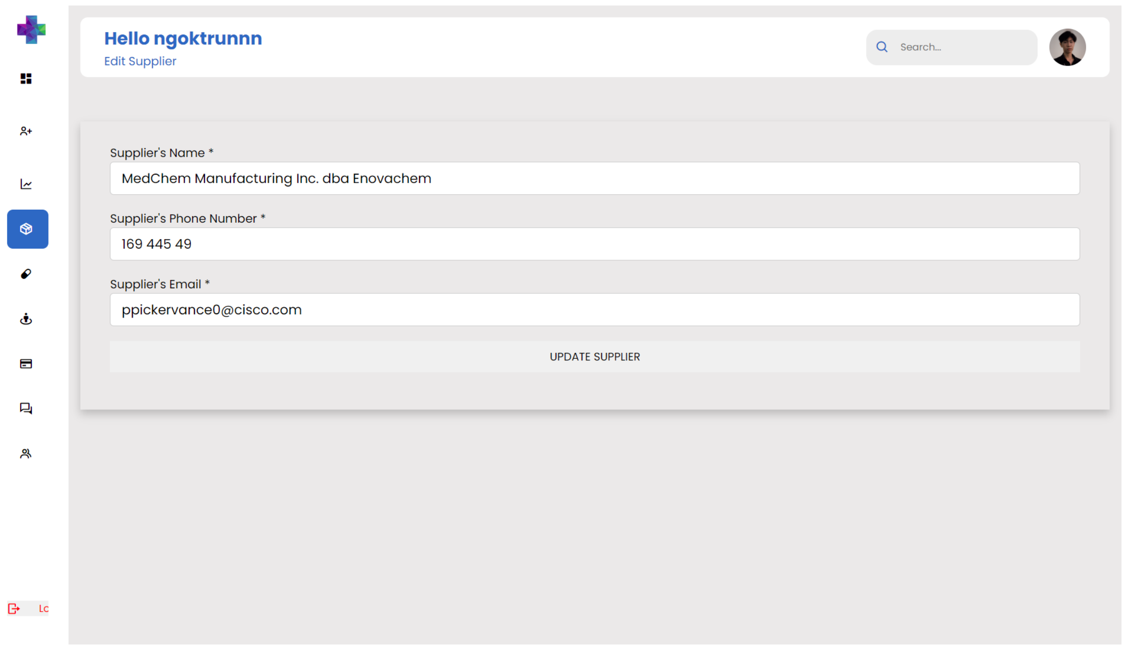
At the end of the page, you'll also find pagination controls, enabling you to navigate through the full list of medicines in your inventory.



12	TOPIRAMATE	3.92	TOPIRAMATE	500	98	2025-10-12	Edit Remove
13	Chloroxylenol	9.13	Winning Hands Antibacterial	822	47	2025-12-16	Edit Remove
14	Psyllium	9.77	Reguloid	545	65	2025-07-08	Edit Remove
15	Levetiracetam	8.64	Levetiracetam	877	70	2024-12-05	Edit Remove
16	Diphenhydramin e HCl	7.72	Sleep-Aid	811	42	2025-02-24	Edit Remove
17	Chlophedianol Hydrochloride, Guaifenesin, Phenylephrine Hydrochloride	2.14	Donatussin	52	52	2024-09-20	Edit Remove
18	ibuprofen	2.47	CHILDRENS ADVIL	78	62	2024-07-14	Edit Remove
19	Acetaminophen, DEXTROMETHORP HAN HYDROBROMIDE	1.82	Childrens Plus Cough and Sore Throat	600	87	2025-11-27	Edit Remove
20	Naproxen	3.14	Naproxen	704	30	2024-12-02	Edit Remove

Figure 8.7: Pagination

The "Edit Supplier" feature in the pharmacy management system allows authorized users to update the details of an existing supplier record. This functionality is an important component for maintaining accurate and up-to-date information about the pharmacy's suppliers. a form with fields for the supplier's name, phone number, and email address. Users can modify these details as needed, ensuring that the pharmacy's supplier data remains current and reliable. This is particularly crucial for tasks like processing orders, managing inventory, and maintaining effective communication with the supplier.

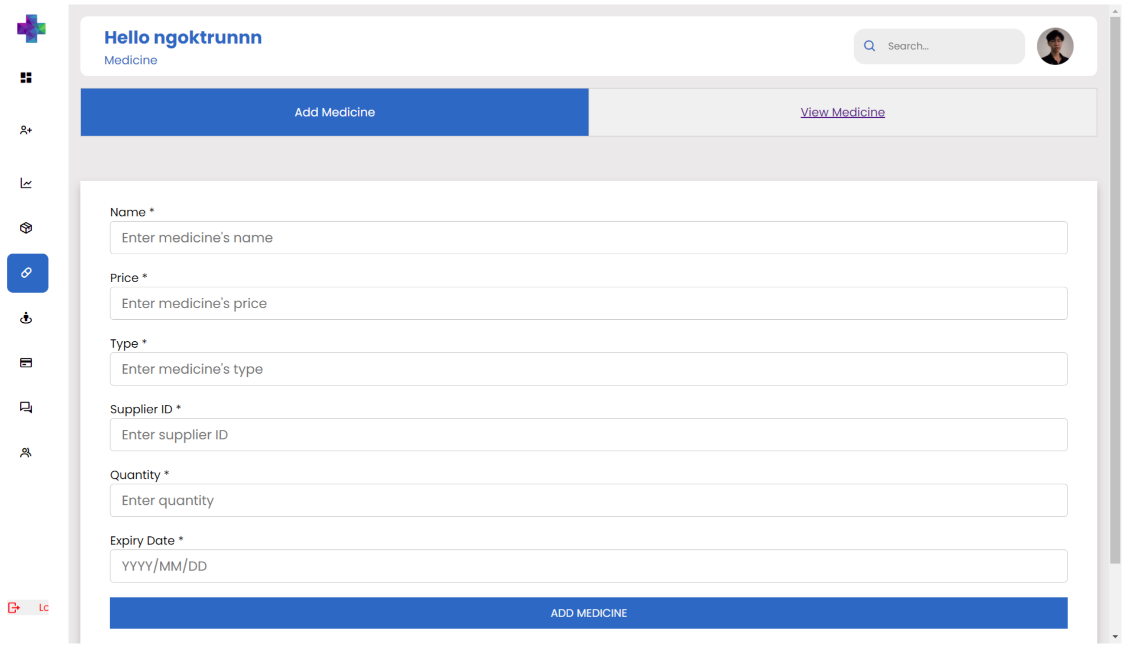


Supplier's Name *
MedChem Manufacturing Inc. dba Enovachem
Supplier's Phone Number *
169 445 49
Supplier's Email *
ppickervance0@cisco.com

Figure 8.8: Edit Suppliers

8.5 Using The Medicine Function

The "Inventory" module also allows you to manage your pharmaceutical inventory with precision. To add new medicines, click the "Add Medicines" button, which will open a form for you to enter the item details. Here, you can input information such as the medicine name, price, type, supplier id, quantity, and expiry date. Once you have filled out all the necessary fields, save the new medicine item to update your inventory records.

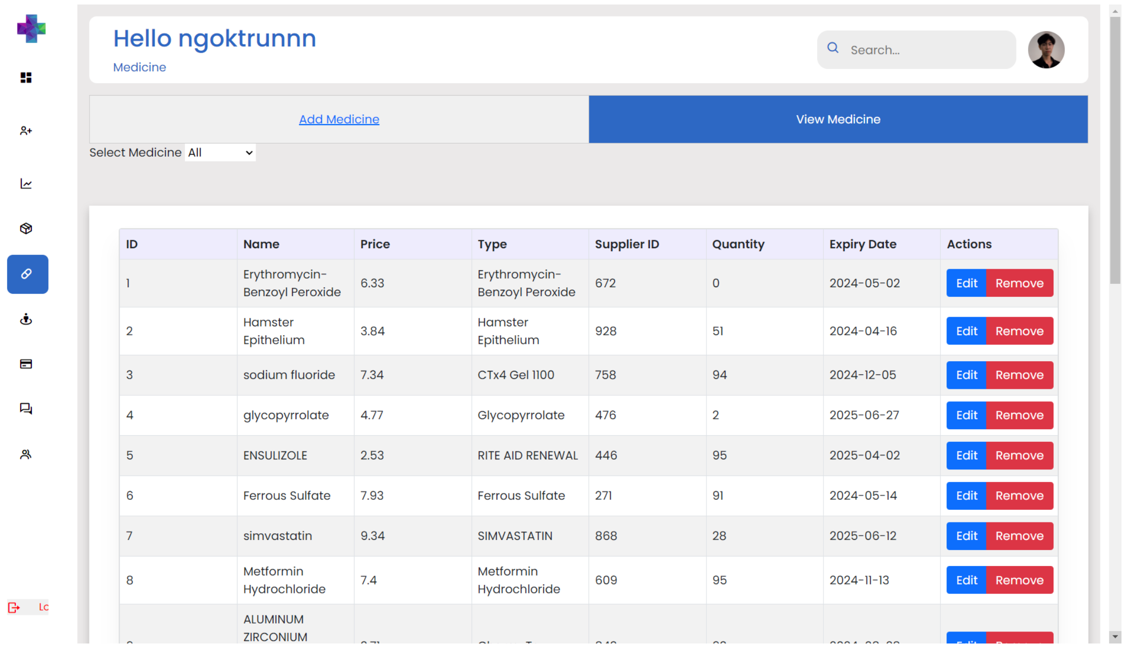


Name *
Enter medicine's name
Price *
Enter medicine's price
Type *
Enter medicine's type
Supplier ID *
Enter supplier ID
Quantity *
Enter quantity
Expiry Date *
YYYY/MM/DD

Figure 8.9: Add Medicine

In addition to adding new medicines, the ”Inventory” module also provides the ability to view your current medicine levels. The ”View Medicine” section presents a detailed overview of your entire pharmaceutical inventory. This includes the unique ID, name, price, type, current quantity in stock, and expiration date for each medicine. Alongside this information, the ”View Medicine” page also includes edit and remove buttons for each listed item. And if a medicine needs to be removed from your inventory entirely, the remove button provides a quick way to delete that product record.

The ”Select Medicine” section below the ”Edit Medicine” area provides the user with several filtering options to view the pharmacy's medicine inventory in different ways. One of the filters allows the user to see the list of all Available Medicines. This view displays the complete inventory of active, non-expired medicines that the pharmacy currently has in stock. This function gives the user a comprehensive overview of the medicines that are readily available for dispensing to patients.



Hello ngoktrunnn
Medicine

Search...

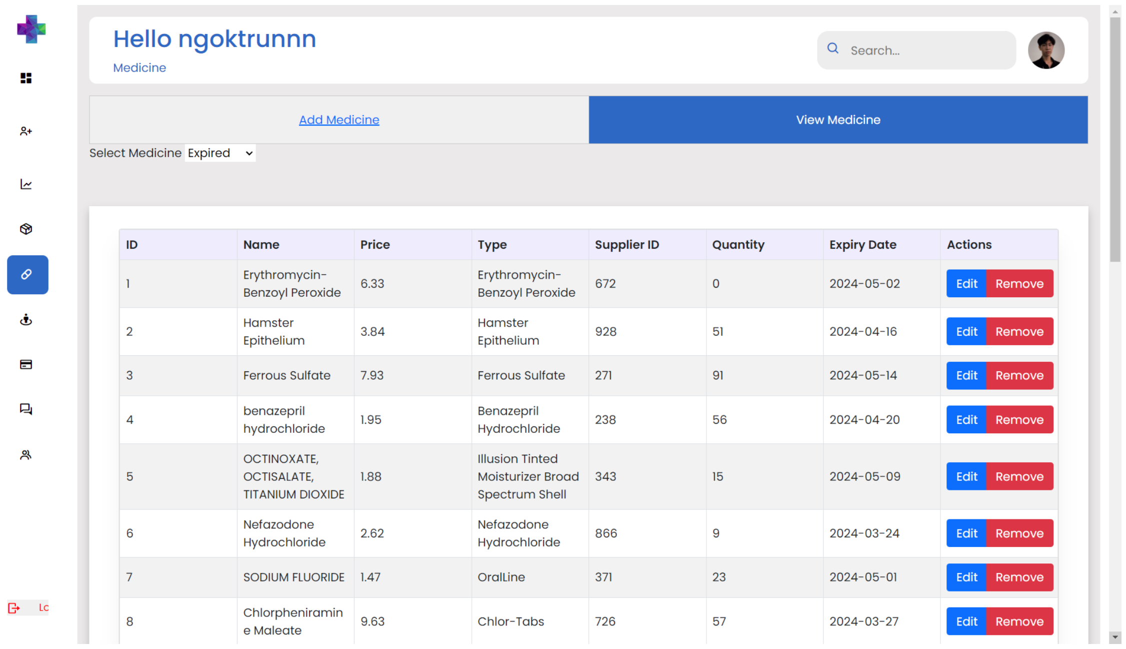
Add Medicine | View Medicine

Select Medicine All

ID	Name	Price	Type	Supplier ID	Quantity	Expiry Date	Actions
1	Erythromycin-Benzoyl Peroxide	6.33	Erythromycin-Benzoyl Peroxide	672	0	2024-05-02	Edit Remove
2	Hamster Epithelium	3.84	Hamster Epithelium	928	51	2024-04-16	Edit Remove
3	sodium fluoride	7.34	CTx4 Gel 1100	758	94	2024-12-05	Edit Remove
4	glycopyrrolate	4.77	Glycopyrrolate	476	2	2025-06-27	Edit Remove
5	ENSULIZOLE	2.53	RITE AID RENEWAL	446	95	2025-04-02	Edit Remove
6	Ferrous Sulfate	7.93	Ferrous Sulfate	271	91	2024-05-14	Edit Remove
7	simvastatin	9.34	SIMVASTATIN	868	28	2025-06-12	Edit Remove
8	Metformin Hydrochloride	7.4	Metformin Hydrochloride	609	95	2024-11-13	Edit Remove
	ALUMINUM ZIRCONIUM						

Figure 8.10: View All Medicine

Another filter enables the user to focus specifically on Expired Items. This is a crucial feature, as it helps the pharmacy staff identify and manage any medicines that have reached their expiration date. Closely monitoring expired products is essential for maintaining patient safety and ensuring the pharmacy's inventory is up-to-date.



Hello ngoktrunnn
Medicine

Search...

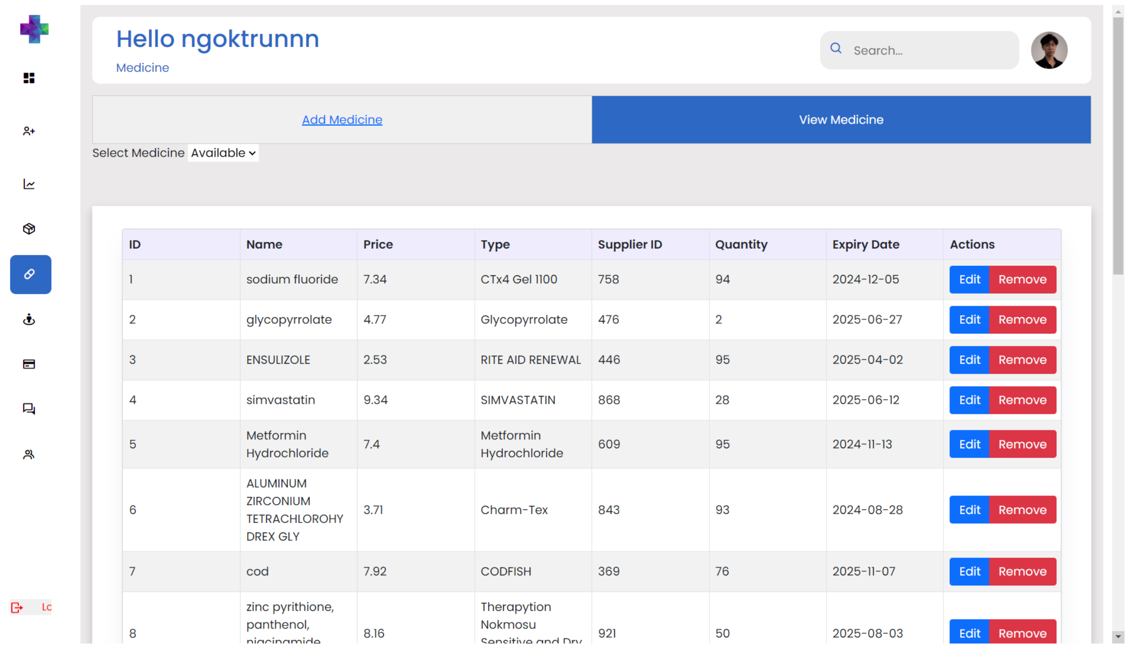
Add Medicine | View Medicine

Select Medicine Expired

ID	Name	Price	Type	Supplier ID	Quantity	Expiry Date	Actions
1	Erythromycin-Benzoyl Peroxide	6.33	Erythromycin-Benzoyl Peroxide	672	0	2024-05-02	Edit Remove
2	Hamster Epithelium	3.84	Hamster Epithelium	928	51	2024-04-16	Edit Remove
3	Ferrous Sulfate	7.93	Ferrous Sulfate	271	91	2024-05-14	Edit Remove
4	benazepril hydrochloride	1.95	Benazepril Hydrochloride	238	56	2024-04-20	Edit Remove
5	OCTINOXATE, OCTISALATE, TITANIUM DIOXIDE	1.88	Illusion Tinted Moisturizer Broad Spectrum Shell	343	15	2024-05-09	Edit Remove
6	Nefazodone Hydrochloride	2.62	Nefazodone Hydrochloride	866	9	2024-03-24	Edit Remove
7	SODIUM FLUORIDE	1.47	OralLine	371	23	2024-05-01	Edit Remove
8	Chlorpheniramine Maleate	9.63	Chlor-Tabs	726	57	2024-03-27	Edit Remove

Figure 8.11: View Expired Medicine

The third filtering option shows the user the list of Medicines Currently in Stock. This view focuses solely on the medicines that are actively being managed and can be immediately provided to patients. This filter allows the user to quickly assess the pharmacy's active inventory without being distracted by expired or out-of-stock items.



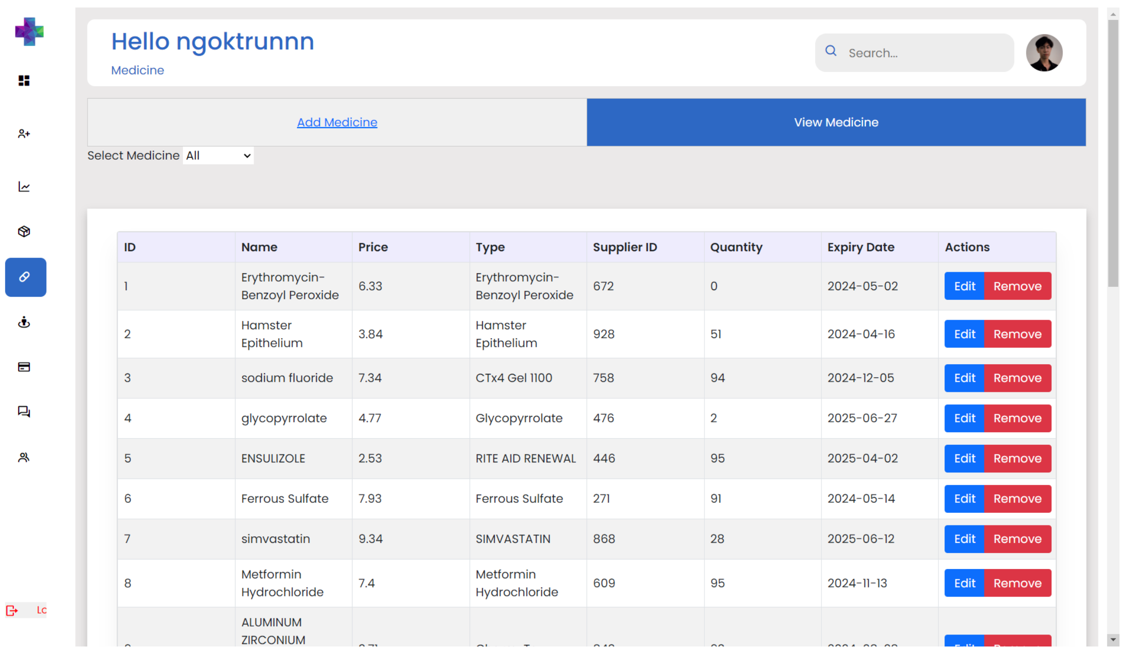
Select Medicine Available

ID	Name	Price	Type	Supplier ID	Quantity	Expiry Date	Actions
1	sodium fluoride	7.34	CTx4 Gel 1100	758	94	2024-12-05	Edit Remove
2	glycopyrrolate	4.77	Glycopyrrolate	476	2	2025-06-27	Edit Remove
3	ENSULIZOLE	2.53	RITE AID RENEWAL	446	95	2025-04-02	Edit Remove
4	simvastatin	9.34	SIMVASTATIN	868	28	2025-06-12	Edit Remove
5	Metformin Hydrochloride	7.4	Metformin Hydrochloride	609	95	2024-11-13	Edit Remove
6	ALUMINUM ZIRCONIUM TETRACHLOROHY DREX GLY	3.71	Charm-Tex	843	93	2024-08-28	Edit Remove
7	cod	7.92	CODFISH	369	76	2025-11-07	Edit Remove
8	zinc pyrithione, panthenol, niacinamide	8.16	Therapytion Nokmosu Sensitive and Dry	921	50	2025-08-03	Edit Remove

Figure 8.12: View Available Medicine

Integrating the search capability directly within the "View Medicine" section provides an optimized user experience. Rather than having to manually browse through the entire list of medicines, users can simply enter their search query and the system will instantly present the matching results. This saves time and enhances the efficiency of managing the pharmacy's drug inventory, while also meeting the users' needs for rapid information retrieval.

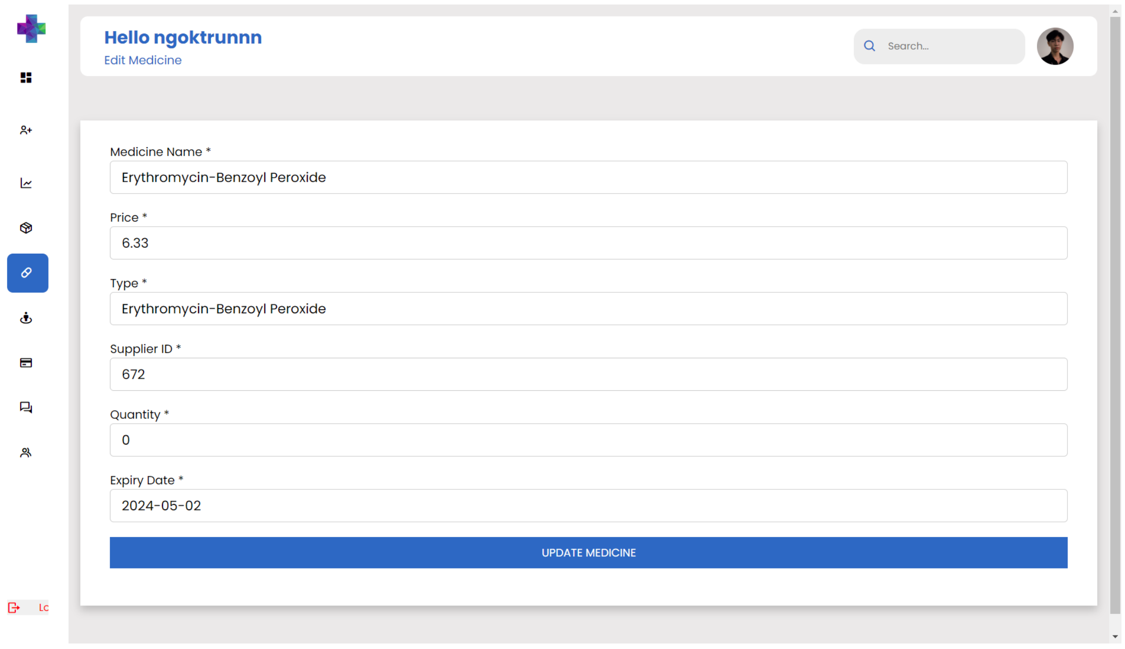
To navigate through the comprehensive list of medicines in your inventory, the "View Medicine" page includes pagination controls at the bottom. You can use these controls to move between pages, ensuring you have access to the full details of your available pharmaceutical stock.



ID	Name	Price	Type	Supplier ID	Quantity	Expiry Date	Actions
1	Erythromycin-Benzoyl Peroxide	6.33	Erythromycin-Benzoyl Peroxide	672	0	2024-05-02	Edit Remove
2	Hamster Epithelium	3.84	Hamster Epithelium	928	51	2024-04-16	Edit Remove
3	sodium fluoride	7.34	CTx4 Gel 1100	758	94	2024-12-05	Edit Remove
4	glycopyrrolate	4.77	Glycopyrrolate	476	2	2025-06-27	Edit Remove
5	ENSULIZOLE	2.53	RITE AID RENEWAL	446	95	2025-04-02	Edit Remove
6	Ferrous Sulfate	7.93	Ferrous Sulfate	271	91	2024-05-14	Edit Remove
7	simvastatin	9.34	SIMVASTATIN	868	28	2025-06-12	Edit Remove
8	Metformin Hydrochloride	7.4	Metformin Hydrochloride	609	95	2024-11-13	Edit Remove
	ALUMINUM ZIRCONIUM						

Figure 8.13: View Medicines

The "Edit Medicine" feature in the pharmacy management system allows authorized users to update the details of an existing medicine record. This functionality is crucial for maintaining accurate and up-to-date information about the medicines stocked and managed by the pharmacy. The image shows a form with various fields related to the medicine, including the name, price, type, supplier ID, quantity, and expiry date. Users can modify these details as needed, ensuring that the pharmacy's medicine data remains current and reliable. This is particularly important for tasks like inventory management, ordering, and dispensing of medicines to patients.



Medicine Name *
Erythromycin-Benzoyl Peroxide

Price *
6.33

Type *
Erythromycin-Benzoyl Peroxide

Supplier ID *
672

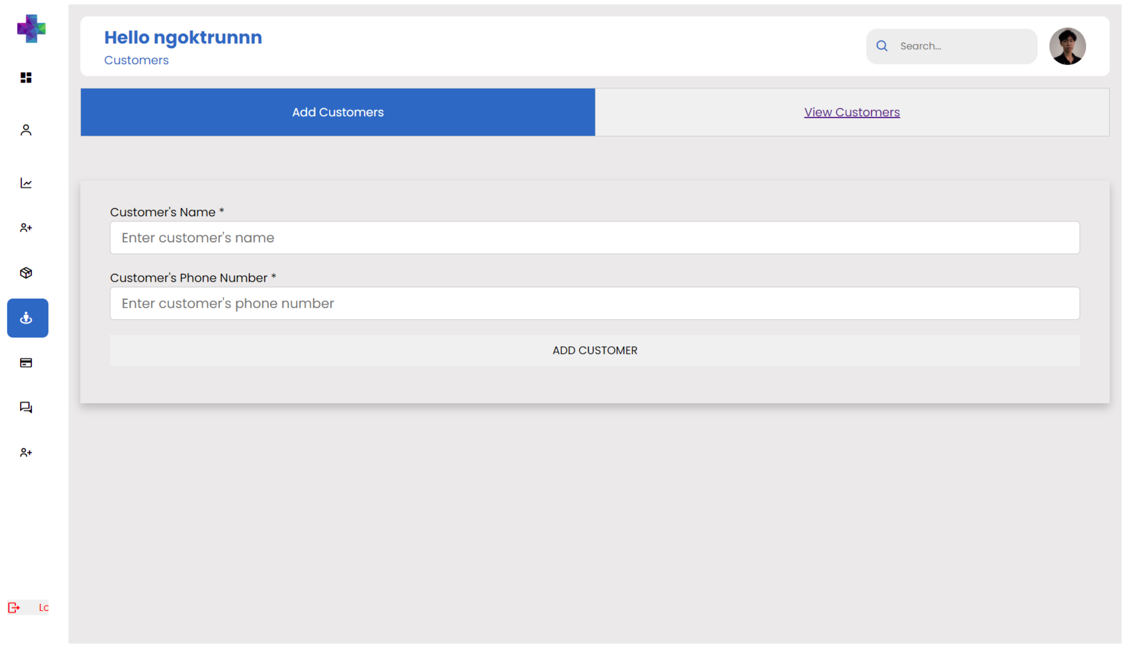
Quantity *
0

Expiry Date *
2024-05-02

Figure 8.14: Edit Medicines

8.6 Customers Management

The Customers section is the central hub for managing all the client information for your pharmacy. This is where you can add new customers, view their details, and make any necessary updates or changes to their profiles. To add a new customer, simply click the "Add Customers" button, which will bring up a form to enter the customer's name and phone number. Once you've filled out these required fields and clicked "ADD CUSTOMER", the new client record will be saved in the system.



Customer's Name *

Enter customer's name

Customer's Phone Number *

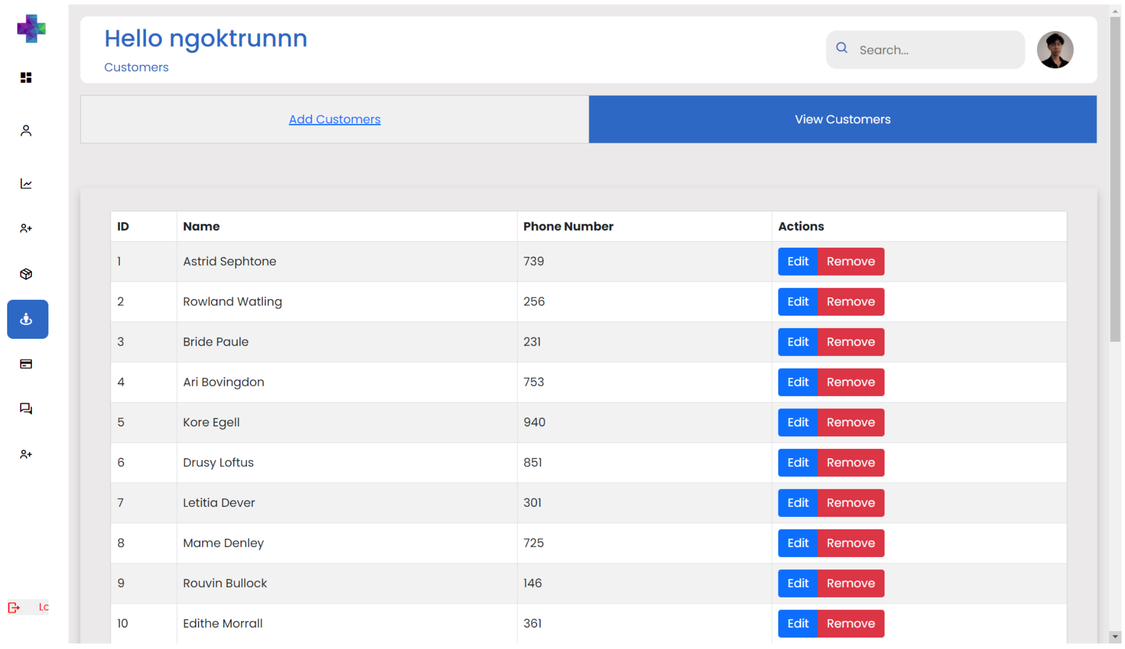
Enter customer's phone number

ADD CUSTOMER

Figure 8.15: Add Customers

The ”View Customers” page then displays a comprehensive list of all registered customers, showing their ID, name, and phone number. From this view, you can easily edit a customer’s name, phone number, or other details by clicking on their information, or remove a customer’s account entirely if needed. The Customers section gives you full control over maintaining an accurate and up-to-date database of your pharmacy’s client base, allowing you to onboard new patrons, update their contact info, and manage each customer.

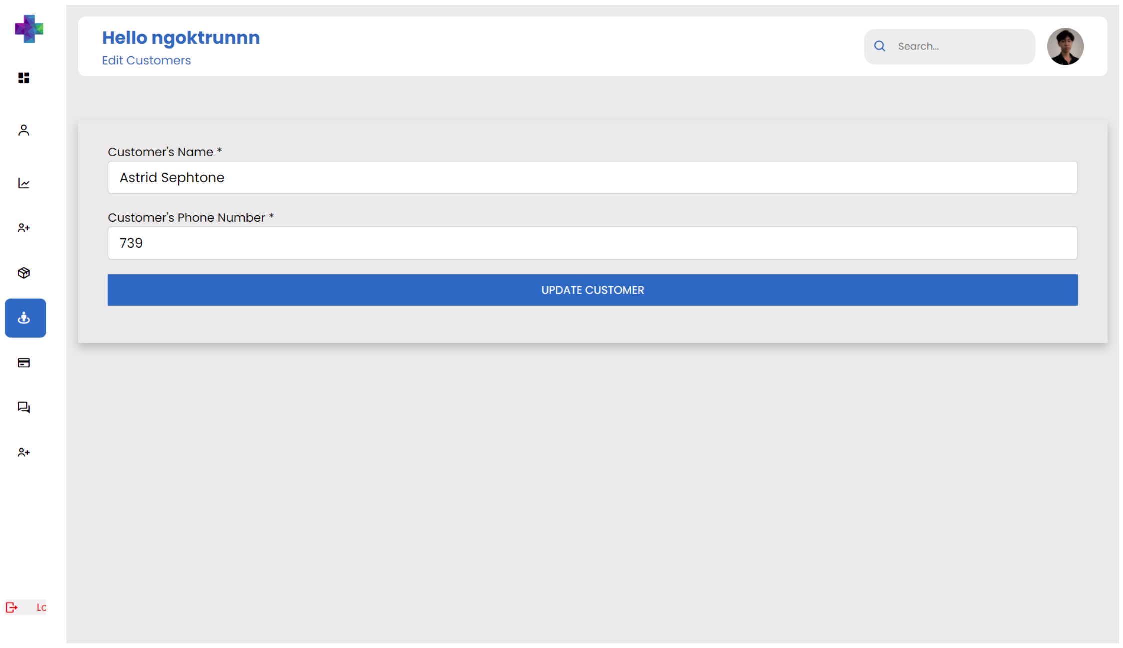
When utilizing this search functionality, users can input keywords related to the customer name, contact details, or any other relevant criteria. The system will then instantly filter and display the matching customer records, helping users quickly locate and access the details they need about specific patients or clients.



ID	Name	Phone Number	Actions
1	Astrid Sephtone	739	Edit Remove
2	Rowland Watling	256	Edit Remove
3	Bride Paule	231	Edit Remove
4	Ari Bovingdon	753	Edit Remove
5	Kore Egell	940	Edit Remove
6	Drusy Loftus	851	Edit Remove
7	Letitia Dever	301	Edit Remove
8	Mame Denley	725	Edit Remove
9	Rouvin Bullock	146	Edit Remove
10	Edithe Morrall	361	Edit Remove

Figure 8.16: View Customers

The "Edit Customers" feature in the pharmacy management system allows authorized users to update the details of an existing customer record. This functionality is crucial for maintaining accurate and up-to-date information about the pharmacy's patients. The image shows a form with fields for the customer's name and phone number. Users can modify these details as needed, ensuring that the pharmacy's customer data remains current and reliable. This is particularly important for tasks like order history, communication, and providing personalized service to the patients.


Hello ngoktrunnn

Edit Customers

Customer's Name *

Astrid Sephtone

Customer's Phone Number *

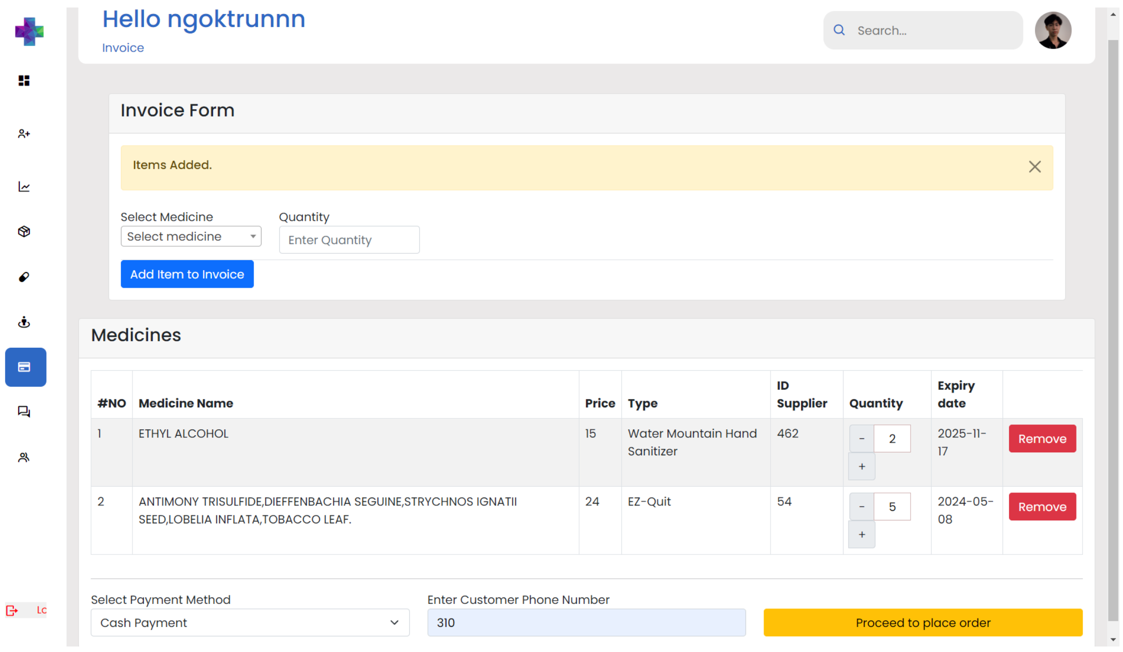
739

Figure 8.17: Edit Customers

To move between pages and view the comprehensive list of customers, the ”View Customers” section includes pagination options at the bottom of the page.

8.7 Generating an Invoice

The pharmacy management system's ”Invoice” section offers a comprehensive and user-friendly interface for creating and managing customer invoices. THe pharmacist can easily add items to an invoice by selecting the desired medicine from the dropdown menu, entering the quantity, and clicking the ”Add Item to Invoice” button. The application then populates the ”Medicines” table with the selected items, including details such as the medicine name, price, type, supplier ID, quantity, and expiry date. Users can remove individual items from the invoice by clicking the ”Remove” button next to each one. Once the items have been added to the invoice, users can select the payment method from the dropdown menu and enter the customer's phone number. After completing these steps, users can click the ”Proceed to place order” button to finalize the invoice and complete the transaction.


Hello ngoktrunnn

Invoice

Search...

Invoice Form

Items Added.

Select Medicine: Select medicine

Quantity: Enter Quantity

Add Item to Invoice

Medicines

#NO	Medicine Name	Price	Type	ID Supplier	Quantity	Expiry date	
1	ETHYL ALCOHOL	15	Water Mountain Hand Sanitizer	462	- 2 +	2025-11-17	Remove
2	ANTIMONY TRISULFIDE,DIEFFENBACHIA SEGUINE,STRYCHNOS IGNATII SEED,LOBELIA INFLATA,TOBACCO LEAF.	24	EZ-Quit	54	- 5 +	2024-05-08	Remove

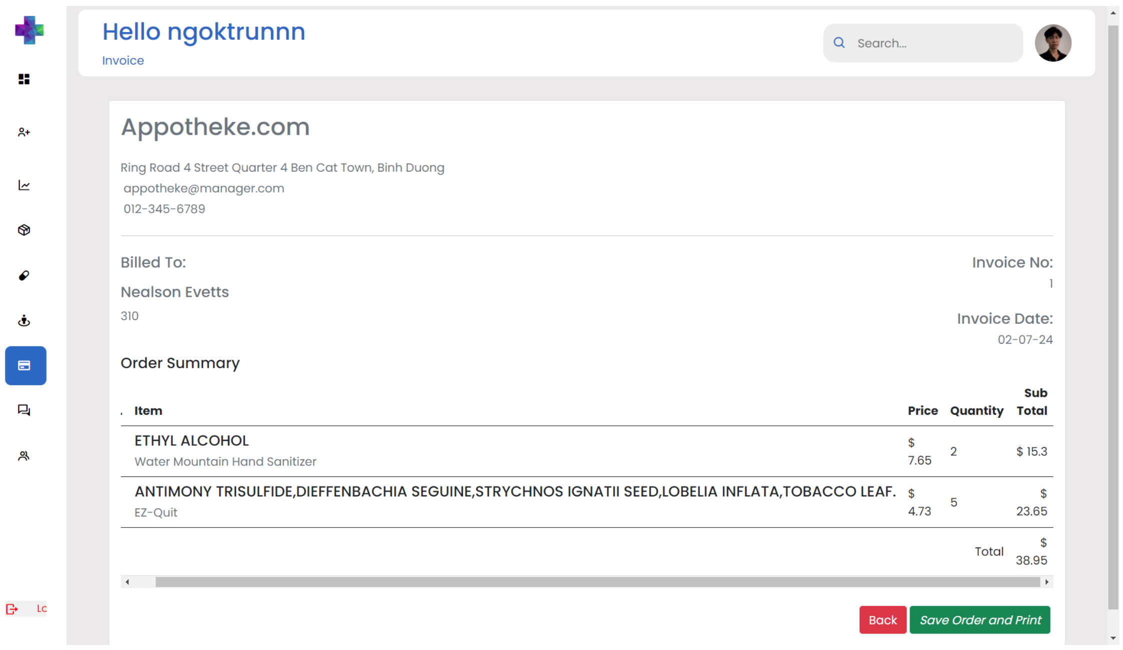
Select Payment Method: Cash Payment

Enter Customer Phone Number: 310

Proceed to place order

Figure 8.18: Generate Invoice

The system then displays a detailed summary, including the pharmacy information, the customer's billing details, the invoice number, and the invoice date. The "Order Summary" section provides a breakdown of the added items, showing the item number, name, price, quantity, and subtotal for each. The total amount for the entire invoice is also clearly presented at the bottom.


Hello ngoktrunnn

Invoice

Search...

Appotheke.com

Ring Road 4 Street Quarter 4 Ben Cat Town, Binh Duong
appotheke@manager.com
012-345-6789

Billed To:
Nealson Evetts
310

Invoice No:
1

Invoice Date:
02-07-24

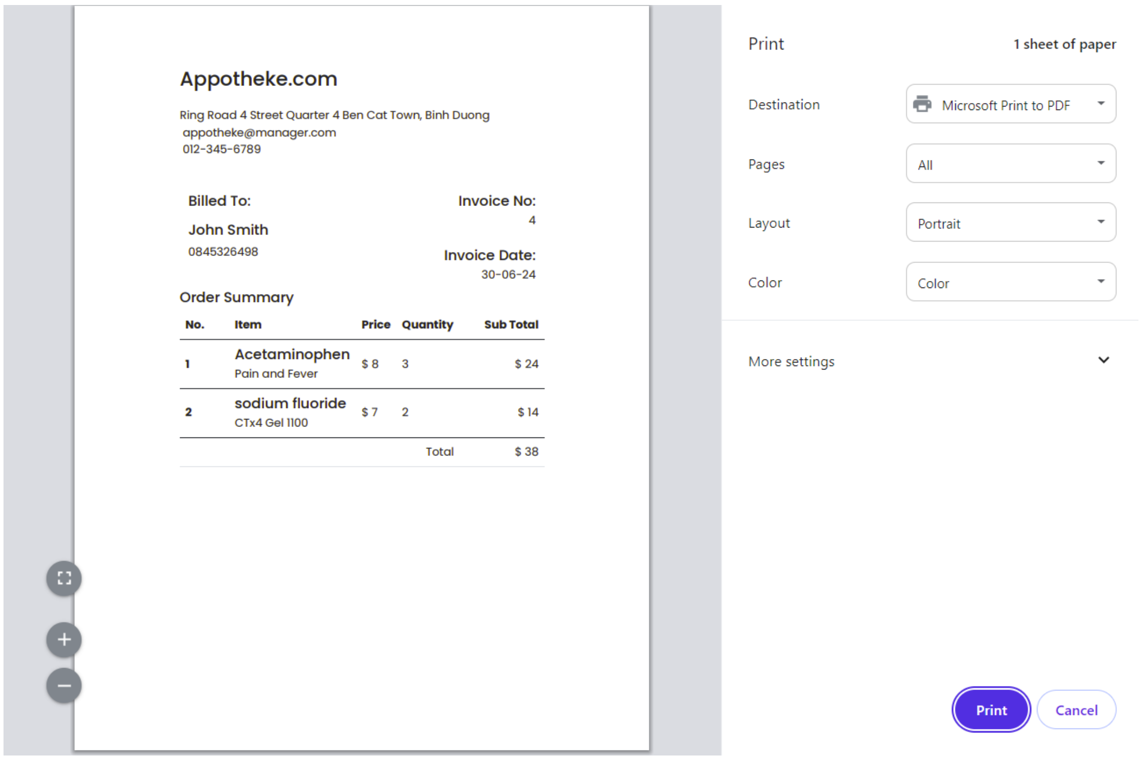
Order Summary

.	Item	Price	Quantity	Sub Total
	ETHYL ALCOHOL Water Mountain Hand Sanitizer	$ 7.65	2	$ 15.3
	ANTIMONY TRISULFIDE,DIEFFENBACHIA SEGUINE,STRYCHNOS IGNATII SEED,LOBELIA INFLATA,TOBACCO LEAF. EZ-Quit	$ 4.73	5	$ 23.65
			Total	$ 38.95

Back Save Order and Print

Figure 8.19: Invoice summary

For printing the invoice, the application offers a user-friendly print preview, where users can configure the settings, such as the destination, pages, layout, and color, before generating the final document. The clean and organized invoice layout ensures a professional and consistent presentation of the customer's order details.



Appotheke.com

Ring Road 4 Street Quarter 4 Ben Cat Town, Binh Duong
appotheke@manager.com
012-345-6789

Billed To:
John Smith
0845326498

Invoice No: 4

Invoice Date: 30-06-24

Order Summary

No.	Item	Price	Quantity	Sub Total
1	Acetaminophen Pain and Fever	$ 8	3	$ 24
2	sodium fluoride CTx4 Gel 1100	$ 7	2	$ 14
			Total	$ 38

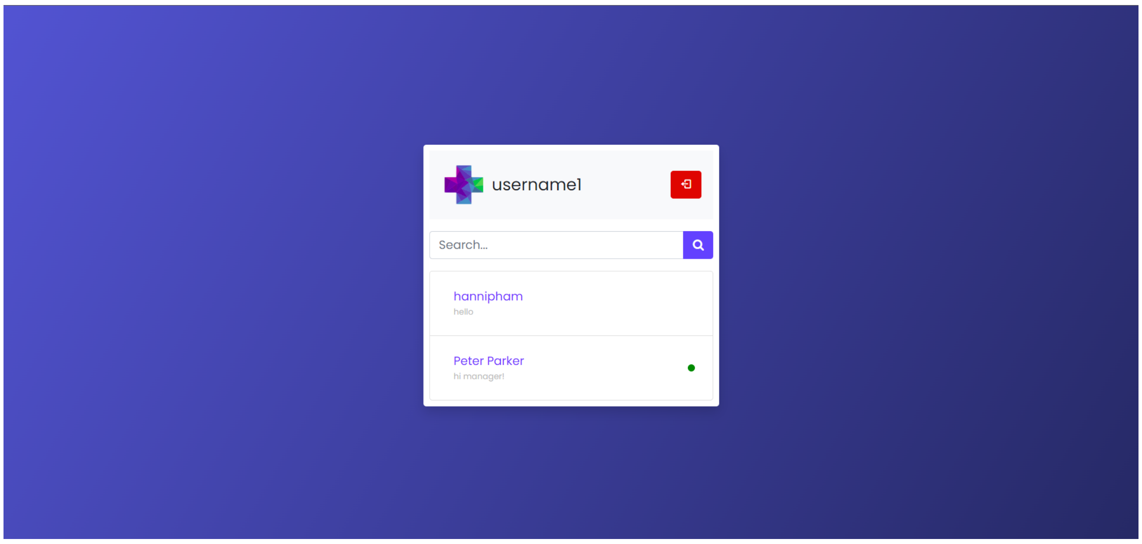
Print — 1 sheet of paper

Destination: Microsoft Print to PDF
Pages: All
Layout: Portrait
Color: Color

Figure 8.20: Printing Page

8.8 Using The Chat Function

The "Messaging" section provides an internal chat function, allowing you to communicate with your staff members. You can initiate new conversations or view existing threads, and the chat history is stored for future reference. Employees can also use this chat feature to communicate with each other, enabling seamless collaboration and information sharing across the organization.

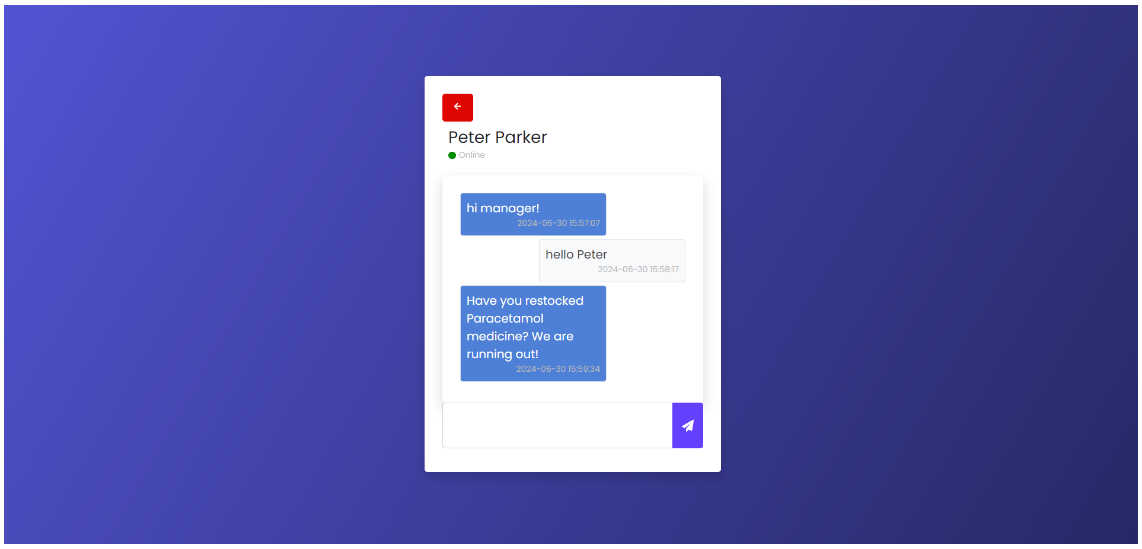


hannipham
hello

Peter Parker
hi manager!

Figure 8.21: Chat Design

In the second image, the chat window shows an ongoing conversation between the user "Username1" and another user. The chat bubble on the left indicates that Username1 has sent a message, with the timestamp displayed below. The chat interface appears clean and intuitive, providing a seamless experience for users to exchange messages and communicate effectively. The presence of the "Send" button in the chat window suggests that users can easily input and transmit their messages, further enhancing the interactive nature of the chat function. Overall, the chat feature showcased in these images appears to be a well-designed and user-focused component, enabling efficient and engaging communication between users within the application.


Peter Parker
Online

hi manager!
2024-06-30 15:57:07

hello Peter
2024-06-30 15:58:17

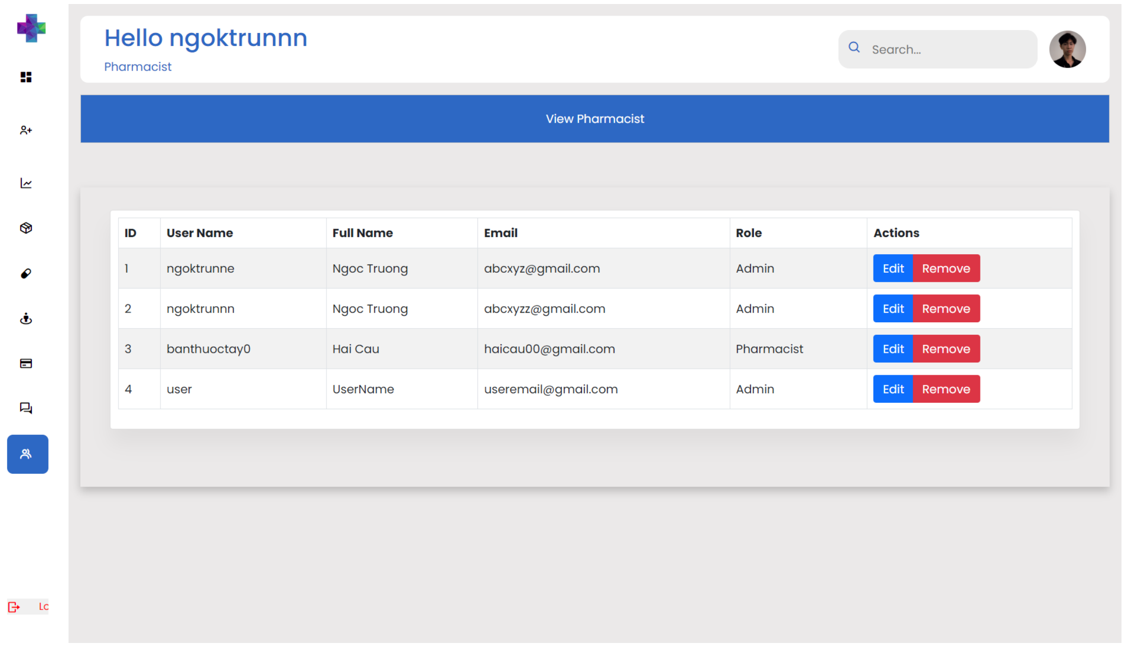
Have you restocked Paracetamol medicine? We are running out!
2024-06-30 15:59:34

Figure 8.22: Chat Message

8.9 Employee Management

The ”Employee” section of the pharmacy management system provides a comprehensive interface for managing the user accounts of the pharmacy staff. This section displays a table that lists all the employee accounts, including their user ID, username, full name, email address, and assigned role. The ”Remove” option enables the removal of an employee from the system if necessary.

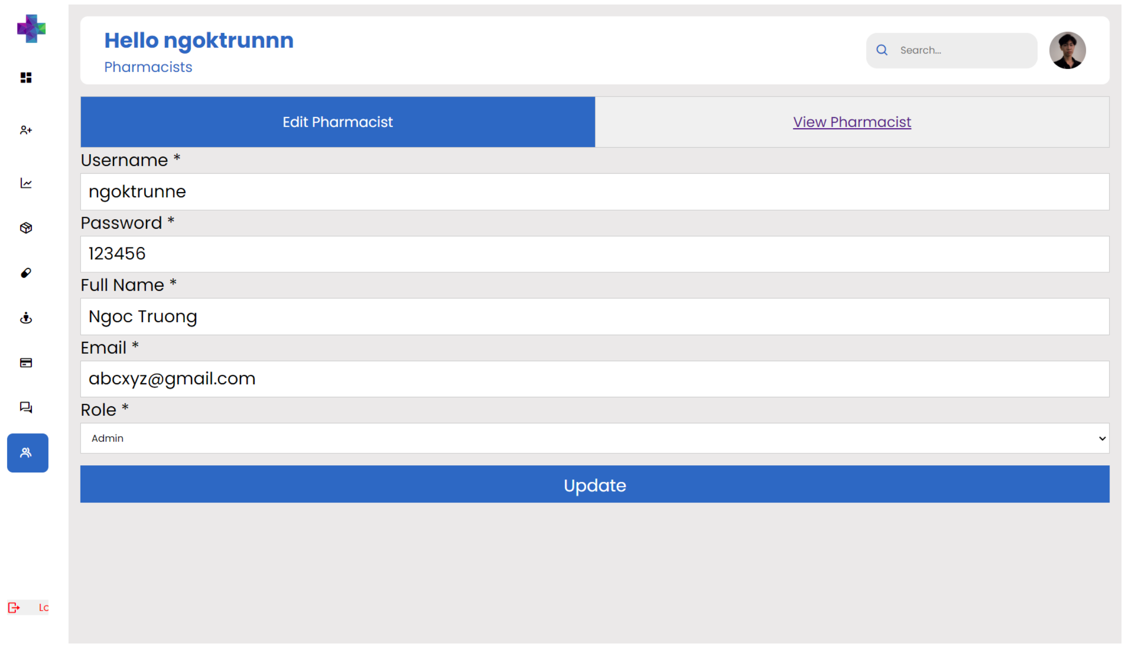
The search capabilities integrated into the ”View Employees” module enhance the overall user experience, empowering pharmacy administrators to manage their personnel more effectively. This, in turn, contributes to the efficient and secure operation of the pharmacy management system, ensuring that employee-related data is readily accessible and properly managed.



ID	User Name	Full Name	Email	Role	Actions
1	ngoktrunne	Ngoc Truong	abcxyz@gmail.com	Admin	Edit Remove
2	ngoktrunnn	Ngoc Truong	abcxyzz@gmail.com	Admin	Edit Remove
3	banthuoctay0	Hai Cau	haicau00@gmail.com	Pharmacist	Edit Remove
4	user	UserName	useremail@gmail.com	Admin	Edit Remove

Figure 8.23: View Pharmacist

The ”Edit Pharmacist” feature in the pharmacy management system allows authorized users to update the details of an existing pharmacist record. This functionality is crucial for maintaining accurate and up-to-date information about the pharmacy’s staff members who are responsible for dispensing medications and providing patient care. The image shows a form with fields for the pharmacist’s username, password, full name, email, and role. Users can modify these details as needed, ensuring that the pharmacy’s pharmacist data remains current and reliable. This is particularly important for tasks like managing user access, tracking medication dispensing activities, and ensuring appropriate staff roles and responsibilities.



Username *
ngoktrunne
Password *
123456
Full Name *
Ngoc Truong
Email *
abcxyz@gmail.com
Role *
Admin

Figure 8.24: Edit Pharmacist

8.10 Logout Button

At the end of the sidebar, users will find the Logout button. This feature serves as a critical security measure, allowing users to securely exit the application and return to the login page when their work is complete.

The inclusion of a prominent Logout button is an essential security best practice for the pharmacy management application. It empowers users to quickly and securely exit the system, protecting sensitive information and ensuring that only properly authenticated personnel can access the software's features and data. This logout functionality is a key safeguard that supports the overall security and compliance requirements of the pharmacy's operations.

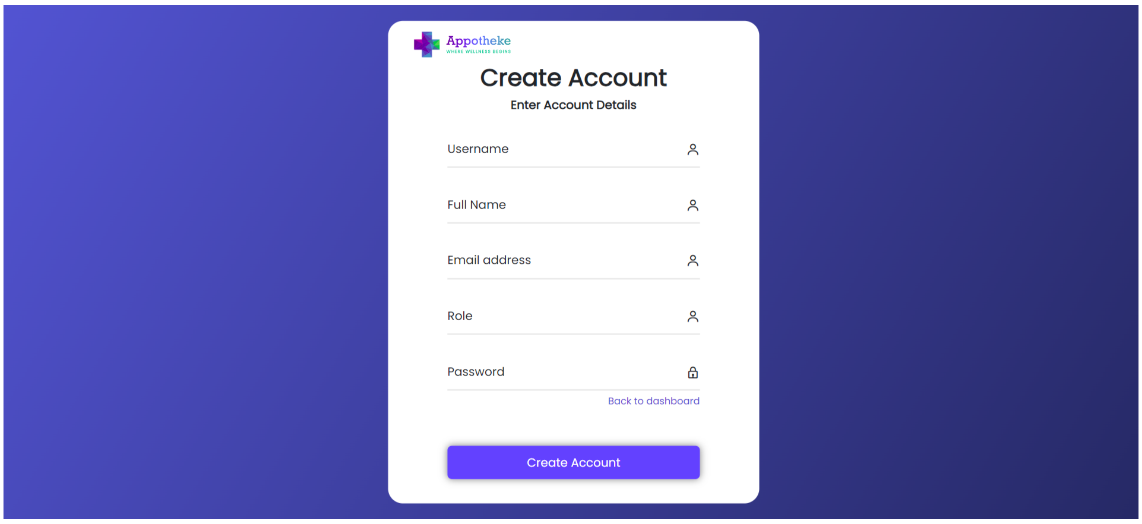


Figure 8.25: Create Account

Chapter 9

Conclusion

The Pharmacy Management System developed by our team is designed to be a comprehensive, efficient, and user-friendly solution for the pharmaceutical industry. By leveraging the robust backend of PHP, the intuitive frontend powered by React, and the reliable MySQL database, the system offers an effective and accessible approach to managing the day-to-day operations of a pharmacy.

The key features of the Pharmacy Management System, including the centralized dashboard, user authentication, role-based access control, and data interaction functions, collectively address the diverse internal needs of pharmacists, and administrators. These features enable pharmacies to streamline their inventory management, streamline financial processes, enhance internal collaboration, and ultimately provide a superior customer experience.

The project's primary objectives is to develop a scalable, and feature-rich Pharmacy Management System that would improve the overall efficiency and productivity of pharmaceutical stores. The system is expected to deliver three main benefits to its users, including improved inventory control and medication availability, enhanced financial management and reporting capabilities, and streamlined communication and collaboration among pharmacy staffs.

As the Pharmacy Management System is deployed and adopted by pharmaceutical stores, it is anticipated to play a pivotal role in optimizing their operations, reducing costs, and enhancing the overall quality of patient care. The comprehensive and user-centric design of the system positions it as a valuable asset for the pharmaceutical industry, contributing to its success and growth in the years to come.