Project Report

Smart Safe

Submitted by

Muhammad Usman Ghani
2020-uam-2133
2020-2024

Muhammad Faizan Haider
2020-uam-2120
2020-2024

Fatima Shaheen
2020-uam-2130
2020-2024

Supervised by

Mr. Muhammad Yasir Khan

INSTITUTE OF COMPUTING
MNS-UNIVERSITY OF AGRICULTURE, MULTAN
PAKISTAN

FINAL APPROVAL

This is to certify that we have read this report submitted by ***Muhammad Usman Ghani, Muhammad Faizan Haider and Fatima Shaheen.*** It's our judgment that this report is of sufficient standard to warrant its acceptance by MNS-University of Agriculture, Multan for the degree of BS (Information Technology).

Committee:

1. **External Examiner** _____

   Dr. Khwaja Tehseen Ahmad
   Associate Professor
   Bahauddin Zakariya University Multan

2. **Supervisor** _____

   Mr. Muhammad Yasir khan
   Lecturer,
   Institute of Computing

.

3. **Head of Institute** _____

   Prof. Dr. Salman Qadri
   Director,
   Institute of Computing
   MNS-University of Agriculture, Multan

DECLARATION

This is to certify that ***Muhammad Usman Ghani (2020-uam-2133), Muhammad Faizan Haider (2020-uam-2120) and Fatima Shaheen (2020-uam-2130)*** Session (2020-2024) have worked on and completed their software project "**Smart Safe***"* at the Institute of Computing, MNS-University of Agriculture, Multan, in partial fulfillment of the requirements for the degree of BS (Information Technology).

Date: _____

Signature: _____

Muhammad Usman Ghani

Reg. No. 2020-uam-2133

Signature: _____

Muhammad Faizan Haider

Reg. No. 2020-uam-2120

Signature: _____

Fatima Shaheen

Reg. No. 2020-uam-2130

DEDICATION

To my Loving Parents

This study is wholeheartedly dedicated to our beloved parents; in every page I turn, in every lesson I learn, and in every achievement, I reach, there's a reflection of your love, sacrifice, and unwavering support. This dedication is not just about my study but a tribute to your boundless encouragement and relentless belief in me. From the earliest moments of my academic journey, you've been my guiding star, illuminating the path with your wisdom and love. Your sacrifices have laid the foundation upon which I build my dreams, and your encouragement has fueled my determination to pursue excellence. Thank you, Mom and Dad, for being my biggest cheerleaders, my strongest supporters, and my greatest inspirations. This journey is as much yours as it is mine, and I dedicate my study to you with all my love and gratitude.

ACKNOWLEDGMENT

Gratitude is not only the heart and essence of Islam; it is also the key to attracting abundance, prosperity, peace, and success in one's life. I can say that Gratitude and Gratefulness are the most important aspects of Islam. As Muslims, we should always be grateful to Almighty Allah for all the blessings that He has bestowed upon us.

In the Holy Quran, Allah Almighty says: **"And remember! Your Lord caused to be declared: If ye are grateful, I will add more unto you" (Quran, 14:5-7)**. I am very grateful to Almighty Allah because without His support, I might not have been able to accomplish this task. I find no words to express my deepest sense of gratitude to the Almighty Allah, the most gracious, compassionate, and beneficent, and to His Prophet Muhammad (Peace Be upon Him), who is a true torch of guidance for all humanity forever.

I am greatly obliged to Allah, by whose grace I have been able to accomplish this project successfully. I would like to express my sincere gratitude, especially to my parents, for their invaluable support, advice, guidance, and encouragement throughout my life, especially throughout the course of this project. I express my gratitude to the Head of Department**, Prof. Dr. Salman Qadri**, who made tireless efforts to provide us with all facilities throughout the academic session.

Special thanks are extended to our final year project supervisor, **Mr. Muhammad Yasir Khan,** whose contributions in stimulating ideas and providing praise helped us to organize our project, especially in writing this report. We are grateful for her guidance and support throughout this endeavor.

PROJECT BRIEF

PROJECT NAME	Smart Safe
UNIVERSITY NAME	Muhammad Nawaz Sharif University of Agriculture, Multan.
UNDERTAKEN BY	Muhammad Usman Ghani, Muhammad Faizan Haider and Fatima Shaheen.
SUPERVISED BY	Mr. Muhammad Yasir Khan.
STARTING DATE	February 1,2024
COMPLETION DATE	May 29, 2024
COMPUTER USED	Intel(R) Core (TM) i3-5005U CPU 2.00 GHz, 8GB RAM, 128GB Solid State disk.
OPERATING SYSTEM	Microsoft Window 10 Pro.
SOURCE LANGUAGE(S)	HTML, CSS, PHP, JavaScript.
DBMS USED	My-SQL
Libraries and Frameworks	Boot Strap, Font Awesome, Material Design Icon, Chart.js

PLAGIARISM UNDERTAKING

I solemnly declare that the work presented in the report titled ***"Smart Safe"*** is solely our work with no significant contribution from any other person. Small contribution/help wherever taken has been duly acknowledged and that complete report has been written by us.

I understand the zero-tolerance policy of the HEC and MNS-University of Agriculture, Multan towards plagiarism. Therefore, I as an Author of the above titled report declare that no portion of my report has been plagiarized and any material used as reference is properly referred/cited.

I undertake that if I am found guilty of any formal plagiarism in the above titled report even after award of the degree, the University reserves the rights to withdraw/revoke my degree and that HEC and the University has the right to publish my name on the HEC/University Website on which names of students are placed who submitted plagiarized report.

Student /Author Signature: _____

Name: Muhammad Usman Ghani

Student/Author Signature: _____

Name: Muhammad Faizan Haider

Student/Author Signature: _____

Name: Fatima Shaheen

ABSTRACT

These days, the pharmaceutical industry is one of the most dynamic sectors, and it requires a web-based platform for innovative approaches to improve business performance and the quality of services provided to patients. Realizing this need, this document introduces “Smart Safe,” a web-based platform that has the potential to change the face of pharmaceutical store management. Smart Safe incorporates innovative technology in some activities of pharmacy, such as sales, inventory, and billing systems. Some benefits include easy access to information about medicines and a well-organized interface that allows for ordering and sales monitoring. Smart Safe has features like graphical analysis and integrated QR code payment to ensure convenience and security are greatly enhanced. Further, it also provides automatic notifications of low-stock and expiring products to avoid the risks of low-stock or expired products. In addition, it allows for the interaction of multiple users who can easily be in touch with one another within the pharmacy setting.

TABLE OF CONTENTS

Chapter 7

Conclusion & References

LIST OF FIGURES

LIST OF TABLES

Chapter 1

INTRODUCTION

Introduction

“Smart Safe” is a web-based platform for pharmaceutical stores that helps to increase the accuracy, safety, and efficiency of the store. This website enables users to purchase medicine online. The design allows users to search easily for medicines and find such important information as categories of medicine, packing, price, and dosage. The features facilitate ease in placing orders, hence affording both time and money. In this project, manual errors will be removed by using computerization to manage the sales, inventory, and billing of the medical store. Due to it being a web-based platform, it implies that many users are engaged, making it efficient for everyone.

“Smart Safe” covers all possible medicine categories, including their storage locations, suppliers, purchases, sales, and billings. All these modules play the role of facilitating work in the pharmacy and ensuring accuracy and reliability. In addition, the platform consists of exciting features like graphical analytics that help the owners capture daily, weekly, and monthly sales data. This enables administrators to produce reports or monitor overall activities in the pharmacy and confirm the existence of systemic responsibility, hence improving the operation’s transparency and responsiveness.

It also provides alert notifications on medicines that are low in stock or are almost expiring, which enables us to restock appropriately. The integration of QR code payment functionality enhances transaction convenience and security, as well as ensuring the safety of users’ payments. Smart Safe has revolutionized the way medical stores are managed and enhanced the faithful and efficient delivery of health services.

1.1 Motivation

The reason for undertaking the project of developing the Smart Safe lies in the fact that various issues relate to the function of pharmacies. As a final-year information technology student, it is a privilege to be able to employ our knowledge and talent to develop a project that may well turn out to be beneficial and revolutionary in the improvement of the healthcare sector. Pharmacies are crucial to the healthcare sector since they are involved in the provision of essential services to patients and also ensure

that medicines are used appropriately without necessarily harming the patient. The problems above can be addressed by developing a “smart safe” system that aims at providing an efficient solution to assist pharmacies in managing their organizations. The system will assist the pharmacies in realizing increased levels of productivity and efficiency, which are vital to enhancing the quality of patients.

1.2 Problem Statement

The pharmacy administration has kept a paper record in the filing cabinet. Managing a large pharmacy with paper records would be slow and difficult to track inventories in terms of medicine in store and quantity of medicine available based on expiration dates, categories, and their functions. The pharmacist has to order the medicine to restock the already depleted stock. Also, the medicine order is being done manually. A lot of time is allocated for order writing as the pharmacist has to go through the stock balance and estimate the order amount based on the data. This project will inform pharmacists about medicines that are close to expiration, prevent these medicines from being sold, and also provide solutions to the problems mentioned earlier.

1.3 Scope of the project

The scope of this project is to create a general web-based “Smart Safe” system that will modernize the processes, and operations of a pharmacy. The system will allow the pharmacy to effectively manage inventory, track sales, and process. The smart safe system developed using modern web technologies such as HTML, CSS, JavaScript, and PHP. The system will be designed to be user-friendly and natural, with a clean and modern interface that makes it easy for pharmacists and other staff to explore and use. One of the key features of the system will be inventory management, which will allow the pharmacy to keep track of all medications, supplies, and expiration information in real-time. Finally, the system will include a sales tracking module, which will allow the pharmacy to track sales and other key metrics. This will allow the pharmacy to make data-driven decisions and improve its processes for determined efficiency and profitability.

1.4 Aims & Objectives of the Project

Our aim is to address these challenges by developing a comprehensive web-based platform that enhances the accuracy, safety, and efficiency of pharmaceutical store operations. The system will provide detailed medicine information, streamline inventory management, facilitate online ordering and billing, generate insightful reports, and improve customer engagement. By integrating advanced features and an admin panel, "Smart Safe" will ensure better inventory tracking, customer support, and overall operational transparency.

- To develop an online pharmacy management system.
- To provide a user-friendly platform for users to access detailed medicine information and order effectively online.
- To improve inventory management by tracking real-time data on medicine stock, including manufacturing and expiration dates.
- To automate order processing and reporting to reduce errors and save time.
- To reduce manual errors, the system aims to digitize everyday tasks such as managing sales, purchases, and inventory.
- To provide suitability, the system monitors users about medicines and enables online purchases.
- To improve efficiency, the system enables the creation of medicine sales bills and centralizes store data. To promote teamwork, the system allows many users to work concurrently.
- To ensure accuracy, the system tracks medicine location details, supplier data, and live inventory.
- To simplify management, the system offers numerous modules, such as Medicine Category and Location Rack.

1.5 Main Contribution to Society

- **Accessibility**

  With the comprehensible interface offered by our web-based pharmacy store management system, people can easily access complete medicine information and make purchases online from any location.

- **Efficiency**

  The system reduces human errors and improves operational efficiency by digitizing all aspects of medical store actions, including sales, inventory management, and billing.

- **Collaboration**

  Because the platform is web-based, it makes it easy for many users to work together, which helps teamwork and increases productivity**.**

- **Comprehensive Solution**

  Our website provides a full solution to modernize pharmacy actions, with modules addressing locations for racks, suppliers, purchases, sales, and billing.

- **Informed Decision-Making**

  The platform helps pharmacy administrators make well-informed decisions and manage their actions for better healthcare outcomes by centralizing store data and offering real-time understandings of sales, purchases, and inventory levels.

Summary

The “Smart Safe” project proposes the development of a web-based platform for pharmaceutical stores with the goal of improving accuracy, safety, and efficiency. It enables the purchase of medicines online, manages stock, and provides information in real-time. The project concerns issues such as manual record-keeping and is expected to enhance productivity and effectiveness in the functioning of pharmacies. Some of the main functions are quick search options, inventory management, and QR payments. It enhances organizational transparency and helps in decision-making within the healthcare sector.

Chapter 2

LITERATURE REVIW

Literature Review

Pharmacy management requires a lot of attention to detail because a small mistake by one person can result in the loss of a person's life. A pharmacy also requires various tasks that include stock management, ensuring that the medicine in the pharmacy is up-to-date, and managing the bill. These days, due to the support and quality services of the pharmacy, customers Pharmacies with a large number of people have a heavy workload, which affects the smooth running of the pharmacy as a result of the traditional way of running the business. In this period, when the number of customers superior to the pharmacy is increasing, the workload of pharmacists increases. This makes it difficult for pharmacists to take care of customers in the short-term using traditional methods. Meanwhile, pharmacists need to retain their customers and ensure that they are provided with the right medicine and quality services. This pharmacy management system is concerned with managing the stock of medicine and selecting the appropriate medicine required by the customers. The foundation of the profession of pharmacist is quality care and its subsequent implications for clinical supervision and control in pharmacy activities. This pharmacy management system will help solve various problems faced by pharmacists.

2.1 Existing System

However, in the current situation, there are many pharmacies that have not moved their operations from conventional manual methods to computerized systems as yet. There are a number of processes, including inventory management, monitoring, billing, and procurement processes, that, due to their nature, are likely to be subjected to human error. Data by hand could incur errors and faults like the wrong amount of stock in storage, lost sales reports, and delays in processing orders. However, these usually fail to integrate and therefore do not enable staff to cope with them efficiently, such as keeping an inventory, selling, and producing an accurate financial report.

Secondly, the lack of centralized officials leads to restricted monitoring of the activity of the pharmacy on an imminent basis as well. This may bring up problems that are related to, for example, stock shortages, stocks remaining on shelves, and outdated medicines after expiration. This is a very complicated and inefficient sales cycle.

The same manual system carries the disadvantage of not having the convenience of online medication buying, which is apparently liked among consumers today. Payment modes are mostly limited to cash or card transactions at the front desk, which are out dated and can't provide payment security through modern online processes.

Davago is a modern pharmacy and wellness chain that offers a comprehensive range of healthcare services in Multan. This includes the sale of prescription medicines, over-the-counter products, and various wellness items. Their system integrates technology to provide a seamless and convenient pharmacy experience.

2.2 Related Projects

In order to develop smart safes, we looked up several similar systems. In the existing system, there are many vulnerabilities, such as the lack of a good authentication method to authorize users. There are also many drawbacks, such as the online delivery of medicine to the user at the doorstep of the customer. E-Pharmacy Platforms These are platforms that enable users to buy medicines over the internet. Examples include 1 mg, Net Meds, and Pharm Easy. They generally contain rich information about medicines, such as price, dosage, and stock status, but at the same time lack rich system integration with the actual physical pharmacy. Inventory Management Systems Others, like Marg ERP and Go Frugal, offer more solutions specifically tailored to pharmacy inventory management. These systems assist in monitoring stocks and sales but do not provide additional functions like e-commerce or payments via QR codes. So, after analyzing such kinds of gaps in the existing system, we decided to design a very compact and integrated system that contains the functions of search medicine, add-to-cart medicine, monitoring daily or weekly sales, and monthly sales. Alert notification for medicine expiry, medicine supplier management, medicine categories, location racks, purchases, and billing. QR payment method.

2.3 Features Comparison

Sr No	Features	Existing Manual System	Davago Pharmacy System	Dawaii.pk	Proposed Solution
1	Online Medicine Purchase	No	Yes	No	Yes
2	Detailed Medicine Information	No	Yes	Limited	Yes
3	Inventory Management	Manual	No	Yes	Yes
4	Sales Tracking	Manual	No	Yes	Yes
5	Billing and Invoicing	Manual	No	Yes	Yes
6	Real Time Stocks – Updates	No	No	Yes	Yes
7	QR Payment Method	No	No	No	Yes
8	User Access Control	Limited	No	Limited	Yes
9	Reporting and Analytics	Manual	No	Yes	Yes

Figure of Features Comparison 2.1

2.4 Proposed Solution

The Smart Safe web-based platform is a potential solution for the problems associated with the use of multiple systems to streamline the operations of different pharmacies.

- **Online Medicine Search and Purchase**

  A customer can quickly find the medicine he/she is interested in and view the information about available forms, packaging, prices, dosage forms, and indications for the necessary medicine. This will also mean that users can order online.

- **Admin Panel**

  An admin panel shall allow the pharmacy managers to monitor and control operations, generate various reports, keep an eye on employees, and promote the systemic accountability. Managers have the opportunity to set specific permissions that specify the rights of each employee.

- **Inventory Management**

  The system will include computerized management of inventory by recording stock levels, requesting replenishment for low stock or outdated drugs, and maintaining information about suppliers.

- **Sales and Billing**

  Selling procedures will be facilitated by use of electronical billing and invoicing. It will also help in the generation of sales bills and centralizing the stores and them records for quick monitoring.

- **QR Code Payment Integration**

  For increased security and user friendliness during transactions the system will be integrated with the use of QR code payments.

- **Real-time Data and Reporting**

  The centralized data storage will also enable the firms to have the real-time information regarding the sales, purchase, and the inventory levels for better decision-making. Combining these functions in a web-based platform will increase workflow effectiveness, minimize paperwork, and eliminate the possibility of human mistakes in this “Smart Safe” system by the pharmacy staff and customers. This approach covers all the aspects relating to pharmacy management and digitalizes the processes, thereby improving the entire performance and quality of services provided by the pharmacy.

Summary

Chapter 2 explains how manual pharmacy operational processes and systems come with inefficiency and inaccuracies and are prone to errors in the processes of stock control, billing, and customer service. It presents current e-pharmacy and inventory management systems, pointing out their drawbacks with regard to integration and functionality. To overcome the above challenges, the chapter designs a Smart Safe web-based platform solution that will offer online medicine search and purchasing, an admin interface for monitoring and control, computerized medicine management, bill generation and payment through QR code, and real-time reports to enhance service delivery in pharmacy businesses.

Chapter 3

SPECIFICATION & DESIGN

3.1 Data Gathering

During the data gathering phase, we conduct meetings with pharmacy management organizations to understand their existing pharmacy management system issues. Here are some of the questions we asked and a summary of their responses.

3.1.1 Questionnaires

1. How is the current pharmacy management system handled manually?

Response:

"Our pharmacy operations are currently managed manually through meticulous record-keeping, regular inventory checks, and personalized customer service to ensure efficient service delivery."

2.What are the common challenges or issues faced with the current Pharmacy management system?

Response:

Common challenges with our current pharmacy management system include manual errors in record keeping, difficulties in effectively managing inventory, and limitations in effectively meeting customer demands.

3. Are there any requirements that organizations want to that in new real estate management system?

- **Response**

  The organization expressed the following requirements:

- **Advanced inventory management**

  Real-time tracking and automated reordering optimize stock levels, reducing waste and ensuring medication availability.

- **Reporting and analytics**

  Detailed insights into key metrics empower informed decision-making, enhancing pharmacy performance and patient care.

- **Data security**

  pharmacy Management System is all about medicine information, inventory details, sales and record. All the information is necessary loss of any information can make a dire for your system. All the organizations want data security for their business.

3.2 Introduction to System Specification:

An SRS is a document that plans the client's hopes as well as those for the web-based platform that is currently under development. It is a necessary component of the web-based platform Development Lifecycle and is a related agreement and insurance policy between the client and the developer. A document or collection of documents called a System Requirements Specification (SRS) outlines the functionalities and characteristics of a software program. It's serious to thoroughly observe mutually the systems and the Web- based platform's non-functional requirements.

- **Functional Requirements**
- **Nonfunctional requirements**

3.2.1 Functional Requirements

There are some functional requirements which are given below

- **Registration and Login**

  Users, including customers, pharmacy staff, and administrators, must be able to register and log into the system strongly.

- **User Roles and Permissions**

  The system should describe roles (e.g., customer, pharmacist, admin) with specific approvals for accessing different functionalities.

- **Medicine Information Management**

  Medicine Database: The system must keep a complete database of medicines, with details such as name, form, price, dosage, manufacturer, and indications.

- **Search Functionality**

  Users should be able to search for medicines by name or category and view detailed information.

- **Stock Management**

  The system must track stock levels in real-time, and send alerting notifications to administrators when stocks are low or expired.

- **Supplier Management**

  Maintained detailed records of suppliers and managed to obtain orders.

- **Batch and Expiry Tracking**

  Track medicine batches and expiry dates to confirm the timely removal of out-of-date stock.

- **Order Management**

  Customers should be capable of adding medicines to their cart and orders online.

- **Billing**

  Produce electronic bills for purchases, including a full itemization of credited medicines.

- **Payment Integration**

  Integrate QR code payment functionality to facilitate secure online transactions

.

- **Sales Reports**

  Generate daily, weekly, and monthly sales reports for observing performance.

- **Inventory Reports**

  Provide reports on current stock levels, upcoming ending dates, and supplier performance.

- **User Activity Reports**

  Track actions of pharmacy staff for accountability and performance assessment.

- **Stock Alerts**

  Send notifications to administrators when stock levels fall below a predefined threshold.

- **Order Status Notifications**

  Inform customers about the status of their orders through email or SMS.

- **Expiry Alerts**

  Send alert notifications to the pharmacists about medicines that are nearing their expiry date.

- **My Cart**

  Our system will provide the service of a cart through which you may add medicine items that you want to buy that will make the balance of your bought item and also contain full information about your bought items.

- **Record Management**

  The proposed system offers great control on the records management whether it is a user record, admin record, or orders record.

- **Searching Bar**

  In this user may search for their medicine by name. If the searched medicine is not available, then the system will make an effort to find another searched medicine by the user.

- **Better Management**

  The proposed system will offer improved management than existing ones.

- **Long Term Goal**

  The long-term goal of this project is to provide all data related to medicines, and users in an electronic form, such that it could be used in a future production system for further tasks.

3.2.2 Nonfunctional Requirements

Nonfunctional requirements are all about the other operations of the project. It's all about how it looks, the level of safety it provides, and the operations that are performed.

- **Reliability**

  Software reliability is the possibility of failure-free operation of a computer program for a specified period in a specified environment. Reliability is a customer-oriented view of software worth. Set hopes for the application's availability, error tolerance, and ability to recover from failures. Outline requirements for system uptime and maximum suitable downtime. The proposed system is more consistent and easier.

- **Efficiency & Speed**

  The current system is not efficient. Our system is faster and much well-organized on the behalf of their performance. Our system identifies the expected response time, throughput, and resource utilization of the application under different capabilities under different workloads or user loads.

- **Security**

  The system should be secure so it can be disallowed from unauthorized access. For the safety of data, system, and user privacy we cannot understate the importance of safekeeping. Some require the necessary security measures and controls to keep the web-based platform from unauthorized access, data breaches, and malicious actions. This includes authentication, authorization, encryption, and safe communications protocols.

- **Availability**

  Which things of our system that it is there and ready to carry out its task when you need It to be done. This system accommodates growth in terms of user data volume, and transactions. Compatibility requirements for the Web-based platform to support operating systems and devices. Follow the Web-based platform functions correctly across different platforms.

- **Maintainability**

  Our system keeps the requirements related to the cease of maintaining and updating the Web-based platform. Our system keeps modularity, documentation, and support for a variety of control systems.

3.3 Introduction to System Design

Based on the user's condition and the detailed analysis of the new system must be designed. This is the chapter of system designing. It is the most crucial phase in the development of a system. The logical system design attains at as a result of system analysis is improved into physical system design.

3.3.1.1 Use Case Diagrams

A use case diagram at its simplest is a image of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is complex. A use case diagram is a dynamic or behavior diagram in UML. Use case

diagrams model the functionality of a system using actors and use cases. The use cases are represented by either circles or ellipses. Use cases are a set of actions, services, and functions that the system needs to perform. The ingredients for this type of diagram are use cases and actors:

Actor

Actor represents roles that user takes on when they use the IT system. An actor agrees a role played by a user or any other system that relates with the subject. The actor of a use case is the stakeholder that calls on the system to deliver one of its services. One actor can act in more than one role towards the IT system.

List of Actors

Admin

- Admin can **log in** any user account or Logout account any User.
- Admin can **manage user** it means add user or block and delete and View user id.
- Admin can view **dashboard** or monitor daily sales, weekly sales, monthly sales, no of stocks Available, maximum sales and minimum sales.
- Admin can **add categories** of medicine like tablets, syrup, injections, capsules and antibiotics etc.
- Admin can easily find the medicine **location details** in which rack medicine is available.
- Admin can add the all-**medicine company** details such as medicine name and medicine company.
- Admin can add the all-**medicine supplier** details such as supplier name, supplier address, supplier contact numbers, supplier email, status, date & time.
- Admin can **medicine** such as medicine name, medicine company, pack details, available Quantity, location rack, status, date & time added on, date & time updated on.

- Admin can manage **medicine purchase** such as medicine name, medicine batch no, medicine supplier, medicine quantity, available quantity, price per unit, total cost, mfg. date, medicine expiry date, medicine sales price, purchase date, status, action.
- Admin can manage **order bill** such as patient name, doctor name, order amount, order created by, status, date & time order added on, order updated on, print order slip, edit and delete.
- Admin can manage and view **customer orders** such as customer id, customer name, customer numbers, customer email, customer payment method, customer address, customer street, customer city, customer country, total products customer orders, total price customer order, customer order date & time.
- Admin can manage and view all **customer details** such as customer name, customer email address, customer number, customer address.
- Admin can get **email and alert notification** When medicine is out of stock and medicine Expired.

Use Case Diagram for Admin

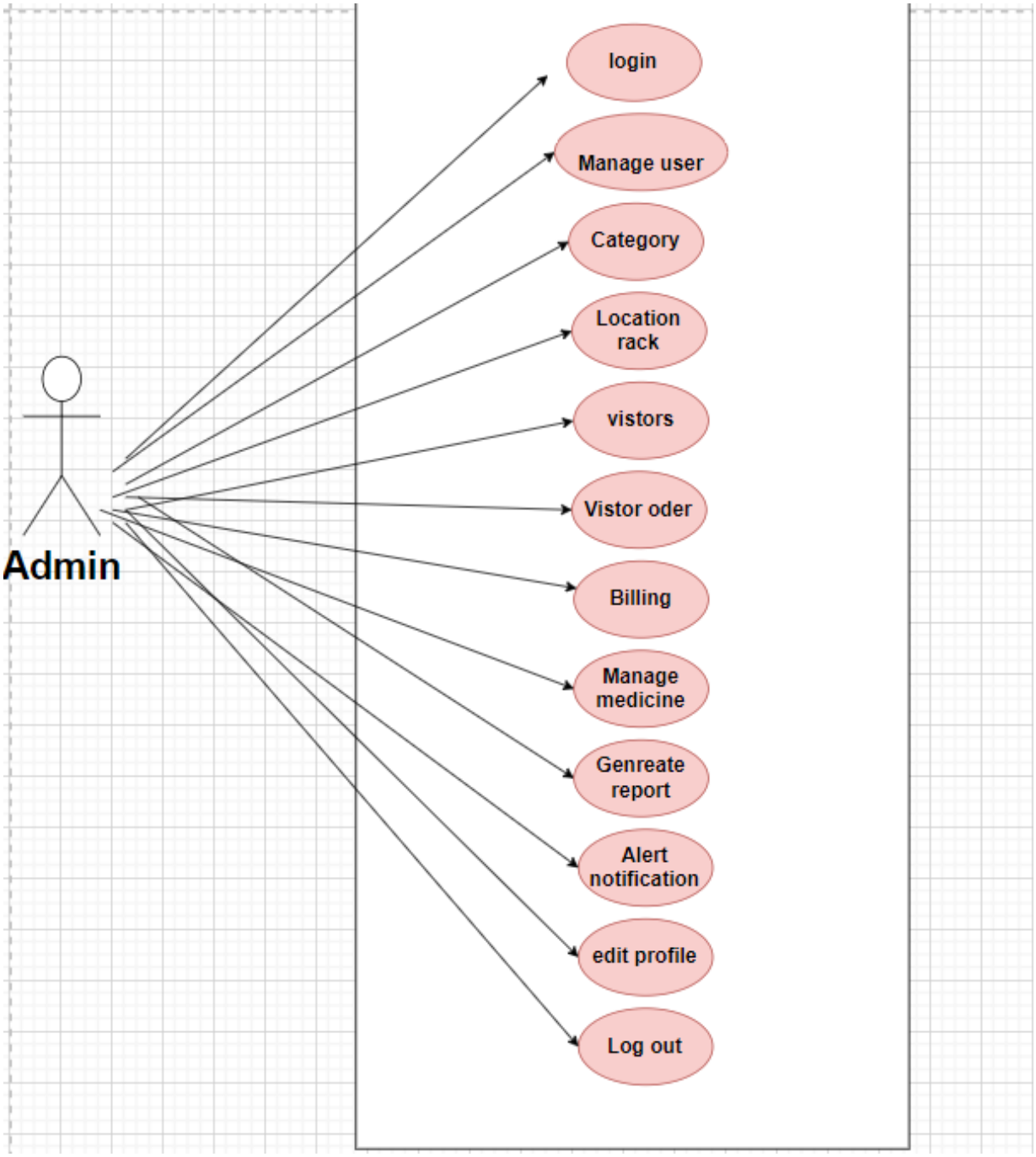


Figure 3.1 Admin Use Case Diagram

Sub Users

- Sub users can **login** and put data like Gmail and password.
- Sub users can manage **medicine purchase** such as medicine name, medicine batch no, medicine supplier, medicine quantity, available quantity, price per unit, total cost, mfg. date, medicine expiry date, medicine sales price, purchase date, status, action.
- Sub users can manage **order bill** such as patient name, doctor name, order amount, order created by, status, date & time order added on, order updated on, print order slip, edit and delete.
- Sub users can manage and view all **customer details** such as customer name, customer email address, customer number, customer address.
- Sub user can get **email and alert notification** When medicine is out of stock and medicine Expired.

Use Case Diagram for Sub User

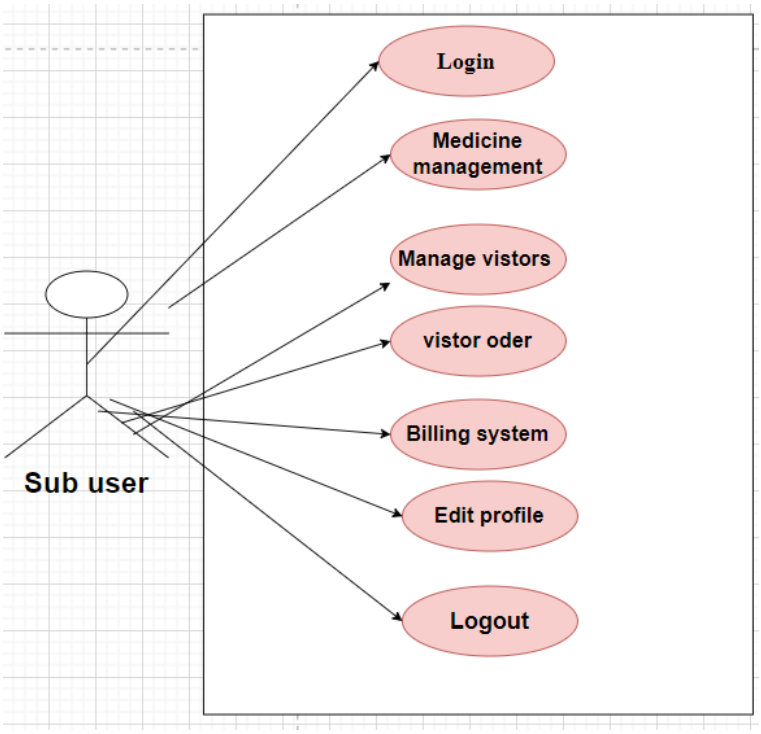


Figure 3.2 Sub User Use Case Diagram

Customers/Users

- Will be able to search about medicine specifications and make medicine purchase online. Customer can **register** and put all mandatory data like Name, Gmail, number, address and password etc.
- Customers after register can **log in** and put data like Gmail and password.
- Customers can **search medicine** different name like brand names.
- Customers can **view all medicine** such medicine name, medicine picture, medicine price, Medicine quantity available.
- Customers can **add to cart** the desire medicine.
- Customers can **proceed to check out** the order which they desire medicine.
- Customers can **complete the order details** such as Customers name, Customers Gmail Address, customers street, customers flat no, customers city, customers numbers, customers payment method, customers country.
- Customers can **get email order confirmation** after the complete the order.

Customer / User Use Case Diagram

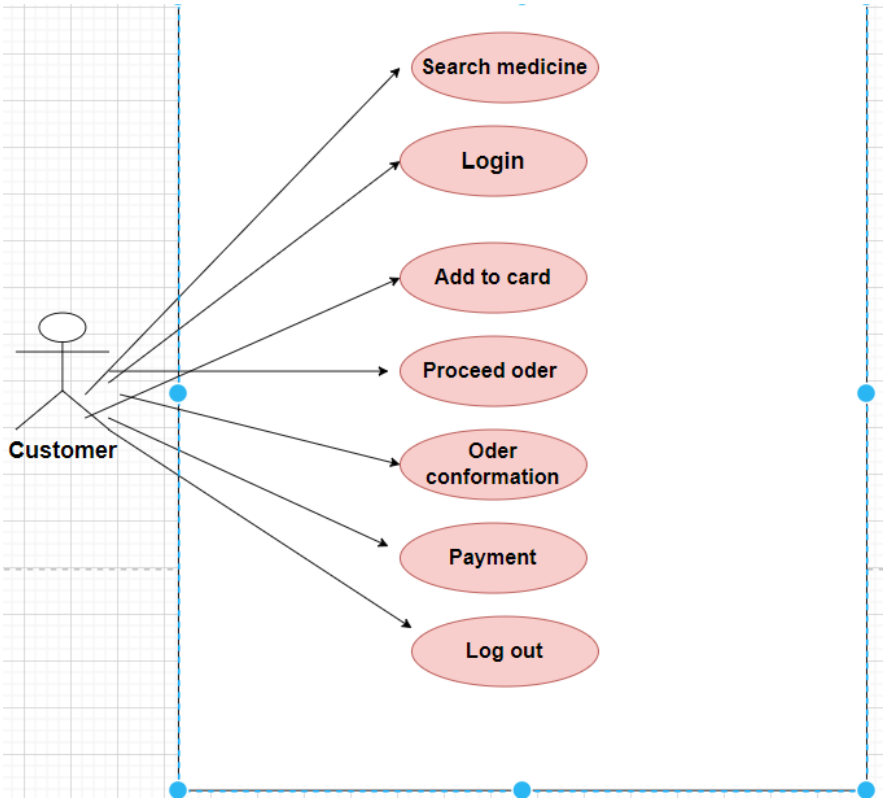


Figure 3.3 Customer / Users Use Case Diagram

3.3.1.2 Sequence Diagrams

A sequence diagram is defined as one of the use case realizations diagrams in the software engineering field used to show the way objects interact in a given scenario of a use case. It depicts how an object sends and receives messages to other objects in the course of performing the intended function.

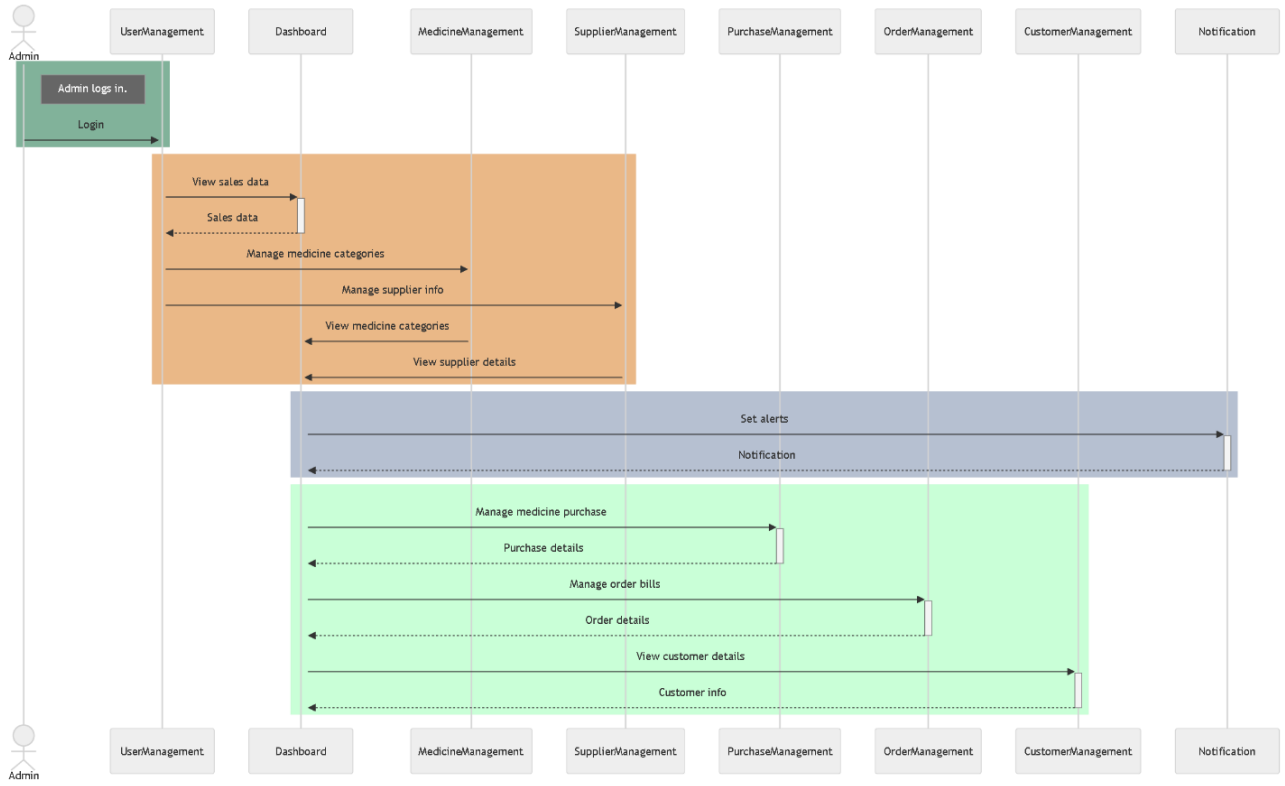


Figure 3.4 Sequence Diagram

3.3.1.3 Activity Diagram

Activity diagram is another important diagram in UML to describe the dynamic aspects of the system. Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. Activity diagram is sometimes considered as the flowchart. Although the diagrams look like a flowchart, they are not. It shows different flows such as parallel, branched, concurrent, and single.

- **Admin Data Flow Diagram**

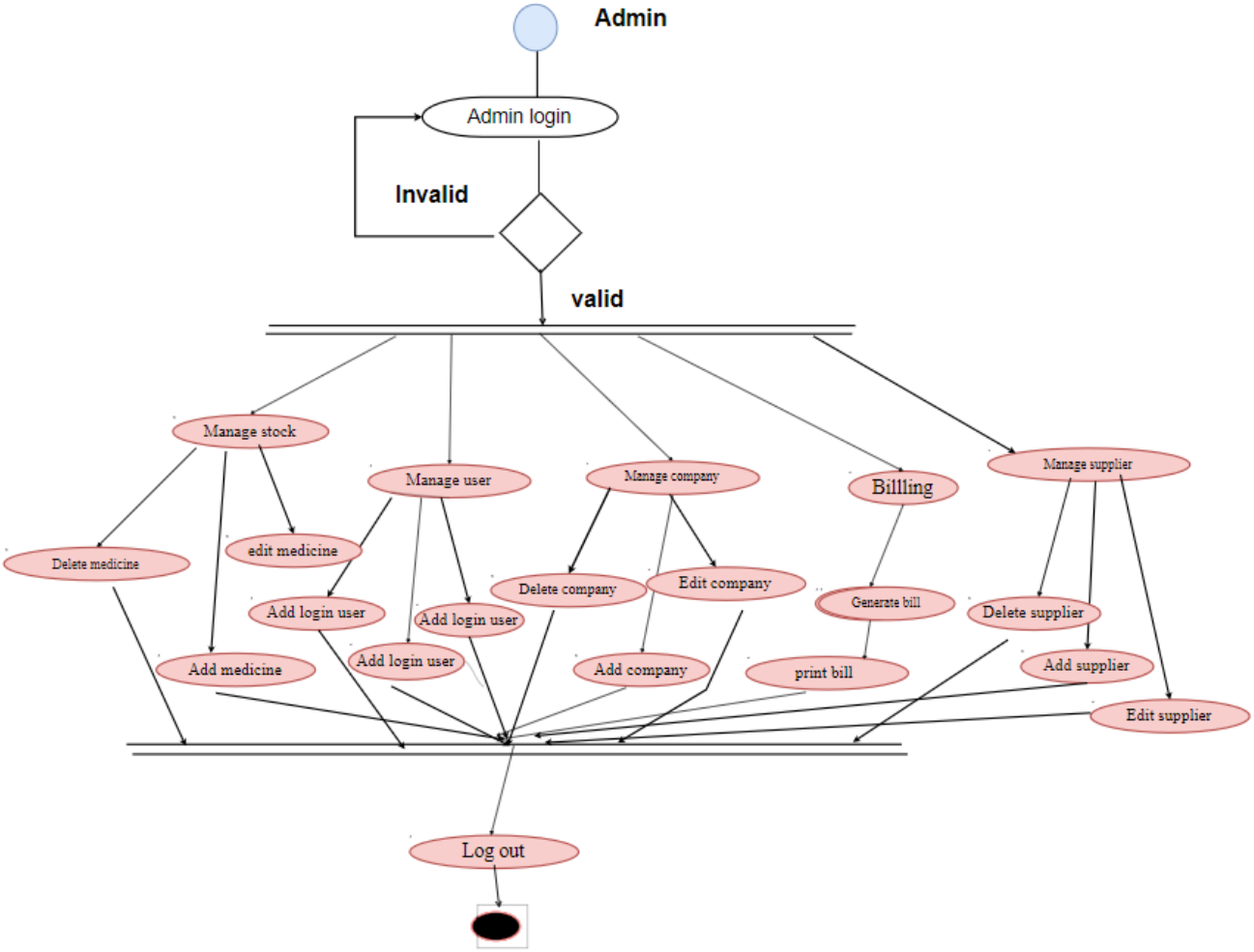


Figure 3.5 Activity Diagram 3.5

➢ Activity Diagram Sub User

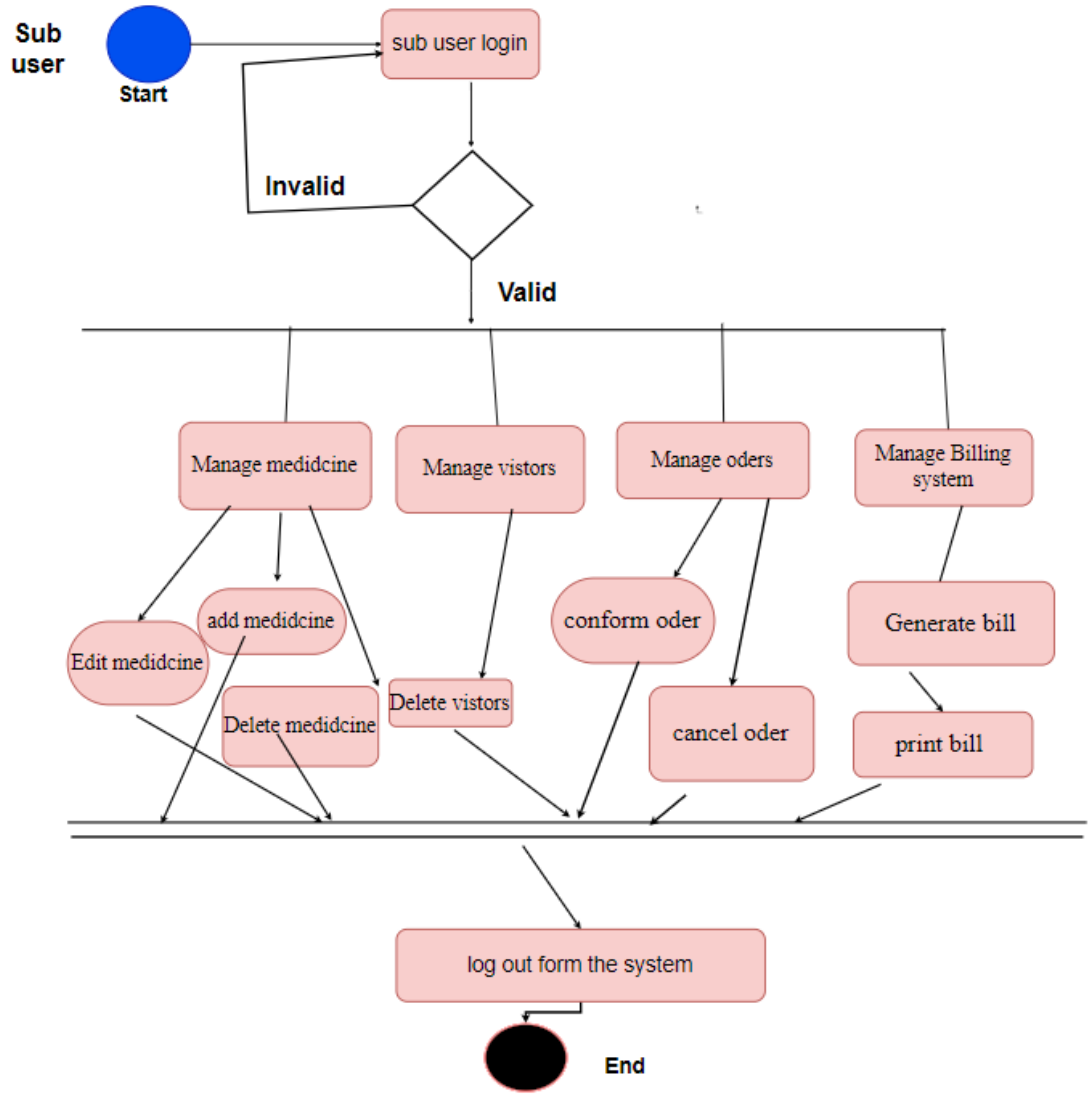


Figure 3.6 Sub User Activity Diagram

➢ Customer / User Activity Diagram

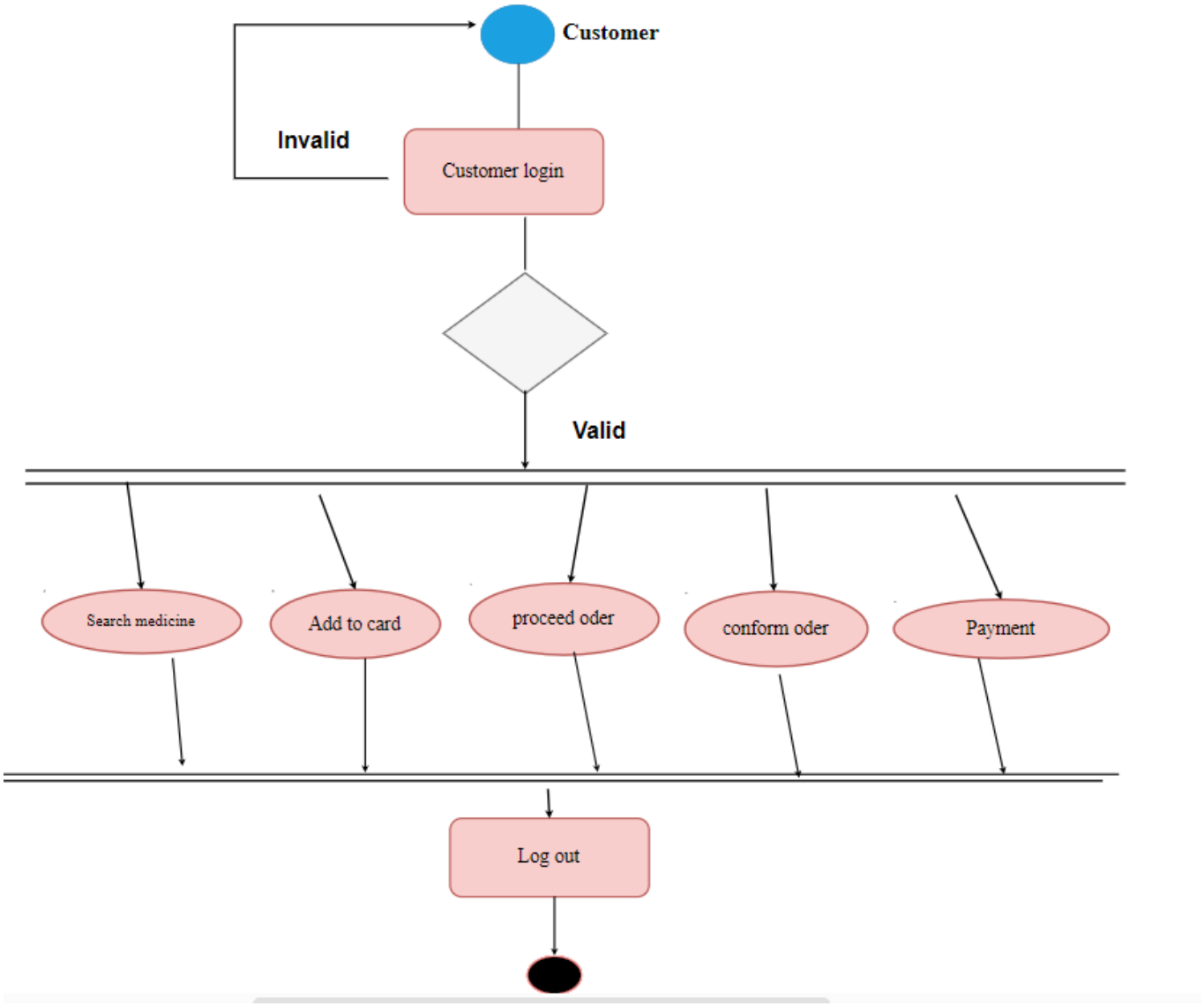


Figure 3.7 Customer / User Activity Diagram

3.3.1.4 System Architecture Diagram

A system architecture diagram is a graphical model of a system with the sub-systems and elements involved showing their connections. It also shows how elements within a system are able to interact in order to provide certain functionalities or meet certain goals/aims.

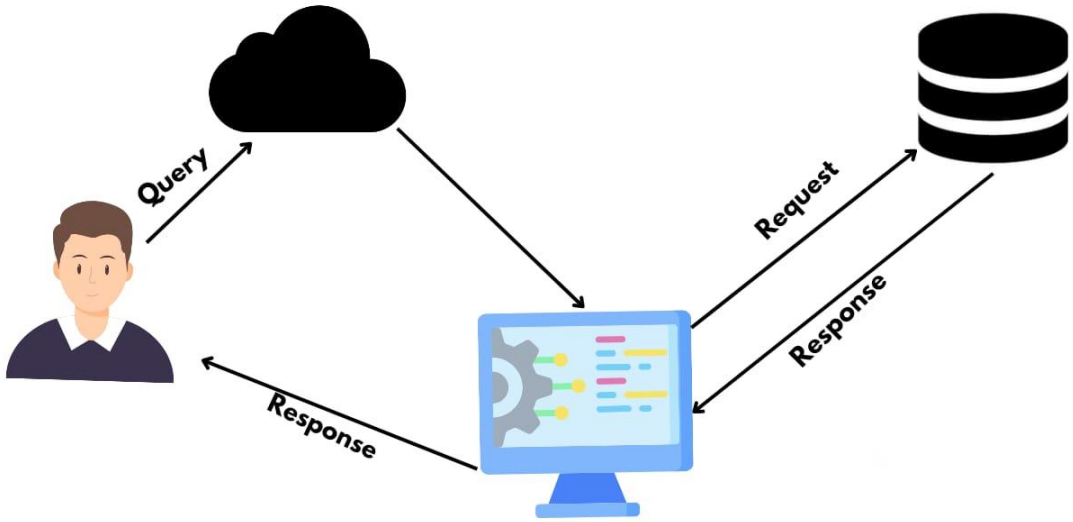


Figure 3.8 System Architecture Diagram

3.3.1.5 Entity Relationship Diagrams (ERDs)

An Entity Relationship (ER) Diagram is a type of flowchart that illustrates how "entities" such as people, objects or concepts relate to each other within a system. ER Diagrams are most often used to design or debug relational databases in the fields of software engineering, business information systems, education and research. NoSQL, unlike SQL which has ER and class diagrams, has neither names nor constraints for data modeling diagrams. The obvious reason is the relaxed rules of NoSQL about relationships, which aim to get a developer started with minimum requirements.

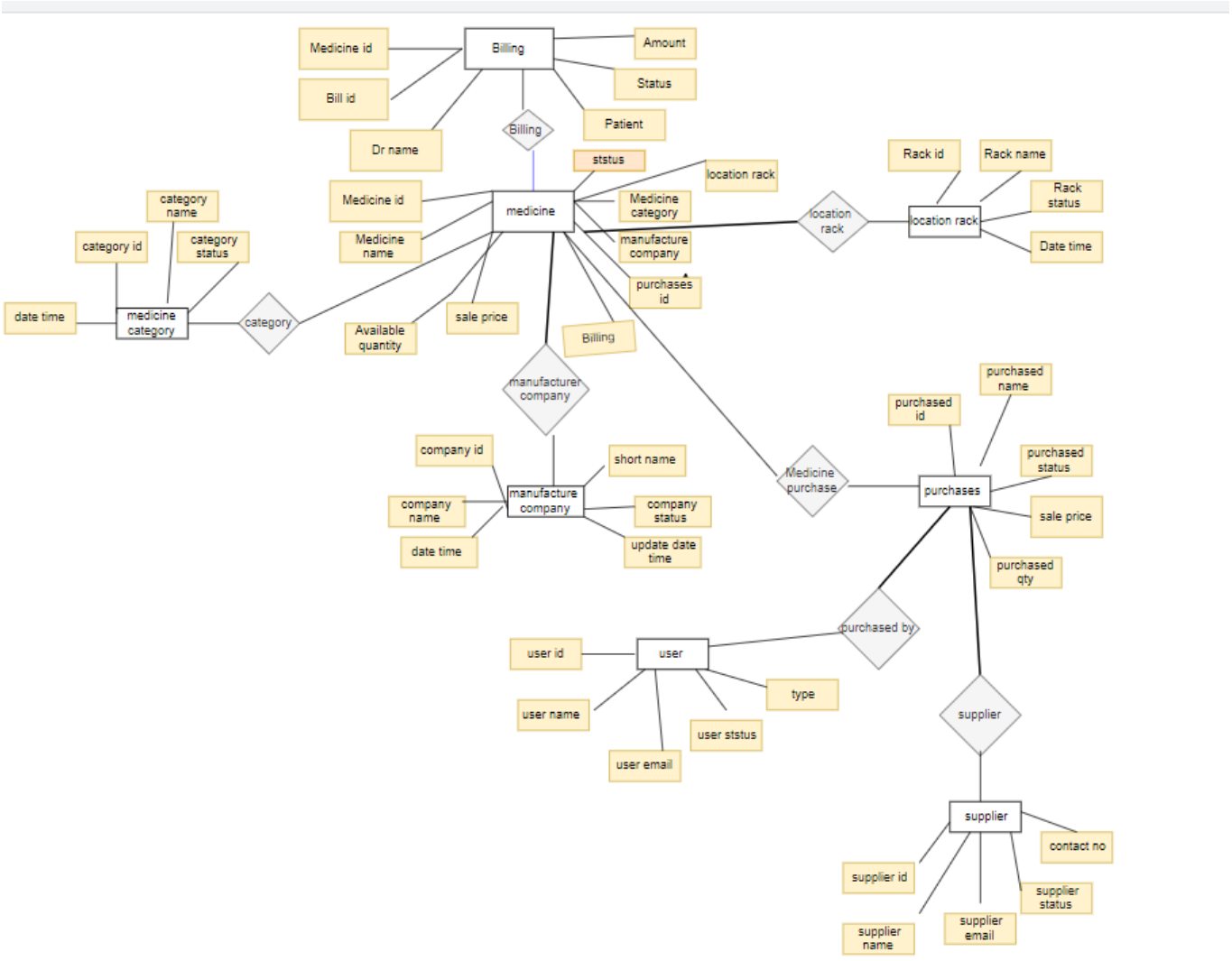


Figure 3.9 Entity Relationship Diagram

Chapter 4

SYSTEM DEVELOPMENT

&

IMPLEMENTATION

4.1. Development Methodology

We accepted the Agile methodology for the development of the **"Smart Safe"** web-based platform. Agile methodology permits for iterative development, where requirements and solutions develop through collaboration between cross-functional teams. This approach helped flexibility and adaptability during the development process, confirming that the final product met user needs and expectations.

4.1.2 System Architecture

The system architecture of the **"Smart Safe"** web-based platform is designed to be robust, scalable, and secure. In a 3-tier architecture the system is divided into three main layers.

- **Presentation Layer(front-end)**

- **Customers:**

  Search Interface: Enables customers to search for the various available medicines.

  User Account Management: They can open their accounts and they are able to manage their accounts on their own.

  Order Placement: Order placing by customers with regard to the medicines they want to purchase.

- **Sub-Users:**

  Restricted Dashboard: Sub-users can only partially control the admin features available, including settings for certain sections, such as inventory-management.

- **Admin:**

  Admin Dashboard: An admin panel to control user activity of the system, stock management, and billing and financial management systems.

- **Logic Layer (Back-end)**
    - Authentication and Authorization: It allows customers, sub-users, and even the admin to gain access securely.
    - Search Service: Deals with search queries from customers.
    - Order Management Service: Deals with orders by customers.
    - User Management Service: Only admin can perform sub-users and customer accounts management task of creating, modifying and deleting sub-users.
    - Inventory Management Service: Admin can control each medicine's location on the rack, as well as its category and supplier.
    - Billing Management Service: This shows that Admin can perform billing activities from within the system.

Data Access Layer (Database)

- User Data: This sub-module contains information about customers and sub-users that exist within the system.
- Inventory Data: It's used for warehousing medicines details like categories, racks, suppliers and many other.
- Order Data: Requires and maintains details of orders placed by customers.
- Billing Data It is used to manage the billing of customers and records the transaction made in the Swift Cash business.

4.2 Tools / Languages Selection

The development of the **"Smart Safe"** web-based platform involved the use of many tools and technologies, including:

- **Frontend Development**

  HTML, CSS, Boot Strap, Font Awesome, Material Design icons, Chart.js and JavaScript, For a dynamic user interface.

- **Backend Development**

  PHP for server-side logic, Express.js for routing, and Sequelize ORM for database management

  .

- **Database**

  MySQL for data storage.

- **Version Control**

  Git for source code management and collaboration.

- **Integrated Development Environment**

  Visual Studio Code for coding and debugging.

4.3 Client-Side Technology

It stands for the part and code of the building that exist within a user's web browser as opposed to the server-side technology that runs on a server. These technologies are for drawing the graphical user interface and also managing user interactions. These are HTML, CSS, and JavaScript, among others, and are essential in ensuring user-friendly and interactive websites.

- **HTML and CSS**

  HTML (Hyper Text Markup Language) is used to create World Wide Web documents or web pages. That is a tool that organizes the content on the web with the help of elements such as headings, paragraphs, lists, links, images, etc. HTML is made of tags that form the structure of a webpage, and the browser recognizes these tags.

- **CSS**

  CSS stands for Cascading Style Sheets, which is used to style HTML or other tags in HTML. Cascading style sheets describe how to format and display elements of an HTML document, such as its layout, color schemes, and font. CSS enables the developers to enhance the aesthetic appeal of the webpages, and it also enables the webpages to adapt to the browser and devices' layouts in the best way possible.

- **Vanilla JavaScript**

  Scripting means plain vanilla JavaScript and does not employ the help of any libraries or frameworks. This is Java script, but in its purest form, with no pre-processing or modifications. Web developers have been known to use plain JavaScript to change content on the web page, listen to events, and make asynchronous HTTP requests, among other state events that apply interactivity to the page.

- **Bootstrap**

  Bootstrap is one of the most frequently used CSS frameworks that was initially created by Twitter. It offers predefined elements and a flexible grid that also makes it easier to create mobile-first sites. Bootstrap contains typography, forms, buttons, navigation, and other interface elements that let the developer define the look and feel that are identical on all devices.

4.4 Server-Side Technology

Server-side technology can therefore be defined as processes and operations that occur on a web server in relation to requests made by clients. It includes database administration, running scripts, and serving content to the user's browser. As opposed to client-side technology, which can be executed strictly within the client's browser, server-side technology works on the server and manages applications' backend.

- **PHP**

  PHP is programming language used for web development, and it is a server-side scripting language. It interprets scripts on the server to produce dynamic content, handle form data, work with databases, and handle other behind-the-scenes operations. PHP scripts are executed on the server side and whatever is produced as output in the form of HTML is sent to the client.

- **MySQL**

  MySQL is a form of RDBMS used to manage data, that is, store, retrieve, update and manipulate information. It runs on the server-side and stores data in a tabular form with rows and columns. MySQL uses SQL, which stands for structured query language, for defining interrogative and manipulative instructions. Sometimes it combines with server-side programming languages such as PHP to develop interactive and data-intensive web -based platforms.

4.5 Code of important Modules

Main Dashboard Admin index.php

index.php 1

C: > xampp > htdocs > apexmedicare-main > index.php > ...

```php
<?php include 'header.php'; ?>
</div>

</div>
<!-- Left Sidebar End -->
<?php

require_once('class/db.php');

$object = new db();

if (!$object->is_login()) {
  header('location:login.php');
}

if (!$object->is_master_user()) {
  header('location:medicine_purchase.php');
}

require_once('header.php');

$object->query = "
    SELECT * FROM medicine_msbs
    INNER JOIN category_msbs
    ON category_msbs.category_id = medicine_msbs.medicine_category
    INNER JOIN  medicine_manufacuter_company_msbs
    ON  medicine_manufacuter_company_msbs.medicine_manufacuter_company_id = medicine_msbs.medicine_manufactured_by
    INNER JOIN location_rack_msbs
    ON location_rack_msbs.location_rack_id = medicine_msbs.medicine_location_rack
    WHERE medicine_status = 'Enable'
    AND medicine_available_quantity < medicine_pack_qty
    ORDER BY medicine_msbs.medicine_name ASC
";
```

index.php 1

C: > xampp > htdocs > apexmedicare-main > index.php > div#miniaresult.main-content > div.page-content > div.container-fluid > div.row > div.col-xl-3.col-md-6 > div.card.bg-wa

```
";

$medicine_result = $object->get_result();

?>

<div class="main-content" id="miniaresult">

  <div class="page-content">
    <div class="container-fluid">
      <h1 class="mt-4">Dashboard</h1>
      <ol class="breadcrumb mb-4">
        <li class="breadcrumb-item active">Admin Panel</li>
      </ol>
      <div class="row">

        <div class="col-xl-3 col-md-6">
          <div class="card bg-primary text-white mb-4">
            <div class="card-body">
              <h2 class="text-center" style="color: white; font-weight: bold;"><?php echo $object->Get_total_no_of_medicine(); ?></h2>
              <h5 class="text-center" style="color: white; font-weight: bold;">In Stock Medicine</h5>
            </div>
          </div>


        </div>
        <div class="col-xl-3 col-md-6">
          <div class="card bg-warning text-white mb-4">
            <div class="card-body">
              <h2 class="text-center" style="color: white; font-weight: bold;"><?php echo $object->Count_outstock_medicine(); ?></h2>
              <h5 class="text-center" style="color: white; font-weight: bold;">Out of Stock Medicine</h5>
            </div>
```

```
            </div>

        </div>
        <div class="col-xl-3 col-md-6">
          <div class="card bg-danger text-white mb-4">
            <div class="card-body">
              <h2 class="text-center" style="color: white; font-weight: bold;"><?php echo $object->cur_sym . number_format(floatval($object
              <h5 class="text-center" style="color: white; font-weight: bold;">Total Purchase</h5>
            </div>
          </div>

        </div>
        <div class="col-xl-3 col-md-6">
          <div class="card bg-success text-white mb-4">
            <div class="card-body">
              <h2 class="text-center" style="color: white; font-weight: bold;"><?php echo $object->cur_sym . number_format(floatval($object
              <h5 class="text-center" style="color: white; font-weight: bold;">Total Sale</h5>
            </div>
          </div>

        </div>

      </div>

      <div class="row">
        <!-- First column: Daily, Yearly, Weekly Sale Status -->
        <div class="col-xl-6">
```

```
<div class="col-xl-6">
  <!-- Daily Sale Chart -->
  <div class="card mb-4">
    <div class="card-header">
      <i class="fas fa-chart-area me-1"></i>
      Sale Status Daily
    </div>
    <div class="card-body">
      <canvas id="saleChartdaily" width="100%" height="40"></canvas>
    </div>
  </div>
  <!-- Yearly Sale Chart -->
  <div class="card mb-4">
    <div class="card-header">
      <i class="fas fa-chart-area me-1"></i>
      Sale Status Weekly
    </div>
    <div class="card-body">
      <canvas id="saleChartweekly" width="100%" height="40"></canvas>
    </div>
  </div>
  <!-- Weekly Sale Chart -->
  <div class="card mb-4">
    <div class="card-header">
      <i class="fas fa-chart-area me-1"></i>
      Sale Status Monthly
    </div>
    <div class="card-body">
```

```
      <canvas id="saleChart" width="100%" height="40"></canvas>
    </div>
  </div>
</div>
<!-- Second column: No. of Medicine Charts -->
<div class="col-xl-6">
  <!-- No. of Medicine 1 -->
  <!-- <div class="card mb-4">
    <div class="card-header">
      <i class="fas fa-chart-bar me-1"></i>
      No. of Medicine
    </div>
    <div class="card-body"><canvas id="stockChart" width="100%" height="40"></canvas></div>
  </div> -->
  <!-- Maximum Sold -->
  <div class="card mb-4">
    <div class="card-header">
      <i class="fas fa-chart-bar me-1"></i>
      Maximum Sold
    </div>
    <div class="card-body"><canvas id="stockChart2" width="100%" height="40"></canvas></div>
  </div>
  <!-- No. of Medicine 3 -->
  <div class="card mb-4">
    <div class="card-header">
      <i class="fas fa-chart-bar me-1"></i>
      Minimum Sold
```

```
<div class="main-content" id="miniaresult">
  <div class="page-content">
    <div class="container-fluid">
      <div class="row">
        <div class="col-xl-6">
          </div>
          <div class="card-body"><canvas id="stockChart9" width="100%" height="40"></canvas></div>
        </div>
      </div>
    </div>


    <div class="card mb-4">
      <div class="card-header">
        <i class="fas fa-table me-1"></i>
        List of Out of Stock Medicine
      </div>
      <div class="card-body">
        <table id="datatablesSimple">
          <thead>
            <tr>
              <th>Medicine Name</th>
              <th>Company</th>
              <th>Pack Detail</th>
              <th>Available Quantity</th>
              <th>Location Rack</th>
              <th>Status</th>
              <th>Added On</th>
              <th>Updated On</th>
              <th>Action</th>
            </tr>
          </thead>
          <tfoot>
            <tr>
              <th>Medicine Name</th>
              <th>Company</th>
              <th>Pack Detail</th>
              <th>Available Quantity</th>
              <th>Location Rack</th>
              <th>Status</th>
              <th>Added On</th>
              <th>Updated On</th>
              <th>Action</th>
            </tr>
          </tfoot>
          <tbody>
            <?php

            foreach ($medicine_result as $row) {
              $medicine_status = '';
              if ($row["medicine_status"] == 'Enable') {
                $medicine_status = '<div class="badge bg-success">Enable</div>';
              } else {
                $medicine_status = '<div class="badge bg-danger">Disable</div>';
              }
              echo '
                                <tr>
                                  <td>' . $row["medicine_name"] . '</td>
                                  <td>' . $row["company_name"] . '</td>
```

```
                                    <td>' . $row["medicine_pack_qty"] . ' ' . $row["category_name"] . '</td>
                                    <td>' . $row["medicine_available_quantity"] . '</td>
                                    <td>' . $row["location_rack_name"] . '</td>
                                    <td>' . $medicine_status . '</td>
                                    <td>' . $row["medicine_add_datetime"] . '</td>
                                    <td>' . $row["medicine_update_datetime"] . '</td>
                                    <td>
                                        <a href="medicine_purchase.php?action=add&code=' . $object->convert_data("add") .
                                    </td>
                                </tr>
                                ';
                            }

                            ?>
                        </tbody>
                    </table>
                </div>

            </div>
        </div>

    </div>
    <?php

    include('footer.php');

    ?>
```

```
    ?>
    <?php
    /*daily*/
    $area_chart_data = $object->Get_last_thirty_day_date();

    $sale_data = $object->Get_last_thirty_day_medicine_sale_data();

    /*monthly*/
    $area_chart_data_monthly = $object->Get_last_twelve_month_date();

    $sale_data_monthly = $object->Get_last_twelve_month_medicine_sale_data();

    /*weekly*/
    $area_chart_data_weekly = $object->Get_last_twelve_week_date();

    $sale_data_weekly = $object->Get_last_twelve_week_medicine_sale_data();
    /*weekly*/

    /*No. of Medicine*/
    $month_data = $object->Get_last_six_month_name();

    $stock_data = $object->Get_last_six_month_medicine_stock_data();
    /*No. of Medicine*/
    // Get the top 5 sold medicines
    $max_sold = $object->Get_max_sold_medicine();
    /* minimum sold*/
    $min_sold = $object->Get_min_sold_medicine();
    /*// Debug: check the content of $max_sold
var_dump($max_sold);*/
```

```
?>
<script>
  // Set new default font family and font color to mimic Bootstrap's default styling
  Chart.defaults.global.defaultFontFamily = '-apple-system,system-ui,BlinkMacSystemFont,"Segoe UI
  Chart.defaults.global.defaultFontColor = '#292b2c';

  // Area Chart Example
  var ctx = document.getElementById("saleChart");
  var myLineChart = new Chart(ctx, {
    type: 'line',
    data: {
      labels: <?php echo json_encode(array_reverse($area_chart_data_monthly['month_date'])); ?>,
      datasets: [{
        label: "Medicine Sale",
        lineTension: 0.3,
        backgroundColor: "rgb(51, 204, 51,0.50)",
        borderColor: "rgba(2,117,216,0.40)",
        pointRadius: 5,
        pointBackgroundColor: "rgb(13, 110, 253)",
        pointBorderColor: "rgb(255, 193, 7)",
        pointHoverRadius: 5,
        pointHoverBackgroundColor: "rgb(25, 135, 84)",
        pointHitRadius: 50,
        pointBorderWidth: 2,
        data: <?php echo json_encode(array_reverse($sale_data_monthly)); ?>,
      }],
    },
    options: {
      scales: {
        xAxes: [{
          time: {
```

```
    options: {
      scales: {
        xAxes: [{
          time: {
            unit: 'date'
          },
          gridLines: {
            display: false
          },
          ticks: {
            maxTicksLimit: 7
          }
        }],
        yAxes: [{
          ticks: {
            min: 0,
            maxTicksLimit: 5
          },
          gridLines: {
            color: "rgba(0, 0, 0, .125)",
          }
        }],
      },
      legend: {
        display: false
      }
    }
  });
```

```
// Get the week dates and sales data from PHP
var weekData = <?php echo json_encode(array_reverse($area_chart_data_weekly['week_date'])); ?>;
var salesData = <?php echo json_encode(array_reverse($sale_data_weekly)); ?>;

// Combine the week names and sales data into a single array
var combinedData = weekData.map(function(week, index) {
  return week + ' (' + salesData[index] + ')';
});

// Area Chart Example
var ctx3 = document.getElementById("saleChartweekly");
var myLineChart = new Chart(ctx3, {
  type: 'line',
  data: {
    labels: combinedData, // Use the combined data array
    datasets: [{
      label: "Medicine Sale",
      lineTension: 0.3,
      backgroundColor: "rgb(51, 204, 51,0.50)",
      borderColor: "rgba(2,117,216,0.40)",
      pointRadius: 5,
      pointBackgroundColor: "rgb(13, 110, 253)",
      pointBorderColor: "rgb(255, 193, 7)",
      pointHoverRadius: 5,
      pointHoverBackgroundColor: "rgb(25, 135, 84)",
      pointHitRadius: 50,
      pointBorderWidth: 2,
      data: salesData, // Use the sales data
    }],
  },
  options: {
    scales: {
      xAxes: [{
        gridLines: {
          display: false
        },
        ticks: {
          maxTicksLimit: 7
        }
      }],
      yAxes: [{
        ticks: {
          min: 0,
          maxTicksLimit: 5
        },
        gridLines: {
          color: "rgba(0, 0, 0, .125)",
        }
      }],
    },
    legend: {
      display: false
    }
  }
});
```

```
// Area Chart Example
var ctx4 = document.getElementById("saleChartdaily");
var myLineChart = new Chart(ctx4, {
  type: 'line',
  data: {
    labels: <?php echo json_encode(array_reverse($area_chart_data['month_date'])); ?>,
    datasets: [{
      label: "Medicine Sale",
      lineTension: 0.3,
      backgroundColor: "rgb(51, 204, 51,0.50)",
      borderColor: "rgba(2,117,216,0.40)",
      pointRadius: 5,
      pointBackgroundColor: "rgb(13, 110, 253)",
      pointBorderColor: "rgb(255, 193, 7)",
      pointHoverRadius: 4,
      pointHoverBackgroundColor: "rgb(25, 135, 84)",
      pointHitRadius: 50,
      pointBorderWidth: 2,
      data: <?php echo json_encode(array_reverse($sale_data)); ?>,
    }],
  },
  options: {
    scales: {
      xAxes: [{
        time: {
          unit: 'date'
        },
        gridLines: {
          display: false
        }
        display: false
      },
      ticks: {
        maxTicksLimit: 7
      }
    }],
    yAxes: [{
      ticks: {
        min: 0,
        maxTicksLimit: 5
      },
      gridLines: {
        color: "rgba(0, 0, 0, .125)",
      }
    }],
  },
  legend: {
    display: false
  }
}
});

var ctx1 = document.getElementById("stockChart");
var myLineChart1 = new Chart(ctx1, {
  type: 'bar',
  data: {
    labels: <?php echo json_encode(array_reverse($month_data['month_name'])); ?>,
```

```
424             datasets: [{
425               label: "No of Medicine",
426               backgroundColor: "rgba(2,117,216,1)",
427               borderColor: "rgba(2,117,216,1)",
428               data: <?php echo json_encode($stock_data); ?>,
429             }],
430           },
431           options: {
432             scales: {
433               xAxes: [{
434                 gridLines: {
435                   display: false
436                 },
437                 ticks: {
438                   maxTicksLimit: 6
439                 }
440               }],
441               yAxes: [{
442                 ticks: {
443                   maxTicksLimit: 5
444                 },
445                 gridLines: {
446                   display: true
447                 }
448               }],
449             },
450             legend: {
451               display: false
```

```
    }
  }
});

/*var ctx9 = document.getElementById("stockChart2");
var myLineChart9 = new Chart(ctx9, {
  type: 'bar',
  data: {
    labels: ["<?php echo $max_sold['medicine_name']; ?>"],
    datasets: [{
      label: "<?php echo $max_sold['medicine_name']; ?>",
      backgroundColor: "rgba(2,117,216,1)",
      borderColor: "rgba(2,117,216,1)",
      data: [<?php echo $max_sold['max_sold']; ?>],
    }],
  },
  options: {
    scales: {
      xAxes: [{
        gridLines: {
          display: false
        },
        ticks: {
          maxTicksLimit: 6
        }
      }],
      yAxes: [{
        ticks: {
```

```
          echo $medicine['max_sold'] . ',';
        } ?>
      ],
    }],
  },
  options: {
    scales: {
      xAxes: [{
        gridLines: {
          display: false
        },
        ticks: {
          maxTicksLimit: 6
        }
      }],
      yAxes: [{
        ticks: {
          maxTicksLimit: 5
        },
        gridLines: {
          display: true
        }
      }],
    },
    legend: {
      display: false
```

```
536         }
537       }
538     });
539   </script>
540   <script>
541     var ctx10 = document.getElementById("stockChart9");
542     var myLineChart9 = new Chart(ctx10, {
543       type: 'bar',
544       data: {
545         labels: [
546           <?php foreach ($min_sold as $medicine) {
547             echo '"' . $medicine['medicine_name'] . '",';
548           } ?>
549         ],
550         datasets: [{
551           label: "Sell Medicine",
552           backgroundColor: "rgba(255, 99, 132, 0.5)", // Red color with 50% opacity
553           borderColor: "rgba(255, 99, 132, 1)", // Red color
554           data: [
555             <?php foreach ($min_sold as $medicine) {
556               echo $medicine['min_sold'] . ',';
557             } ?>
558           ],
559         }],
560       },
561       options: {
562         scales: {
```

```
560       },
561       options: {
562         scales: {
563           xAxes: [{
564             gridLines: {
565               display: false
566             },
567             ticks: {
568               maxTicksLimit: 6
569             }
570           }],
571           yAxes: [{
572             ticks: {
573               maxTicksLimit: 5
574             },
575             gridLines: {
576               display: true
577             }
578           }],
579         },
580         legend: {
581           display: false
582         }
583       }
584     });
585   </script>
586
587   <?php include 'footer.php'; ?>
```

```
480                maxTicksLimit: 5
481              },
482              gridLines: {
483                display: true
484              }
485            }],
486          },
487          legend: {
488            display: false
489          }
490        }
491      });*/
492    </script>
493    <script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
494    <script>
495      var ctx9 = document.getElementById("stockChart2");
496      var myLineChart9 = new Chart(ctx9, {
497        type: 'bar',
498        data: {
499          labels: [
500            <?php foreach ($max_sold as $medicine) {
501              echo '"' . $medicine['medicine_name'] . '",';
502            } ?>
503          ],
504          datasets: [{
505            label: "Sell Medicine",
506            backgroundColor: "rgba(75, 192, 192, 1)", // Green color
507            borderColor: "rgba(75, 192, 192, 1)", // Green color
508            data: [
509              <?php foreach ($max_sold as $medicine) {
```

Summary

In this chapter, different features of system development and implementation strategies, system design, and the choice of tools and programming languages were discussed. We looked at client-side and server-side technologies as well as the importance of each in the context of web development. Further, we explained the need to code key modules in a system in order to provide the core functionality and quality of the system.

Chapter 5

TESTING

5.1 Testing

Testing was shown in multiple stages to ensure the system's reliability and performance:

- **Unit Testing**

  Each module was tested autonomously to verify that it performs as expected.

- **Integration Testing**

  Modules were integrated and tested together to confirm they work seamlessly.

- **System Testing**

  The complete system was tested in an environment similar to the production environment to identify any issues.

- **User Acceptance Testing (UAT)**

  The web-based platform was tested by end-users to ensure it meets their requirements and expectations.

5.2 Test Plan

The software test plan is designed to prescribe the scope, approach, resources and schedule of testing activities. The plan identifies items to be tested, features to be tested, the types of testing to be performed and schedule required to complete testing.

5.2.1 Process of test plan

• Identify the requirements to be tested.
• Identify the expected results for each test.
• Perform the test.
• Note down the test data, test cases used during the testing process.

5.2.2 Test Plan Admin

Sr	Test Area	Test Cases	Status
1	Login	Login 01 Login 02 Login 03 Login 04	Pass Pass Pass Pass
2	Forget Password	Forget Password 01 Forget Password 02	Pass Pass
3	Add User	User 01 User 02	Pass Pass
4	Add Category	Category 01 Category 02	Pass Pass
5	Add Medicine Location Details	Location 01 Location 02	Pass Pass
6	Add Medicine Company	Medicine Company 01 Medicine Company 02	Pass Pass
7	Add Medicine Supplier	Medicine Supplier 01 Medicine Supplier 02	Pass Pass
8	Add Medicine	Medicine 01 Medicine 02	Pass Pass
9	Add Medicine Purchase	Medicine Purchase 01 Medicine Purchase 02	Pass Pass

10	Order	Order 01 Order 02	Pass Pass
11	Customer's Orders	Customer's Orders 01 Customer's Orders 02	Pass Pass
12	Customers	Customers 01 Customers 02	Pass Pass
13	Email / Notification	Email / Notification 01 Email / Notification 02	Pass Pass

Test Plan Table 1

5.2.3 Sub User Test Plan

Sr	Test Area	Test Cases	Status
1	Medicine Purchase	Medicine Purchase 01 Medicine Purchase 02	Pass Pass
2	Oder	Order 01 Order 02	Pass Pass
3	Customer's Orders	Customer's Order 01 Customer's Orders 02	Pass Pass

4	Customer's	Customers 01 Customers 02	Pass Pass
5	Email & Notifications	Email Notifications 01 Email & Notifications 02	Pass Pass

Test Plan Table 2

5.2.4 Customers / User Test Plan

Sr	Test Area	Test Cases	Status
1	Register	Register 01 Register 02 Register 03	Pass Pass Pass
2	Login	Login 01 Login 02 Login 03	Pass Pass Pass
3	Search Medicine	Search Medicine 01 Search Medicine 02	Pass Pass
4	Add to Cart Medicine	Add to Cart Medine 01 Add to Cart Medicine 02	Pass Pass

5	Add and Remove Medicine Quantity	Add and Remove Medicine Quantity 01 Add and Remove Medicine Quantity 02	Pass Pass
6	Delete Medicine Cart	Delete Medicine Cart 01 Delete Medicine Cart 02	Pass Pass
7	Proceed to Check out	Proceed to Check out 01 Proceed to Check out 02	Pass Pass
8	Order	Order 01 Order 02	Pass Pass
9	Email & Notification Order Confirm	Email & Notification Order Confirm 01 Email & Notification Order Confirm 02	Pass Pass

Test Plan Table 3

Summary

Chapter 5 covers the testing procedures used to ascertain the validity of the system and its performance by conducting unit testing on single modules, integration testing of multiple coupled modules, system testing under simulated and real conditions, and finally user acceptance testing (UAT). It shows how various aspects of testing will be carried out: what is to be tested, what should be expected, the resources required, and timescales. The chapter also contains test plan tables on different aspects, for instance, login, password recovery, user management, and orders, and all the test cases passed as indicated.

Chapter 6

USER GUIDE

Introduction

Smart safe web-platform has a user-friendly interface and most of the controls are self-explanatory. This chapter intends to provide a comprehensive guide to use the Web-based platform So, it clears any ambiguity if it exists. This chapter helps to understand functionality of Web-based platform. This chapter contains screen shots which taken from this Smart safe Web-based platform.

6.1 Admin Activity

- **Admin Login**

  Admin can login to the Web-based platform by giving the required information e.g. Email and password

  .

Username or Email: admin@gmail.com

Password: admin

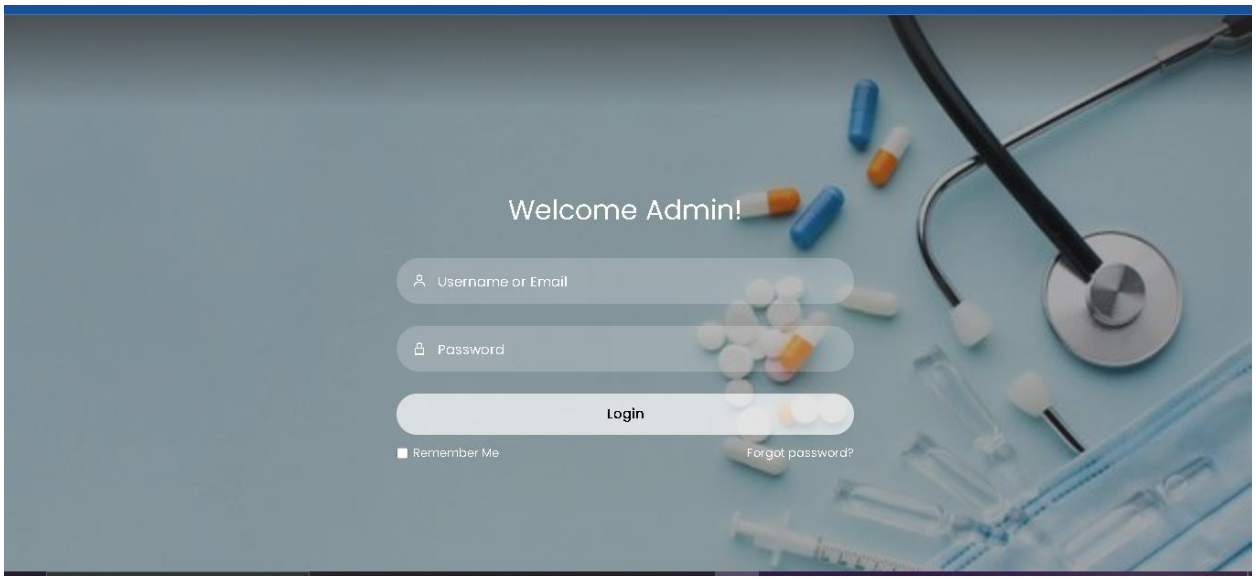


Figure 6.1 Admin Login Page

- **Admin Dashboard**

  Admin can view daily sales, weekly sales, monthly sales, maximum sales, minimum sales Admin can per perform function of manage user list, medicine categories, medicine location Details, medicine company, medicine supplier, medicine details, medicine purchase, order, customer orders, customers details, alert notification and email.

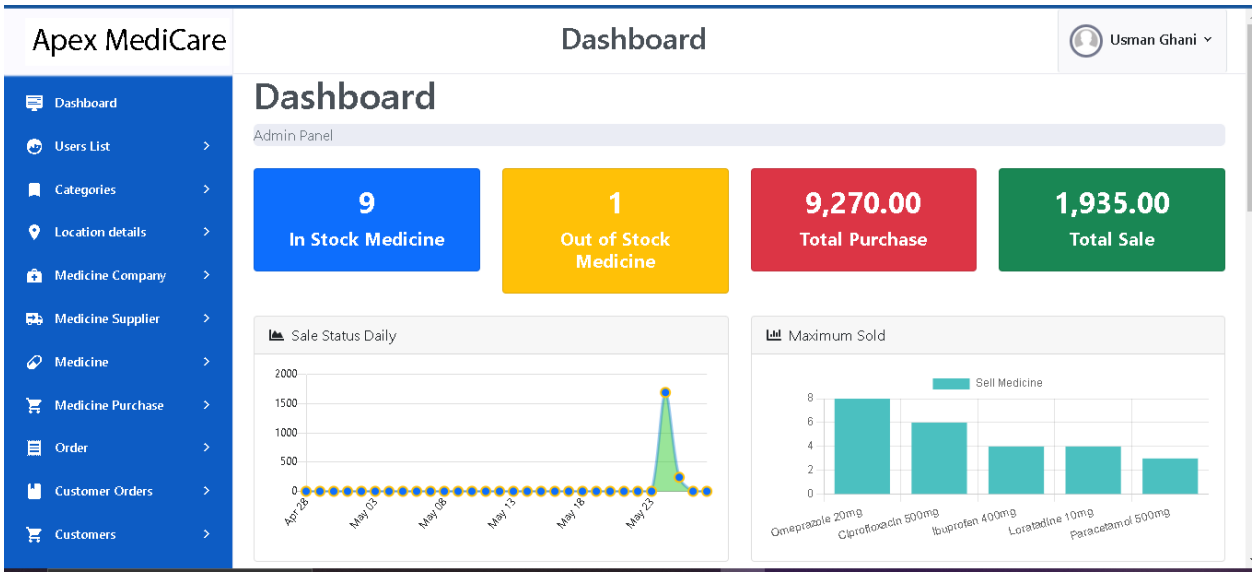


Figure 6.2 Admin Dashboard

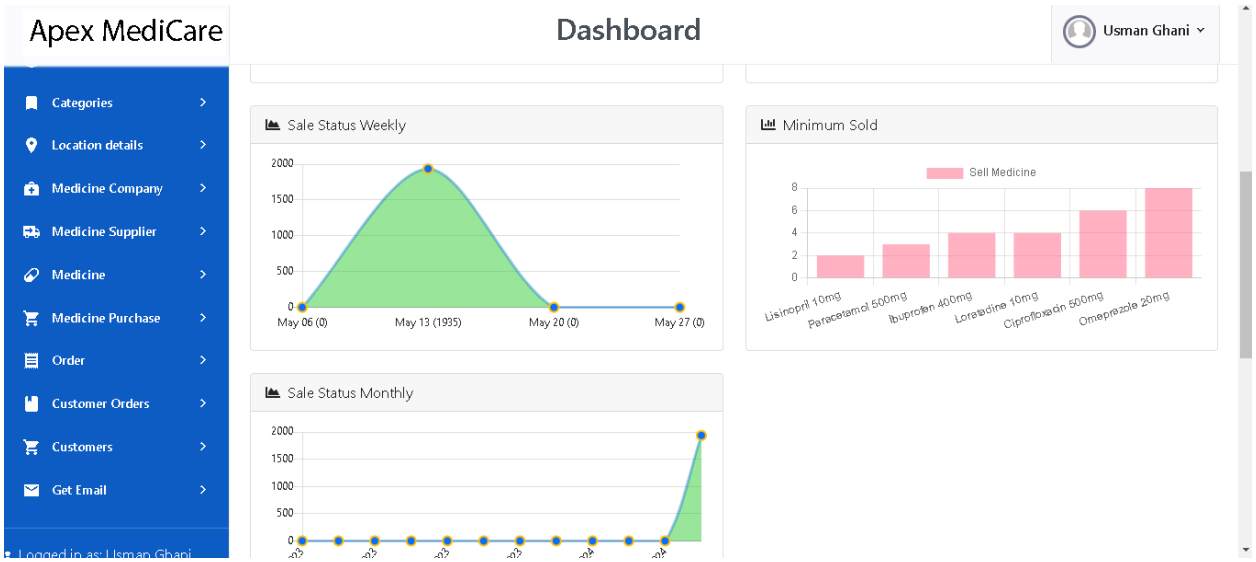


Figure 6.3 Admin Sales Dashboard

➢ Admin Manage Users List

Admin can manage user it means add user or block and delete and View user id.

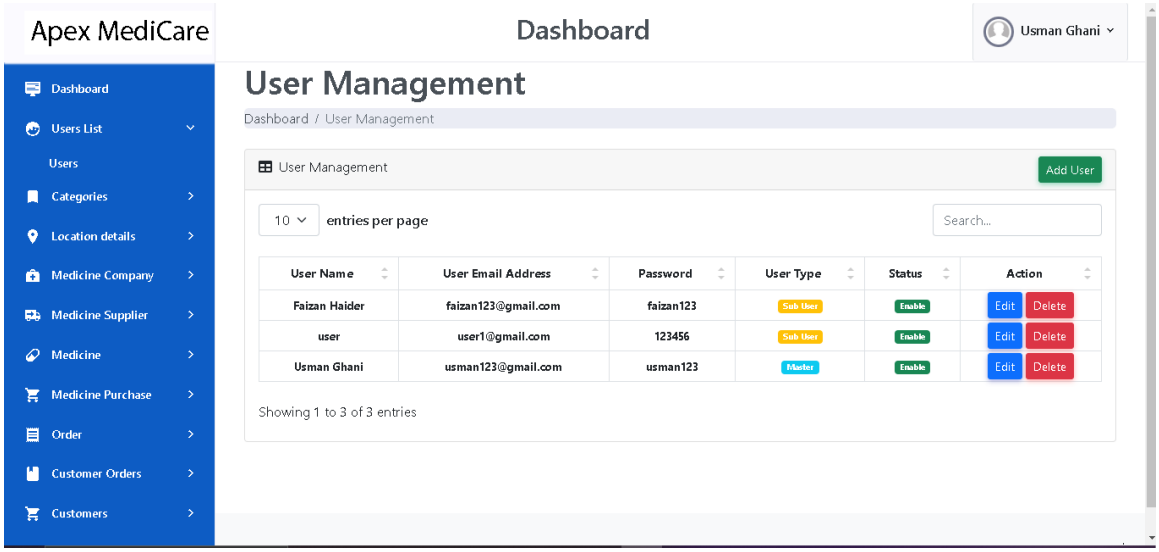


Figure 6.4 Admin Manage User List

➢ Admin Add Categories

Admin can add categories of medicine like tablets, syrup, injections, capsules and antibiotics etc.

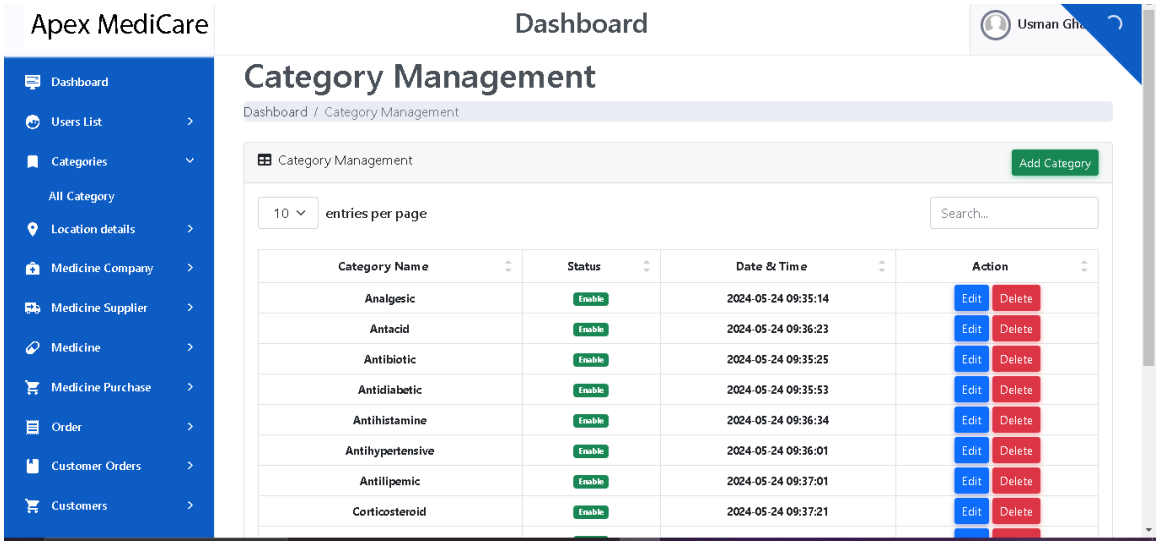


Figure 6.5 Admin Manage Category

- **Admin Manage Medicine Location**

  Admin can easily find the medicine location details in which rack medicine is available.

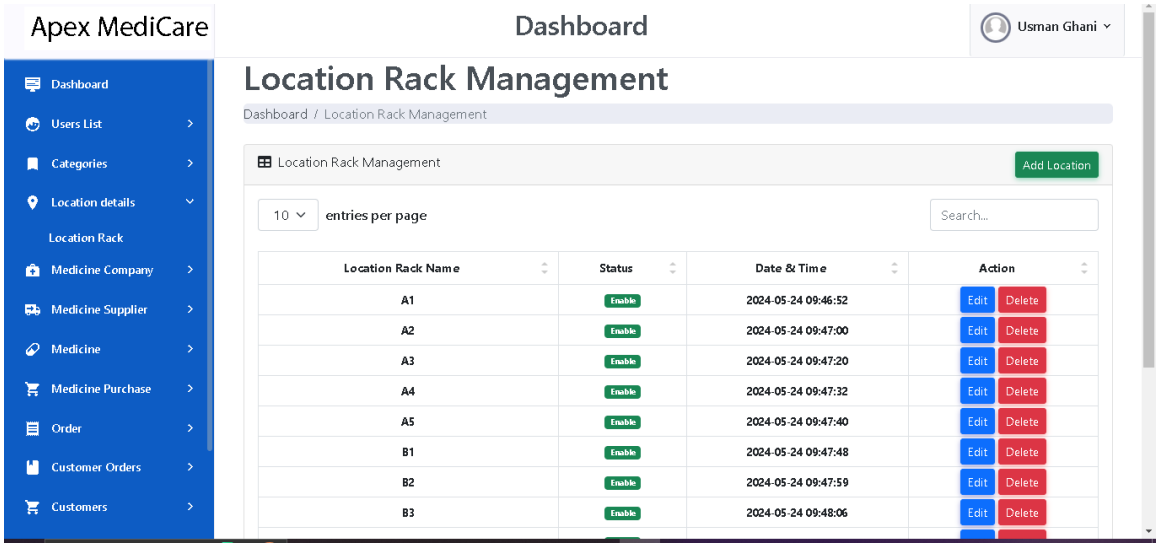


Figure 6.6 Admin Manage Medicine Location

- **Admin Manage Medicine Company Details**

  Admin can add the all-medicine company details such as medicine name and medicine company.

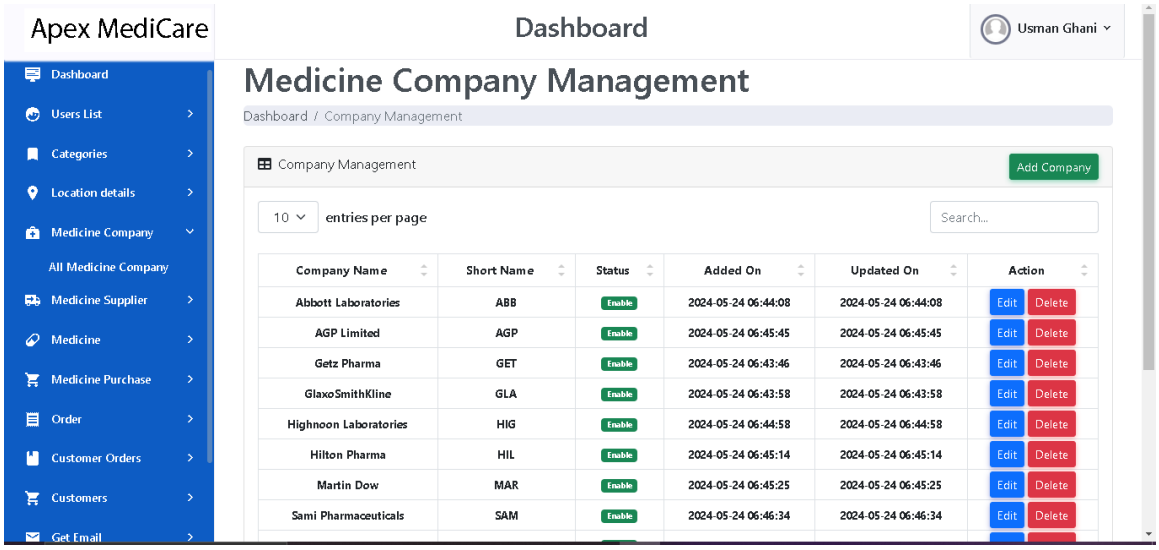


Figure 6.7 Admin Manage Medine Company Details

➢ Admin Manage Medicine Supplier

Admin can add the all-medicine supplier details such as supplier name, supplier address, supplier contact numbers, supplier email, status, date & time.

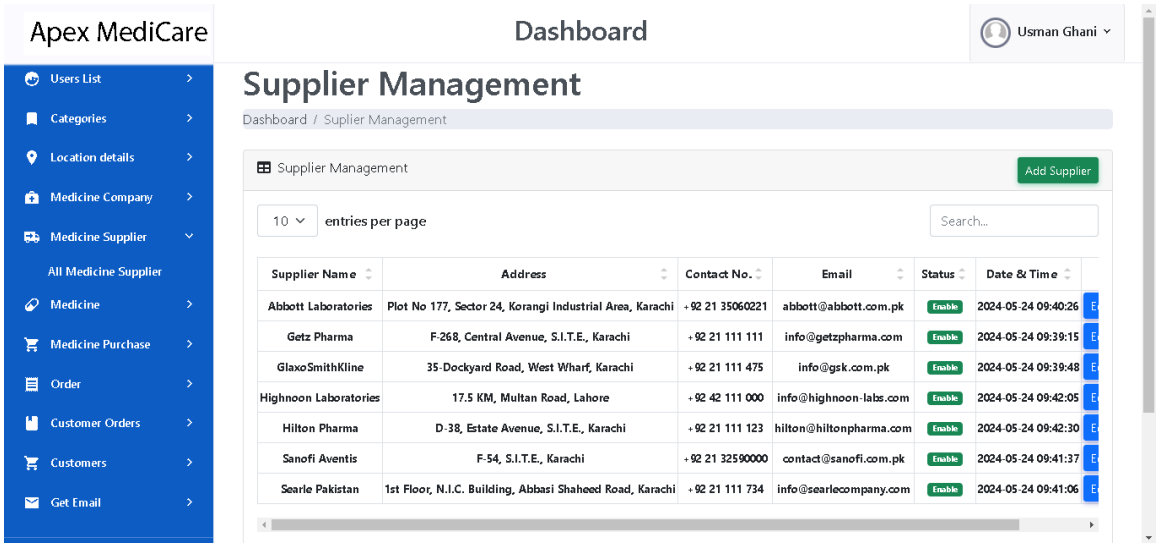


Figure 6.8 Admin Manage Medicine Supplier

➢ Admin Manage Medicine Details

Admin can medicine such as medicine name, medicine company, pack details, available Quantity, location rack, status, date & time added on, date & time updated on.

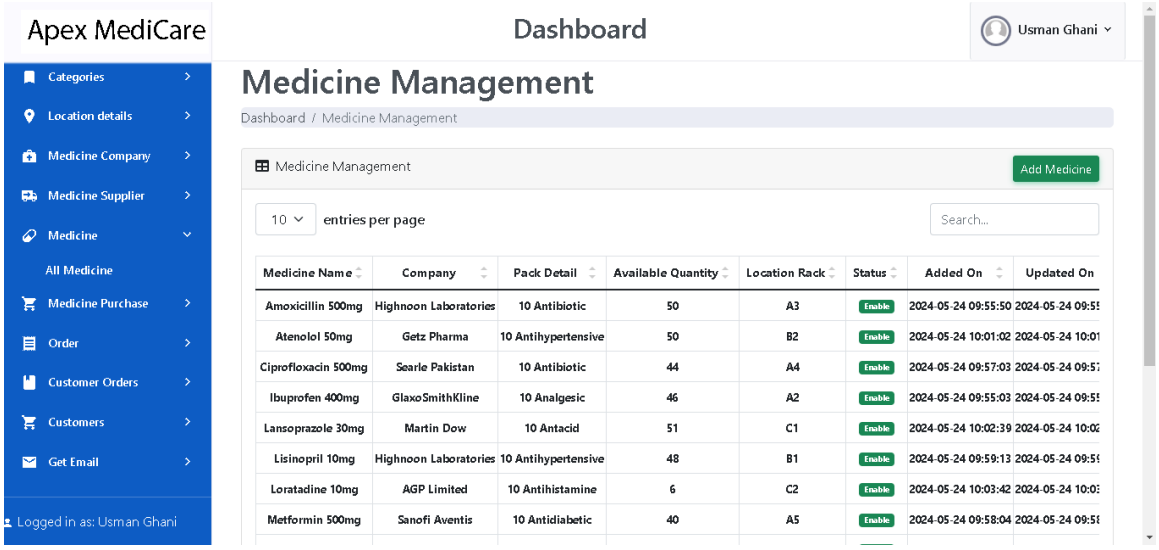


Figure 6.9 Admin Manage Medicine Details

➢ Admin Manage Medicine Purchase

Admin can manage medicine purchase such as medicine name, medicine batch no, medicine supplier, medicine quantity, available quantity, price per unit, total cost, mfg. date, medicine expiry date, medicine sales price, purchase date, status, action.

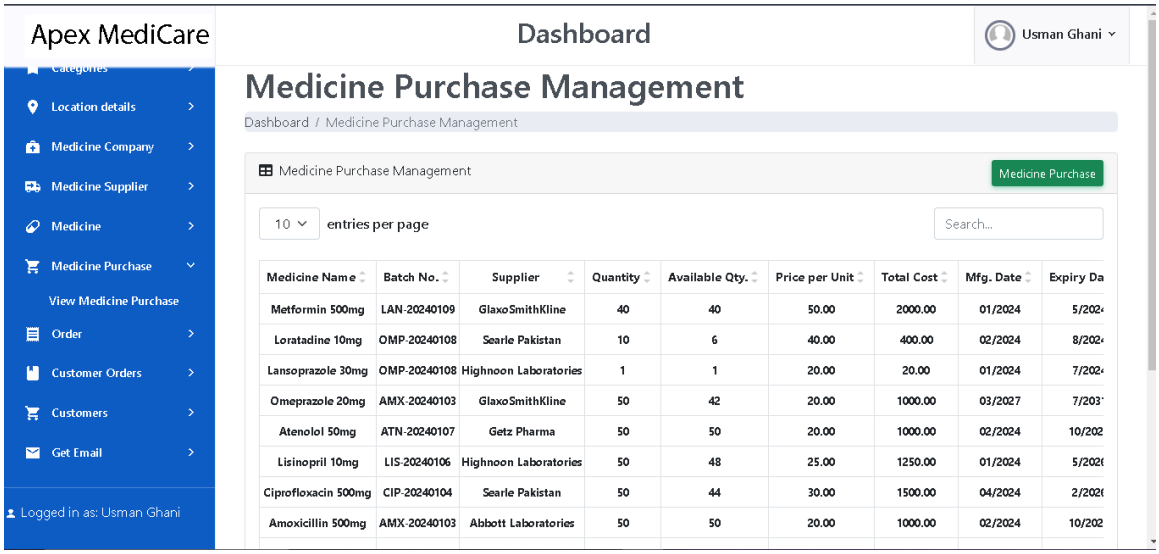


Figure 6.10 Admin Manage Medicine Purchase

➢ Admin Manage Orders

Admin can manage order bill such as patient name, doctor name, order amount, order created by, status, date & time order added on, order updated on, print order slip, edit and delete

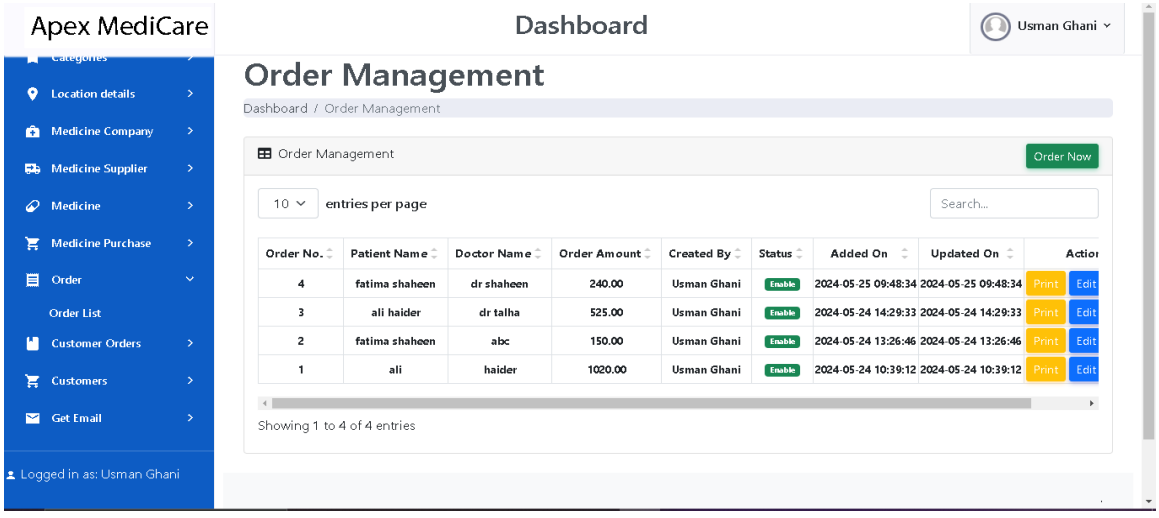
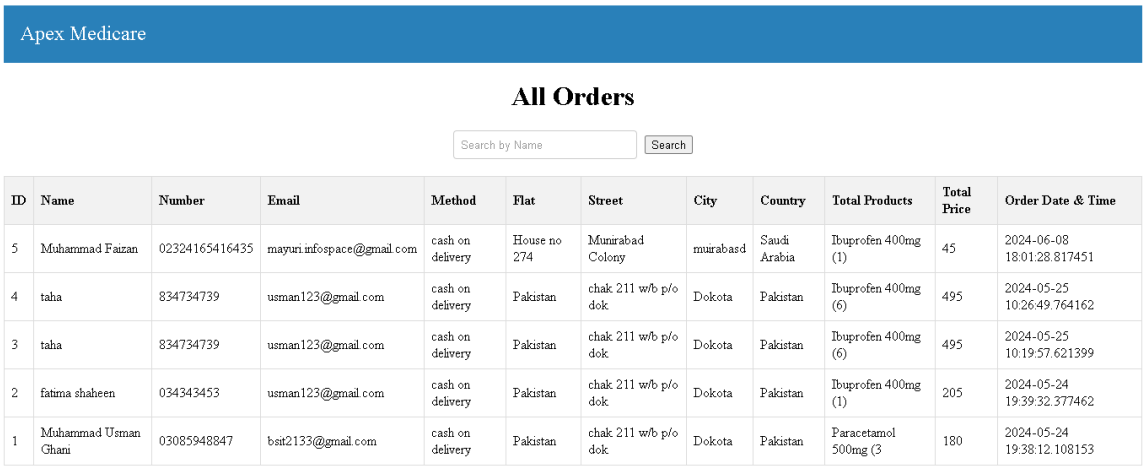


Figure 6.11 Admin Mange Orders

- **Admin Mange Customer's Orders**

  Admin can manage and view customer orders such as customer id, customer name, customer numbers, customer email, customer payment method, customer address, customer street, customer city, customer country, total products customer orders, total price customer order, customer order date & time.

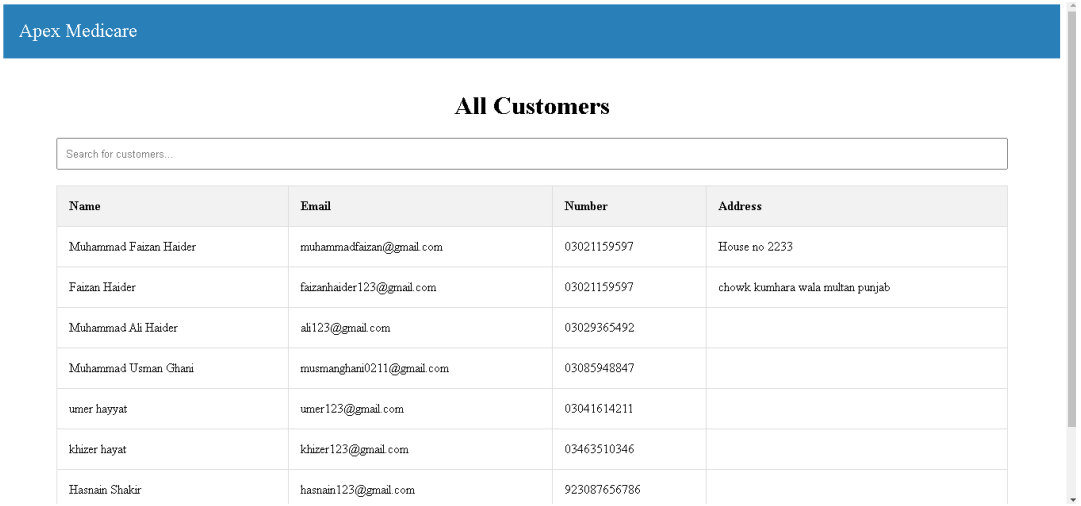


ID	Name	Number	Email	Method	Flat	Street	City	Country	Total Products	Total Price	Order Date & Time
5	Muhammad Faizan	02324165416435	mayuri.infospace@gmail.com	cash on delivery	House no 274	Munirabad Colony	muirabasd	Saudi Arabia	Ibuprofen 400mg (1)	45	2024-06-08 18:01:28.817451
4	taha	834734739	usman123@gmail.com	cash on delivery	Pakistan	chak 211 w/b p/o dok	Dokota	Pakistan	Ibuprofen 400mg (6)	495	2024-05-25 10:26:49.764162
3	taha	834734739	usman123@gmail.com	cash on delivery	Pakistan	chak 211 w/b p/o dok	Dokota	Pakistan	Ibuprofen 400mg (6)	495	2024-05-25 10:19:57.621399
2	fatima shaheen	034343453	usman123@gmail.com	cash on delivery	Pakistan	chak 211 w/b p/o dok	Dokota	Pakistan	Ibuprofen 400mg (1)	205	2024-05-24 19:39:32.377462
1	Muhammad Usman Ghani	03085948847	bsit2133@gmail.com	cash on delivery	Pakistan	chak 211 w/b p/o dok	Dokota	Pakistan	Paracetamol 500mg (3	180	2024-05-24 19:38:12.108153

Figure 6.12 Admin Manage Customer's Orders

- **Admin Manage Customer's Details**

  Admin can manage and view all customer details such as customer name, customer email address, customer number, customer address.



Name	Email	Number	Address
Muhammad Faizan Haider	muhammadfaizan@gmail.com	03021159597	House no 2233
Faizan Haider	faizanhaider123@gmail.com	03021159597	chowk kumhara wala multan punjab
Muhammad Ali Haider	ali123@gmail.com	03029365492	
Muhammad Usman Ghani	musmanghani0211@gmail.com	03085948847	
umer hayyat	umer123@gmail.com	03041614211	
khizer hayat	khizer123@gmail.com	03463510346	
Hasnain Shakir	hasnain123@gmail.com	923087656786	

Figure 6.13 Admin Manage Customer's Details

6.2 Sub User Activity

- **Sub User Login**

  Sub users can login and put data like Gmail and password.

  Username or Email: user@gmail.com

  Password: user

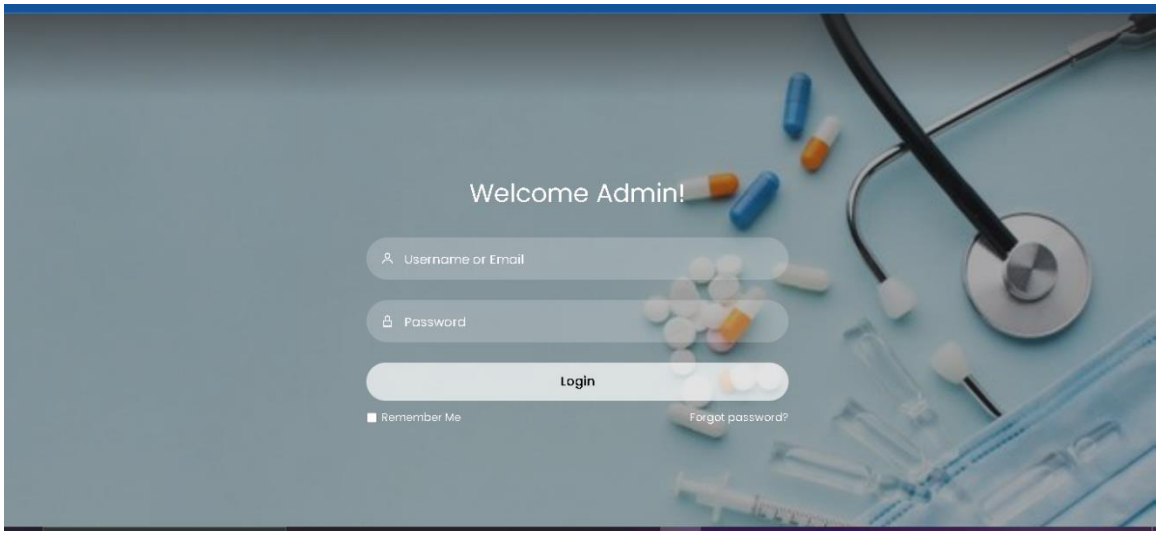


Figure 6.14 Sub User Login

- **Sub Users Manage Medicine Purchase**

  Sub users can manage medicine purchase such as medicine name, medicine batch no, medicine supplier, medicine quantity, available quantity, price per unit, total cost, mfg. date, medicine expiry date, medicine sales price, purchase date, status, action.

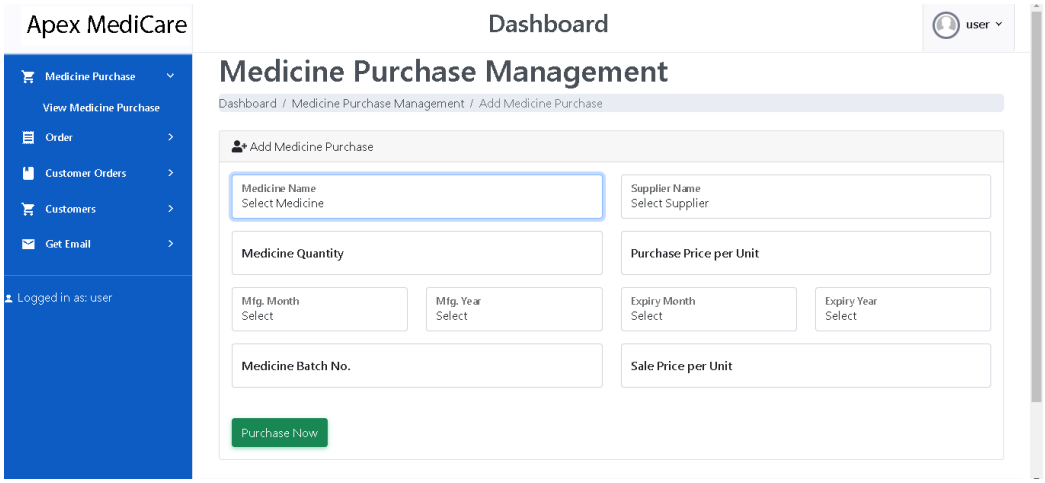


Figure 6.15 Sub User Manage Medicine Purchase

➢ Sub User Manage Order

Sub users can manage order bill such as patient name, doctor name, order amount, order created by, status, date & time order added on, order updated on, print order slip, edit and delete.

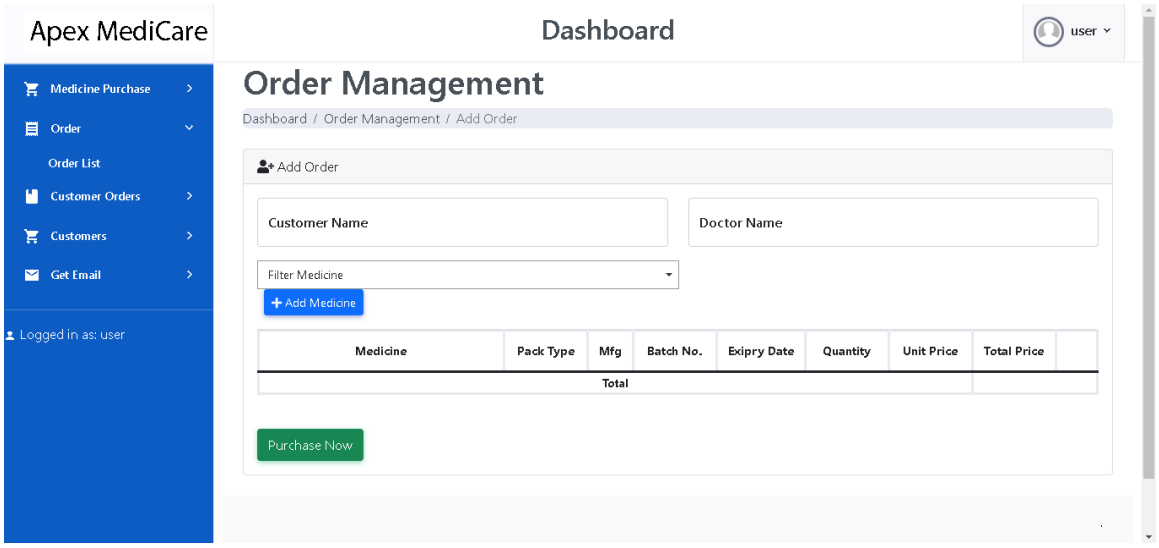
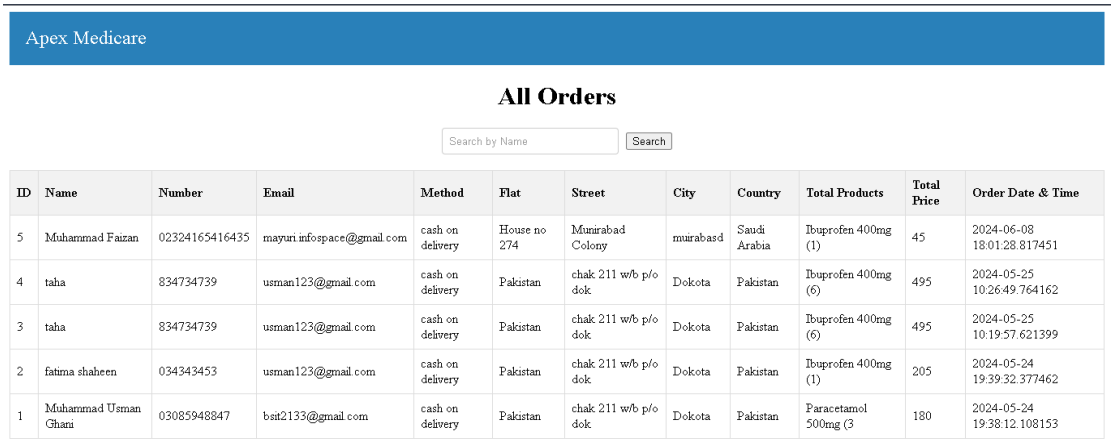


Figure 6.16 Sub User Manage Order

➢ Sub User Manage Customer's Order

Admin can manage and view customer orders such as customer id, customer name, customer numbers, customer email, customer payment method, customer address, customer street, customer city, customer country, total products customer orders, total price customer order, customer order date & time.



ID	Name	Number	Email	Method	Flat	Street	City	Country	Total Products	Total Price	Order Date & Time
5	Muhammad Faizan	02324165416435	mayuri.infospace@gmail.com	cash on delivery	House no 274	Munirabad Colony	muirabasd	Saudi Arabia	Ibuprofen 400mg (1)	45	2024-06-08 18:01:28.817451
4	taha	834734739	usman123@gmail.com	cash on delivery	Pakistan	chak 211 w/b p/o dok	Dokota	Pakistan	Ibuprofen 400mg (6)	495	2024-05-25 10:26:49.764162
3	taha	834734739	usman123@gmail.com	cash on delivery	Pakistan	chak 211 w/b p/o dok	Dokota	Pakistan	Ibuprofen 400mg (6)	495	2024-05-25 10:19:57.621399
2	fatima shaheen	034343453	usman123@gmail.com	cash on delivery	Pakistan	chak 211 w/b p/o dok	Dokota	Pakistan	Ibuprofen 400mg (1)	205	2024-05-24 19:39:32.377462
1	Muhammad Usman Ghani	03085948847	bsit2133@gmail.com	cash on delivery	Pakistan	chak 211 w/b p/o dok	Dokota	Pakistan	Paracetamol 500mg (3	180	2024-05-24 19:38:12.108153

Figure 6.17 Sub User Manage Customer's Order

- **Sub User Manage Customer's Details**

  Sub users can manage and view all customer details such as customer name, customer email address, customer number, customer address.

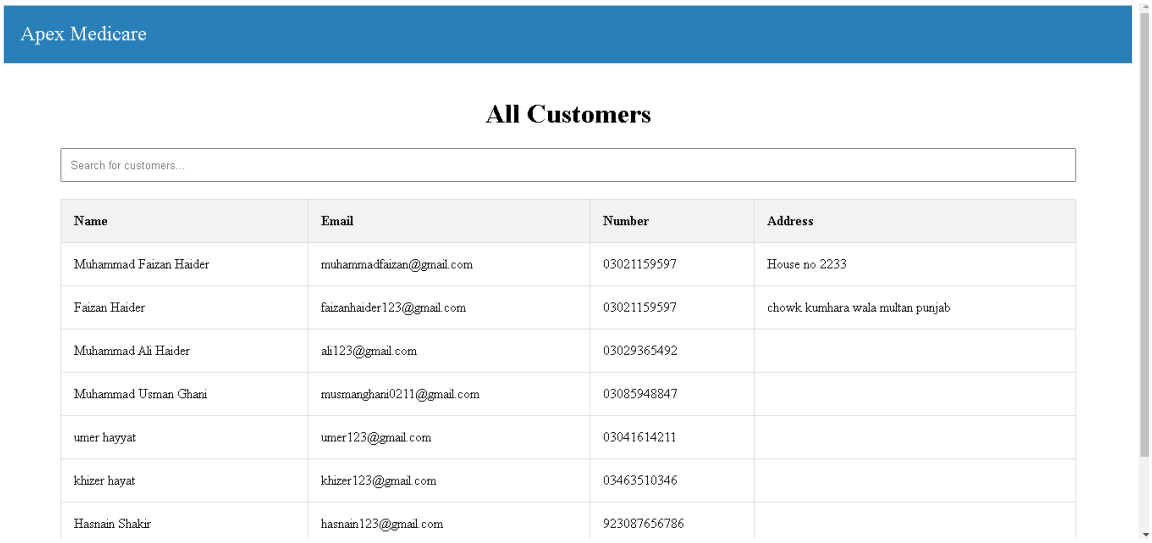


Figure 6.18 Sub User Manage Customer's Details

6.3 Customer's User Activity

- **Customer's User Register**

  Will be able to search about medicine specifications and make medicine purchase online. Customer can register and put all mandatory data like Name, Gmail, number, address and password etc.

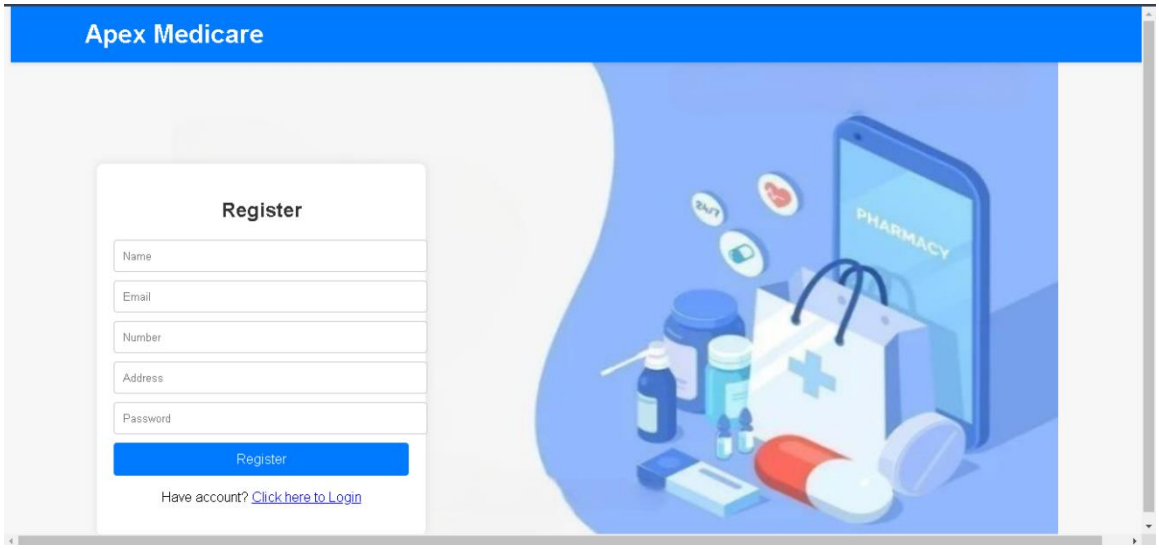


Figure 6.19 Customer's User Register

- **Customer's User Login**

  User can register him/herself on the Website by giving credential. If user has already an account, then he just login.

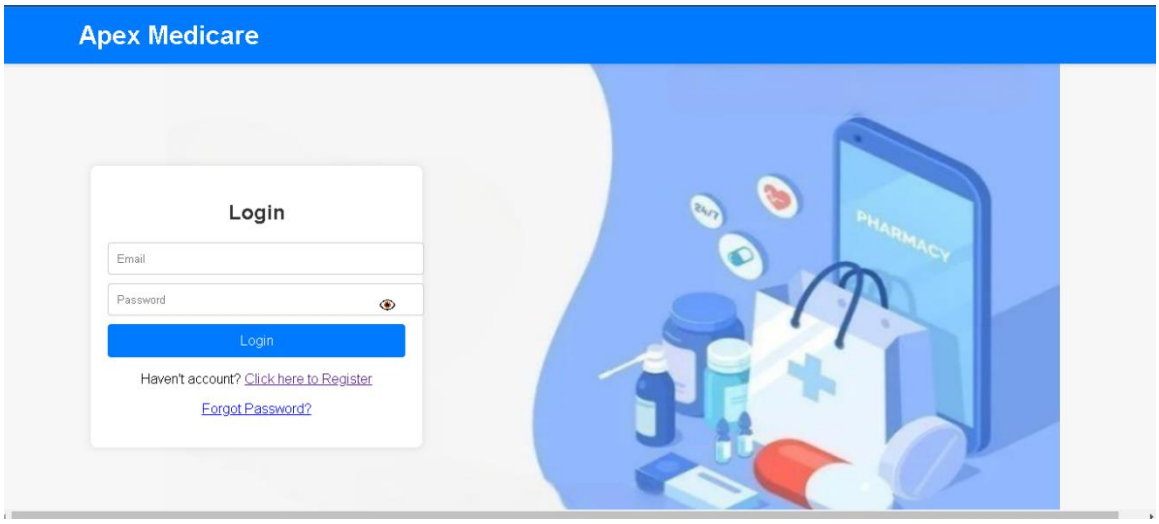


Figure 6.20 Customer's User Login

➢ Customer's User Search Medicine

In this screen user can search the desire medicine view medicine price and medicine picture

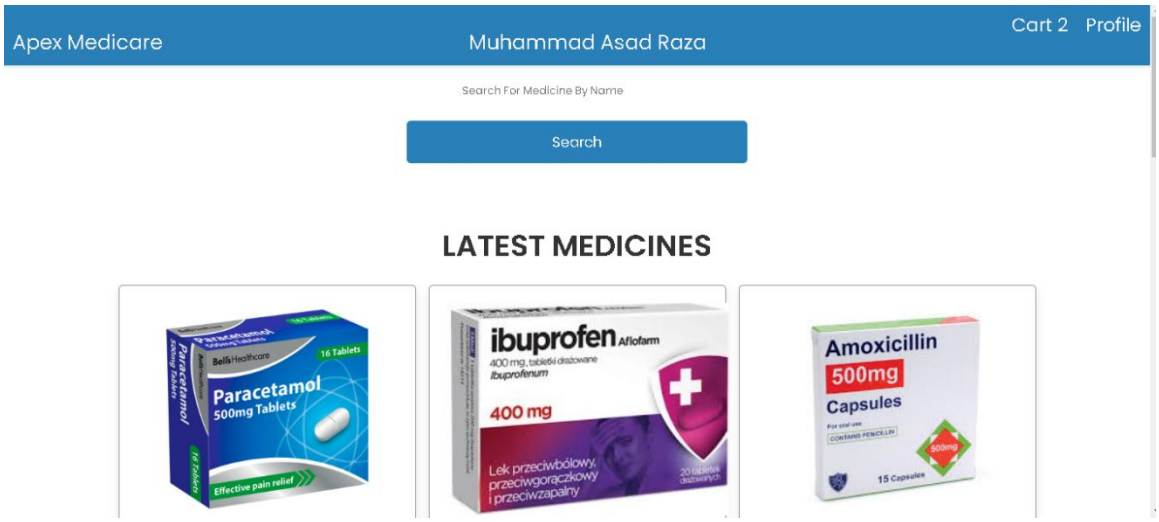


Figure 6.21 Customer's User Search Medicine

➢ Add to Cart and contact detail

Customer can add to cart desired medicine and provide detail of contact.

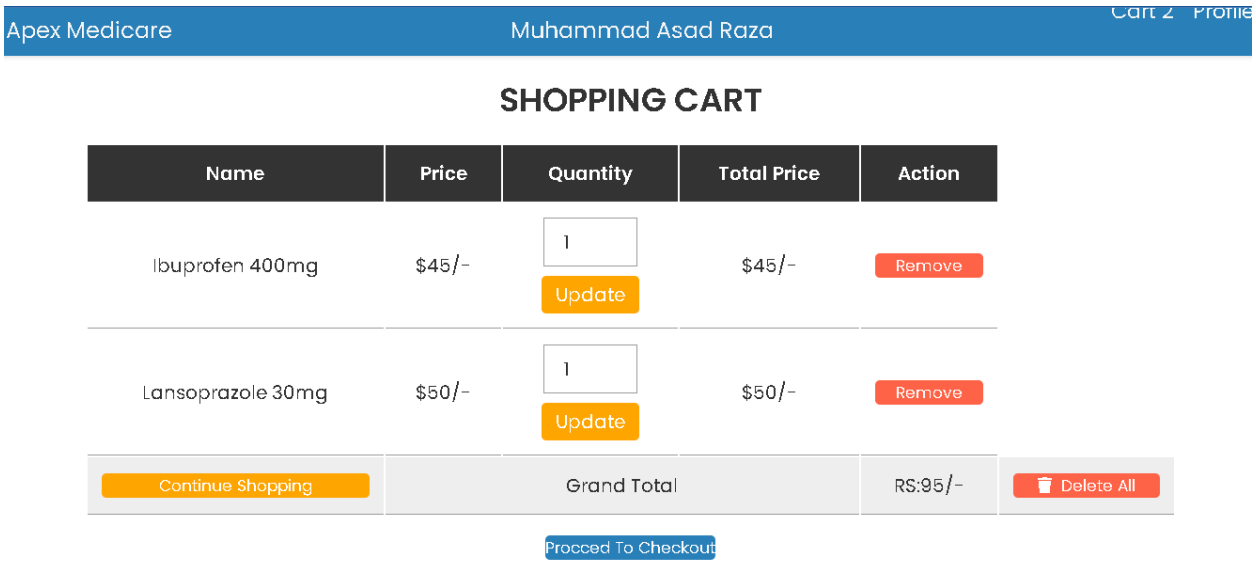


Figure 6.22 add to Cart Medicine

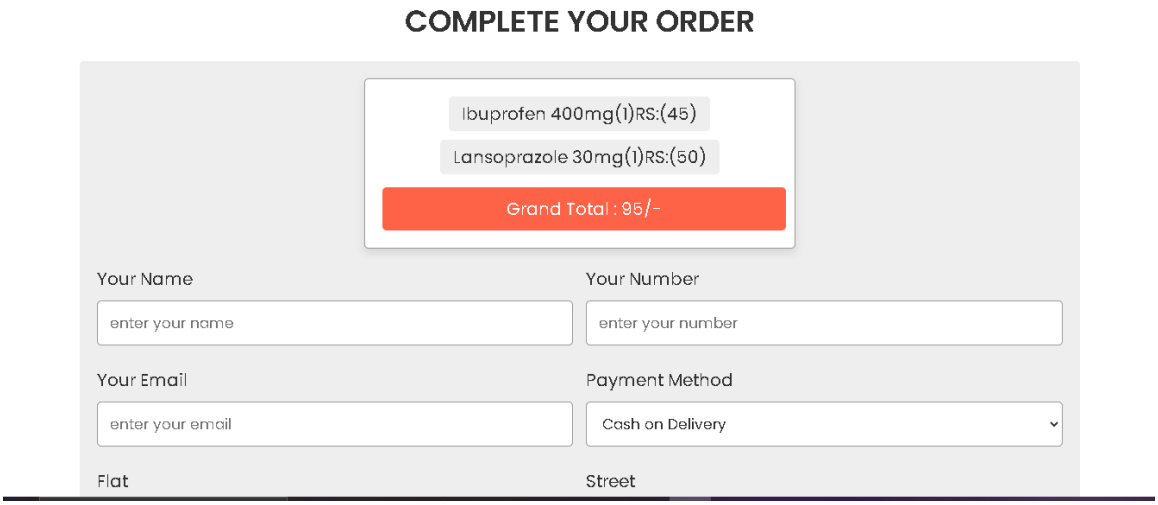


Figure 6.23 Complete Your order

Summary

Chapter 6 is the user guide for the Smart Safe web-platform, where it describes all the functionalities and activities for admin, sub-user, and customer. It provides functions like logging into the system, user management, category management, medicine details management, purchase and order management, and customer management with the help of screenshots for better understanding. Sub-user activities include being in charge of purchasing medicines and orders, while customer activities include registering, logging in, searching for the medicines, and making the purchase. The chapter helps users in the application to be able to interact with the application with ease.

Chapter 7

CONCLUSION & References

Conclusion

“Smart Safe" is a transformative web-based designed to update and optimize operations for pharmacy management systems. It significantly reduces manual errors by offering a complete platform for both users and administrators. Reduces costs and improves efficiency in many aspects of pharmacy management, including sales, inventory, reporting, and billing. Users benefit from an intuitive interface that allows them to easily find medications, access detailed information, and allow easy and secure ordering. The inclusion of advanced features such as password recovery, profile management, and multiple payment options further enhances the user experience. Administrators are equipped with the best tools to monitor and manage the entire operation, from tracking sales trends through detailed graphical analytics to generating reports and verifying systemic responsiveness. The platform's alert notifications for low stock and upcoming expirations ensure proactive inventory management, while QR code payment functionality ensures secure and smooth transactions. In short, "Smart Safe" sets a new standard for medical store management, providing a robust, user-friendly solution that promises to increase accuracy, efficiency, and relevance in the delivery of healthcare services.

References

A wide range of web resources were used during the development of this project following are some of the references we used:

1) “Pharmacy Management System Software – IT VISION | Web & Software Technology,” *www.itvision.com.pk*. https://www.itvision.com.pk/pharmacy-management-system-software.

2) “PHARMACY DASHBOARD,” *Dribble*. https://dribbble.com/shots/8659613-PHARMACY-DASHBOARD.

3) “eCommerce Archives,” *Colorlib*. https://colorlib-com.webpkgcache.com/doc/-/s/colorlib.com/wp/cat/ecommerce/

4) “Dynamically generating a QR code with PHP,” *Stack Overflow*. https://stackoverflow.com/questions/5943368/dynamically-generating-a-qr-code-with-php.

5) “Find Icons with the Perfect Look & Feel | Font Awesome,” *fontawesome.com*. https://fontawesome.com/icons/

6) “chillerlan/php-qrcode,” *GitHub*, Sep. 06, 2023. https://github.com/chillerlan/php-qrcode.

7) altexsoft, “Pharmacy Management System: Benefits, Features, and Providers of Pharmacy Software,” *AltexSoft*, Jul. 13, 2021. https://www.altexsoft.com/blog/pharmacy-management-system/

8) “Creating Real Time Notification System in PHP and AJAX,” *www.cloudways.com*, Jul. 04, 2017. https://www.cloudways.com/blog/real-time-php-notification-system/

9) “Creating Real Time Notification System in PHP and AJAX,” *www.cloudways.com*, Jul. 04, 2017. https://www.cloudways.com/blog/real-time-php-notification-system/

10) HTTP authentication with PHP,” *php-legacy-docs.zend.com*. https://php-legacy-docs.zend.com/manual/php4/en/features.http-auth