

VISVESVARAYA TECHNOLOGICAL UNIVERSITY
“JNANA SANGAMA”, BELAGAVI - 590 018



A MINI PROJECT REPORT

on

“Pharmacy Management System”

Submitted by

Karthik J

4SF20IS041

Mohammed Uyam Abbas

4SF20IS055

In partial fulfillment of the requirements for the VI semester

File Structures Laboratory With Mini Project(18ISL67)

of

BACHELOR OF ENGINEERING

in

INFORMATION SCIENCE & ENGINEERING

Under the Guidance of

Ms. Jayapadmini Kanchan

Assistant Professor, Department of ISE

at



SAHYADRI

College of Engineering & Management

An Autonomous Institution

MANGALURU

2022 - 23

SAHYADRI
College of Engineering and Management
Adyar, Mangaluru - 575 007

Department of Information Science & Engineering



CERTIFICATE

This is to certify that the mini project entitled “**Pharmacy Management System**” has been carried out by **Karthik J (4SF20IS041)** and **Mohammed Uyam Abbas (4SF20IS055)** the bonafide students of Sahyadri College of Engineering and Management, Bachelor of Engineering in Information Science & Engineering of Visvesvaraya Technological University, Belagavi during the year 2022-23. It is certified that all corrections / suggestions indicated for internal assessment have been incorporated in the report deposited in the departmental library. The mini project report has been approved as it satisfies the academic requirements in respect of mini project work prescribed in File Structures Laboratory with Mini Project(18ISL67) for the said degree in sixth semester.

Signature of the Guide1
Ms. Jayapadmini Kanchan

Signature of the Guide2
Mrs. Madhura N Hegde

Signature of the HOD
Dr. Mustafa Bhastikodi

External Viva:

Examiner's Name

Signature with Date

1.

.....

2.

.....

SAHYADRI
College of Engineering and Management
Adyar, Mangaluru - 575 007

Department of Information Science & Engineering



DECLARATION

We hereby declare that the entire work embodied in this Mini Project Report titled “**Pharmacy Management System**” has been carried out by us at Sahyadri College of Engineering and Management, Mangaluru under the supervision of **Ms. Jayapadmini Kanchan**, for **Bachelor of Engineering in Information Science & Engineering**. This report has not been submitted to this or any other University for the award of any other degree.

Karthik J (4SF20IS041)

Mohammed Uyam Abbas (4SF20IS055)

Dept. of ISE, SCEM, Mangaluru

Abstract

Pharmacy is a crucial field that plays a vital role in healthcare, focusing on the safe and effective use of medications. Pharmacists serve as experts in medication management, providing guidance and support to individuals in maintaining their health and well-being. This system utilizes primary indexing, a fundamental method for efficient data organization, to manage stocks of medicines effectively. The goal of a pharmacy management system is to streamline and enhance the overall efficiency and accuracy of pharmacy operations. By maintaining a comprehensive record of day-to-day transactions for medicinal products, this system ensures smooth operations and facilitates seamless customer service. The scope of a pharmacy management system typically includes functions such as inventory management, billing and payment processing. Pharmacists face the challenge of managing a vast number of records, which can be a time-consuming task. However, with the aid of automated features, they can efficiently handle large volumes of data and organize it systematically. These features simplify the process of storing records in a structured manner, making it easier for pharmacists to access specific medications whenever needed.

Acknowledgement

It is with great satisfaction and euphoria that we are submitting the Mini Project Report on “**Pharmacy Management System**”. We have completed it as a part of the curriculum of Visvesvaraya Technological University, Belagavi for the award of Bachelor of Engineering in Information Science & Engineering.

We are profoundly indebted to our guide, **Ms. Jayapadmini Kanchan**, Assistant Professor, Department of Information Science & Engineering for innumerable acts of timely advice, encouragement and we sincerely express our gratitude.

We express our sincere gratitude to **Dr. Mustafa Basthikodi**, Professor and Head, Department of Information Science & Engineering for his invaluable support and guidance.

We sincerely thank **Dr. Rajesha S**, Principal, Sahyadri College of Engineering and Management who have always been a great source of inspiration.

Finally, yet importantly, we express our heartfelt thanks to our family and friends for their wishes and encouragement throughout the work.

Karthik J (4SF20IS041)

Mohammed Uyam Abbas (4SF20IS055)

Table of Contents

Abstract	i
Acknowledgement	ii
Table of Contents	iv
List of Figures	v
1 Introduction	1
1.1 Overview	1
1.2 Scope and Purpose	2
2 Requirements Specification	3
2.1 Hardware Specification	3
2.2 Software Specification	3
3 System Design	4
3.1 Architecture Diagram	4
3.2 Application Modules	5
4 Implementation	6
4.1 Languages Used	6
4.1.1 C++	6
4.1.2 Storage	6
4.2 Pseudocode	7
4.2.1 Pseudocode for Daily Summary of Orders:	7
4.2.2 Pseudocode for Deleting Customer Order:	8
4.2.3 Pseudocode for Displaying Customer:	8
4.2.4 Pseudocode for Displaying Medicines:	9
4.2.5 Pseudocode for Modifying Customer Order:	9
4.2.6 Pseudocode for Printing Order Receipt:	10

4.2.7	Pseudocode for Searching Medicine:	10
4.2.8	Pseudocode for Taking Customer Order:	11
5	Results and Discussion	12
5.1	Menu Page	12
5.2	Displaying Available Medicine	13
5.3	Taking Customer Order	14
5.4	Payment	15
5.5	Displaying Customer Detail	16
5.6	Daily Summary	17
6	Conclusion	18
	References	19

List of Figures

3.1	Architecture Diagram for Pharmacy Management System	4
4.1	Pseudocode for Daily Summary of Orders	7
4.2	Pseudocode for Deleting Customer Order	8
4.3	Pseudocode for Displaying Customer	8
4.4	Pseudocode for Displaying Medicines	9
4.5	Pseudocode for Modifying Customer Order	9
4.6	Pseudocode for Printing Order Receipt	10
4.7	Pseudocode for Searching Medicine	10
4.8	Pseudocode for Taking Customer Order	11
5.1	Menu Page	12
5.2	Displaying Available Medicine	13
5.3	Taking Customer Order	14
5.4	Payment	15
5.5	Displaying Customer Detail	16
5.6	Daily Summary	17

Chapter 1

Introduction

This System has been developed to override the problems prevailing in the manual system. This software is supported to eliminate and in some cases reduce the hardship faced by the existing system. The software is reduced as much as to avoid errors while entering the data. It is a user-friendly software. As described, it can lead to be an error free, secure, fast management system. It will assist the user to concentrate more on other works rather than concentrating and spending their valuable time in keeping records. Every pharmacy, has a challenge to overcome and managing the information of stocks, medicines, billing receipt.

Every user has their own specific needs. Thus the system is designed as a user-friendly interface that are adapted to user requirements. Whether its managing stocks or pharmacy bill receipt, all these requirements are fulfilled in this system leading to work and time conserving system. Ultimately, this system will help the pharmacy to better manage their resources.

1.1 Overview

The size of data is growing at a much higher rate, it becomes very important to create new systems which can handle high volume of data without affecting the performance and management of those data. In these tough times when the need of doctors and medicines are high, managing all the data of stocks, medicines and also receipts of the pharmacy is very time and energy consuming. This system overcomes all those problems and will manage all the data for a long period and with way less human effort.

1.2 Scope and Purpose

It may help collecting perfect management in details. It helps in keeping track of the management of past and present ongoing management by helping the pharmacist to know the management and works perfectly. The project aims at automating the process i.e. the system is computerized with various process from managing the stocks, taking medicine order, modifying the orders to printing the bill receipt of the pharmacies.

The main purpose of this system is to manage the data or information of stocks, availability of medicines and the bill. The project is built according to the needs of the pharmacist so that they can use the system according their specific requirements. The ultimate purpose of the project is to reduce work and time that is given in the existing system and to allow the users to use that valuable resource in their other tasks.

Chapter 2

Requirements Specification

2.1 Hardware Specification

- Processor : Intel(R) Core(TM) i7-1005G1 CPU @ 1.20GHz
- RAM : 16GB
- Hard Disk : 512GB
- Input Device : Standard keyboard and Mouse
- Output Device : Monitor

2.2 Software Specification

- Programming Language : C++
- IDE :Visual Code Version 1.79.2
- Compiler:MinGW 11.0.0
- OS:Windows 11

Chapter 3

System Design

3.1 Architecture Diagram

The architecture diagram for this System typically consists of multiple components working together to provide a cohesive software solution.

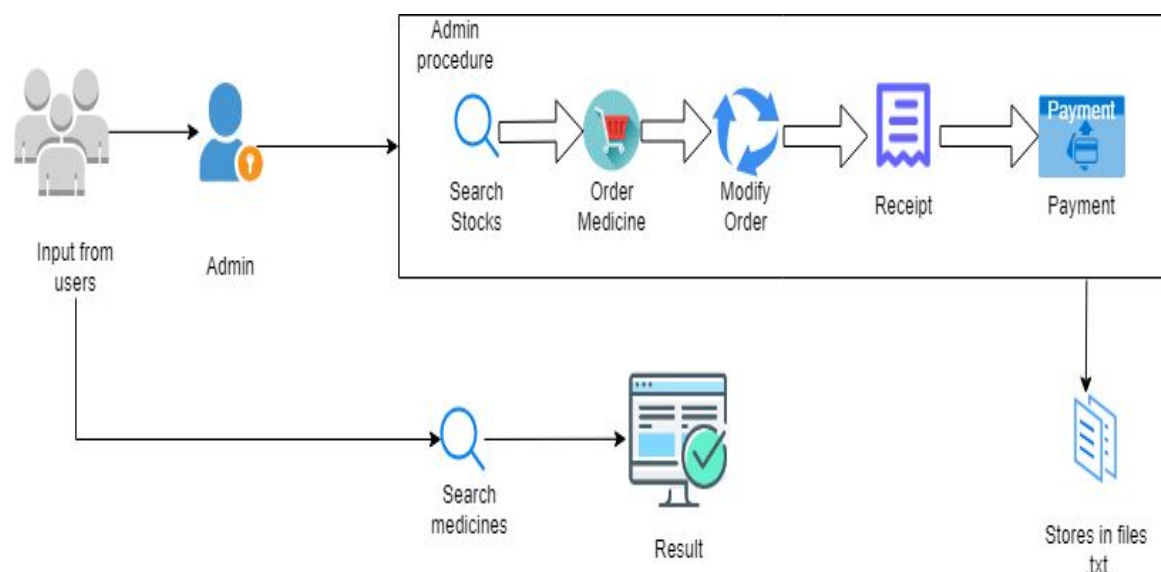


Figure 3.1: Architecture Diagram for Pharmacy Management System

The core of the system is a centralized database that stores various informations. Pharmacist will interact with the system through user interface, and can take the required orders. Once the order is taken the system will generate a receipt based on the order. The pharmacist can modify the orders if required. Pharmacist can also insert, delete and search for available stocks. At the end of the day all the details about the details of the receipt will be stored in a file.

3.2 Application Modules

- **Search Module:** This module handles searching for medicine in the file structure based on various criteria such as medicine name, dosage, or other attributes.
- **Display Module:** This module is responsible for displaying the available medicines from the file structure, providing information such as medicine name, quantity, price, and other relevant details.
- **Order Module:** This module handles taking customer orders for medicine. It allows customers to select medicines, specify quantities, and add them to their order.
- **Modification Module:** This module facilitates modifying customer orders. It enables users to update quantities, remove medicines, or make any necessary changes to the order.
- **Receipt Module:** This module generates a receipt for the customer's order. It includes details such as the ordered medicines, quantities, prices, total amount, and any applicable taxes or discounts.
- **Payment Module:** This module handles the payment process for the customer's order. It could interact with a payment gateway or integrate with a payment system to accept various payment methods (e.g., credit cards, online banking, etc.).

Chapter 4

Implementation

4.1 Languages Used

4.1.1 C++

C++ is a powerful and versatile programming language widely used for developing a wide range of software applications. It is an extension of the C programming language with added features, including object-oriented programming (OOP) capabilities. C++ combines low-level programming control with high-level abstractions, making it suitable for both system-level programming and application development. C++ is known for its efficiency and performance, as it allows programmers to directly manipulate memory and hardware resources. It supports features such as pointers, which enable direct memory access and efficient memory management.

4.1.2 Storage

Backend storage in Windows often relies on text files, which serve as a fundamental method for storing and manipulating plain text data. These files contain easily readable characters and can be effortlessly created, edited, and viewed using diverse text editors and word processors. In Windows, text files typically carry the ".txt" extension and boast simplicity and versatility, ensuring compatibility across various software applications and operating systems. They find extensive use in storing program source code, configuration files, log files, and other forms of textual data. Notably, text files are lightweight, portable, and facilitate easy information sharing, making them a convenient choice in numerous contexts for storing and exchanging data.

4.2 Pseudocode

4.2.1 Pseudocode for Daily Summary of Orders:

The Daily Summary of orders provides a concise overview of the prescription orders processed within a given day. It generates a report that summarizes key information related to the orders, such as the total number of orders received, the number of orders dispensed, and any orders pending or canceled. The summary may also include details like the average processing time per order or any significant issues or trends observed during the day.

```
void pharmacyManagementSystem::dailySummary()
{
    int i, num;
    system("CLS");
    node *temp;
    temp = start_ptr;
    float totalPrice = 0;

    if (temp == NULL) // Invalid Receipt Code
    {
        cout << "No Orders To Show!\n"
              << endl;
    }
    else
    {
        cout << "\n";
        cout << "===== " << endl;
        cout << "Daily Summary Of All Orders \n"; // Print All Transactions Occurred
        cout << "===== " << endl;

        while (temp != NULL)
        {
            cout << "Reciept Number : " << temp->recieptNumber;
            cout << "\n";
            cout << "Customer Name: " << temp->customerName << endl;
            cout << "Order Date : " << temp->date << endl;
            cout << "_____ " << endl;
            cout << "===== " << endl;
            cout << "| Medicine Name | Quantity | Total Price |" << endl;
            cout << "===== " << endl;
        }
    }
}
```

Figure 4.1: Pseudocode for Daily Summary of Orders

4.2.2 Pseudocode for Deleting Customer Order:

Deleting a customer order in this System involves the removal of a previously placed order from the system. This action may occur if a customer requests to cancel their order or if an order was created in error. When deleting a customer order, the pharmacy staff accesses the system, identifies the specific order to be deleted, and initiates the deletion process. Deleting a customer order ensures accurate record-keeping and inventory management within the pharmacy, allowing for proper tracking and reconciliation of orders.

```
void pharmacyManagementSystem::deleteOrder()
{
    system("CLS");
    int i, num, count;
    ofstream out;
    out.open("takeCustomerInfo.txt", ios::out | ios::app);

    cout << "Enter the Receipt Number You Wish To Delete : ";
    cin >> num;
    node *q;
    node *temp;
    bool found;

    if (start_ptr == NULL)
        cerr << "Empty List, No Entries Detected To Delete\n";
    else
    {
        if (start_ptr->receiptNumber == num)
        {
            q = start_ptr;
            start_ptr = start_ptr->next;
            count--;
            if (start_ptr == NULL)
                last = NULL;
            delete q;
            cout << "The Reciept is Deleted Successfully!" << endl;
        }
        else
    }
}
```

Figure 4.2: Pseudocode for Deleting Customer Order

4.2.3 Pseudocode for Displaying Customer:

Displaying customers in this System involves presenting relevant customer information for efficient retrieval and reference. The system provides a user interface that allows pharmacy staff to access and view customer details in a structured manner. This information typically includes the customer's name, contact information, billing details.

```
void pharmacyManagementSystem::displayCustomers()
{
    system("CLS");
    cout << "===== " << endl;
    cout << "Back-Up Customer Details \n"; // Print All Transactions Occurred
    cout << "===== " << endl;
    string takeCustomerInfo;
    ifstream in("takeCustomerInfo.txt");
    while (!in.eof())
    {
        getline(in, takeCustomerInfo);
        cout << takeCustomerInfo << endl;
    }
    system("PAUSE");
    system("CLS");
}
```

Figure 4.3: Pseudocode for Displaying Customer

4.2.4 Pseudocode for Displaying Medicines:

Displaying medicines in this System which involves presenting a comprehensive list or catalog of available medications for efficient reference and retrieval. The system provides a user-friendly interface that allows pharmacy staff to access and view detailed information about each medicine.

```
void pharmacyManagementSystem::displayMedicine()
{
    system("CLS");
    cout << "===== " << endl;
    cout << "Medicines In Stock \n"; // Print All Medicines In Stock
    cout << "===== " << endl;
    string takeOrderText;
    ifstream in("takeOrder.txt");
    for (int i = 0; i < 14; i++)
    {
        getline(in, takeOrderText);
        cout << takeOrderText << endl;
    }
    cout << "" << endl;
    system("PAUSE");
    system("CLS");
}
```

Figure 4.4: Pseudocode for Displaying Medicines

4.2.5 Pseudocode for Modifying Customer Order:

Modifying a customer order in this System allows for necessary changes or updates to be made to an existing order. When a customer requests a modification to their order, such as changing the quantity of medication or adding/removing items, the pharmacy staff accesses the system and retrieves the relevant order details.

```
void pharmacyManagementSystem::modify()
{
    system("CLS");
    int i, sid;
    bool found;
    found = false;
    float totalPrice = 0;
    node *temp;
    temp = start_ptr;
    ofstream out;
    out.open("takeCustomerInfo.txt", ios::out | ios::app);

    cout << "Enter Receipt Number To Modify: ";
    cin >> sid;
    if (temp == NULL || sid == 0)
    {
        cout << "No Such Record Found To Modify!" << endl;
    }
    else
    {
        while (temp != NULL && !found)
        {
            if (temp->receiptNumber == sid)
            {
                found = true;
            }
            else
            {
                temp = temp->next;
            }
        }
    }
}
```

Figure 4.5: Pseudocode for Modifying Customer Order

4.2.6 Pseudocode for Printing Order Receipt:

Printing order receipts in this System involves generating a document that summarizes the details of a customer's order. The system generates a receipt that includes information such as the customer's name, date of purchase, ordered items, quantities, prices, and the total amount due. The receipt serves as a proof of purchase for the customer and aids in accurate record-keeping and financial reconciliation within the pharmacy.

```
void pharmacyManagementSystem::orderList()
{
    int i, num, num2;
    bool found = false; // Search Variable
    system("cls");
    node *temp;
    temp = start_ptr;
    float totalPrice = 0;

    cout << "Enter the Receipt Number To Print The Receipt : ";
    cin >> num2;
    cout << "\n";
    cout << "===== " << endl;
    cout << "List Of Orders :\n";
    cout << "===== " << endl;

    if (temp == NULL) // Entered Receipt Number Is Invalid
    {
        cout << "No Such Receipt Number Found In Our Record!\nKindly Enter The Correct Rec
    }
    while (temp != NULL && !found)
    {
        if (temp->receiptNumber == num2)
        {
            found = true;
        }
        else
        {
            temp = temp->next;
        }
        if (found) // Print The Receipt
    }
}
```

Figure 4.6: Pseudocode for Printing Order Receipt

4.2.7 Pseudocode for Searching Medicine:

Searching medicines in this System involves utilizing a search functionality to find specific medications. The system allows users to input keywords, such as medication names or codes, to retrieve relevant results. This feature enables quick and efficient access to specific medicines, enhancing workflow and facilitating accurate medication dispensing.

```
void pharmacyManagementSystem::search()
{
    system("CLS");
    string search;
    string line;
    int choice, i, offset, flag = 0;
    node *temp;
    temp = new node;
    cout << "===== " << endl;
    cout << "Search :\n";
    cout << "===== " << endl;
    cout << "Enter The Name Of The Medicine You Wish To Search : "; // Search For Medicine
    cin.ignore();
    getline(cin, search);
    cout << endl;
    for (i = 0; i < MAX; i++)
    {
        if (strcmp((search.c_str()), (temp->medicineName[i].c_str())) == 0)
        {
            flag = 1;
            break;
        }
    }
    if (flag)
    {
        cout << "Found" << endl;
        cout << "===== " << endl;
    }
}
```

Figure 4.7: Pseudocode for Searching Medicine

4.2.8 Pseudocode for Taking Customer Order:

Taking a customer order in a this System involves the process of recording and capturing the details of a customer's medication requirements. When a customer submits their prescription or medication request, the pharmacy staff enters the relevant information into the system, including the customer's name, medication name, dosage, and quantity. The system validates the prescription, checks for any potential drug interactions or contraindications, and updates the inventory accordingly. This process ensures accurate recording of customer orders, facilitating efficient medication dispensing and inventory management within the pharmacy.

```
// Take Order Function
void pharmacyManagementSystem::takeOrder()
{
    system("CLS");
    int i;
    float totalPrice = 0;
    string takeOrderText;
    cout << "\nSelect The Medicines That You Would Like To Buy\n";
    cout << "----- \n\n";
    node *temp;
    temp = new node;
    //ofstream out("takeCustomerInfo.txt");
    ofstream out;
    out.open("takeCustomerInfo.txt", ios::out | ios::app);
    ifstream in("takeOrder.txt");
    for (i = 0; i < 14; i++)
    {
        getline(in, takeOrderText);
        cout << takeOrderText << endl;
    }
    cout << "" << endl;
    cout << "Type Reciept Number: ";
    cin >> temp->recieptNumber;
    out << "Reciept Number : " << temp->recieptNumber << endl;
    cout << "Enter Customer Name: ";
    //cin.ignore();
    cin>>temp->customerName;
    out << "Customer Name : " << temp->customerName << endl;
    cout << "Enter Date: ";
}
```

Figure 4.8: Pseudocode for Taking Customer Order

Chapter 5

Results and Discussion

5.1 Menu Page

The Menu Page in this System serves as the main interface for navigating different functionalities and options. It provides a structured layout with categorized menus, such as customer management, medication inventory, order processing, and reporting. Users can easily access and select the desired menu to perform specific tasks, enhancing the usability and efficiency of the system.

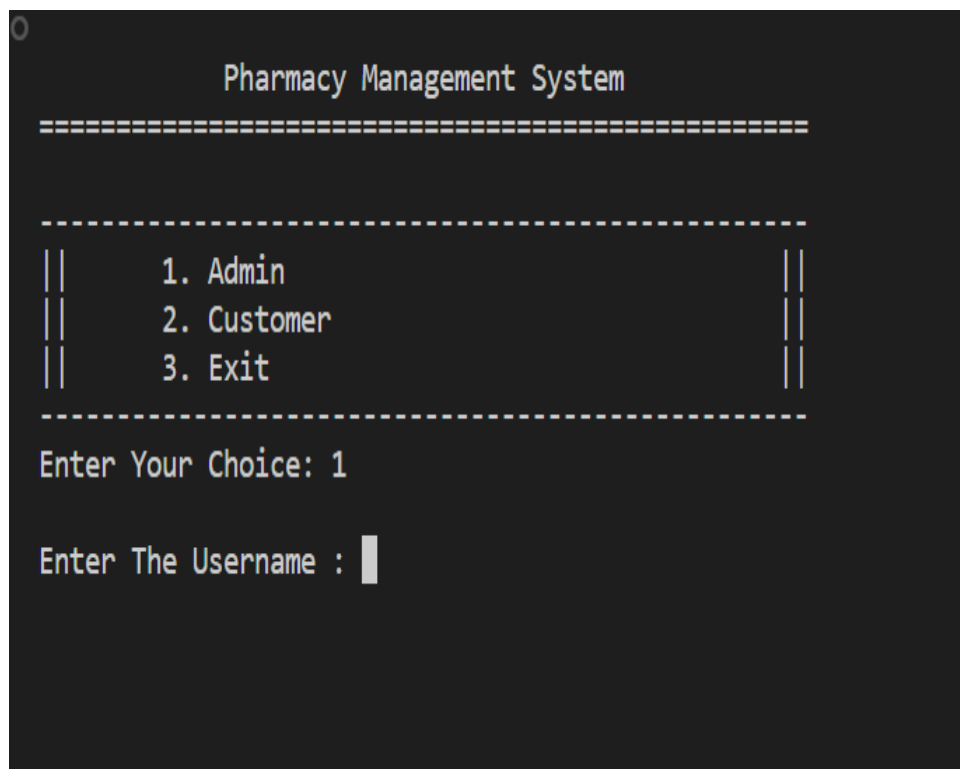


Figure 5.1: Menu Page

5.2 Displaying Available Medicine

Displaying available medicines in this System involves presenting a list of medications currently in stock. The system provides an interface that displays the medication names, quantities, and other relevant details. This feature allows pharmacy staff to quickly identify the availability of specific medicines, aiding in efficient medication dispensing and inventory management.

```

=====
Medicines In Stock
=====
*****
| DRUGS ID | DRUG'S NAME | DRUG'S PURPOSE | DRUGS PRICE(Rs.) |
*****
| 0001 | Dolo 650 | Paracetamol | Rs. 200.00 |
| 0002 | CITRA C | Vitamin C(500mg) | Rs. 200.00 |
| 0003 | Bupropion | Antidepressants | Rs. 500.00 |
| 0004 | Zoloft | Antianxiety | Rs. 400.00 |
| 0005 | Sumitop | Antimigraine | Rs. 500.00 |
| 0006 | Omeprazole | Gastric Tablets | Rs. 300.00 |
| 0007 | Digene | Antacid Tablets | Rs. 100.00 |
| 0008 | HealthVit | Multivitamin | Rs. 400.00 |
| 0009 | Naproxen | Sinus | Rs. 300.00 |
| 0010 | Neurobion | Vitamin | Rs. 300.00 |
*****

Press any key to continue . . .

```

Figure 5.2: Displaying Available Medicine

5.3 Taking Customer Order

Taking a customer order in this System involves recording the customer's prescription details, including medication name, dosage, and quantity. The system verifies the prescription, checks for any drug interactions, and updates the inventory. This process ensures accurate order management, streamlined dispensing, and effective inventory control within the pharmacy.

```

Select The Medicines That You Would Like To Buy
-----

*****
| DRUGS ID | DRUG'S NAME | DRUG'S PURPOSE | DRUGS PRICE(Rs.) |
|-----|-----|-----|-----|
| 0001 | Dolo 650 | Paracetamol | Rs. 200.00 |
| 0002 | CITRA C | Vitamin C(500mg) | Rs. 200.00 |
| 0003 | Bupropion | Antidepressants | Rs. 500.00 |
| 0004 | Zoloft | Antianxiety | Rs. 400.00 |
| 0005 | Sumitop | Antimigraine | Rs. 500.00 |
| 0006 | Omeprazole | Gastric Tablets | Rs. 300.00 |
| 0007 | Digene | Antacid Tablets | Rs. 100.00 |
| 0008 | HealthVit | Multivitamin | Rs. 400.00 |
| 0009 | Naproxen | Sinus | Rs. 300.00 |
| 0010 | Neurobion | Vitamin | Rs. 300.00 |
*****

Type Reciept Number: 2
Enter Customer Name: Deeksha
Enter Date: 29-6-2023

"NOTE : Maximum 10 Medicine Can Be Ordered Per Transaction"
Please Enter The Total Count Of Medicine You Would Like To Order: 1
Please Enter The Medicine That You Wish To Buy : 4
Medicine Name: Zoloft
Count Of The Respective Medicine : 2
You've Been Charged With Amount Of Rs. 800.00

=====
Order Placed Successfully
=====
Head To The Reciept Menu To Pay The Bill
=====
Press any key to continue . . . █

```

Figure 5.3: Taking Customer Order

5.4 Payment

Payment in a this System involves the process of settling the financial transaction for a customer's order. The system calculates the total amount due based on the medication prices and quantities. Customers can make payment through various methods like cash, credit/debit cards, or insurance processing. This feature ensures accurate and efficient payment processing, enhancing customer convenience and facilitating financial management within the pharmacy.

```
Enter the Reciept Number To Print The Reciept : 2

=====
List Of Orders :
=====
Reciept Number : 2
Customer Name: Deeksha
Order Date : 29-6-2023

=====
=====
|  Medicine Name  |      Quantity   |   Total Price   |
=====
|    Zoloft      |         2       |    Rs.800       |
=====
Total Bill :  Rs.800.00

Type In The Amount To Be Payed : 800
Payment Done Successfully!
Thank You

=====
Press any key to continue . . .
```

Figure 5.4: Payment

5.5 Displaying Customer Detail

Displaying customer details in this System involves presenting relevant information about individual customers. The system provides an interface to view details such as name, contact information, medical history, and insurance coverage. This feature enables pharmacy staff to access and reference customer information quickly, enhancing personalized care, and facilitating efficient customer service within the pharmacy.

```
Reciept Number : 2
Customer Name : Deeksha
Date : 29-6-2023
Number Of Medicine Ordered : 1
Medicine Name : Zoloft
Medicine Quantity : 2
Total Price : Rs. 800.00

=====

Reciept Number : 3
Customer Name : Karthik
Date : 29-10-2023
Number Of Medicine Ordered : 2
Medicine Name : Dolo 650
Medicine Quantity : 2
Medicine Name : Bupropion
Medicine Quantity : 1
Total Price : Rs. 900.00

=====

Reciept Number : 4
Customer Name : suhas
Date : 16-7-2023
Number Of Medicine Ordered : 3
Medicine Name : Dolo 650
Medicine Quantity : 4
Medicine Name : Dolo 650
Medicine Quantity : 2
Medicine Name : Digene
Medicine Quantity : 3
Total Price : Rs. 1500.00

=====
```

Figure 5.5: Displaying Customer Detail

5.6 Daily Summary

Daily Summary in this System provides a concise overview of daily operations. It includes information such as total prescriptions processed, medications dispensed, and revenue generated. This feature allows pharmacy staff and management to track performance, identify trends, and make informed decisions for optimal operations and financial management.

Customer Name: suhas		
Order Date : 16-7-2023		
=====		
Medicine Name	Quantity	Total Price
=====		
Dolo 650	4	Rs.800

Dolo 650	2	Rs.400

Digene	3	Rs.300

Total Bill : Rs.1500.00		

Reciept Number : 3		
Customer Name: Karthik		
Order Date : 29-10-2023		
=====		
Medicine Name	Quantity	Total Price
=====		
Dolo 650	2	Rs.400

Bupropion	1	Rs.500

Total Bill : Rs.2400.00		

Reciept Number : 2		
Customer Name: Deeksha		
Order Date : 29-6-2023		
=====		
Medicine Name	Quantity	Total Price
=====		
Zoloft	2	Rs.800

Total Bill : Rs.3200.00		

Press any key to continue . . . █		

Figure 5.6: Daily Summary

Chapter 6

Conclusion

This application has been developed to override the problems prevailing in the manual system. This software enables pharmacies to keep the record of stocks and the pharmacy receipts. This system will not only enable them to manage those data efficiently but also enable the pharmacist to look for medicines and order them. It provides facility for pharmacy to manage, view orders and also modify the orders. This system can be enhanced to enable pharmacy to manage more data on wider aspects of this field. This System can be further enhanced to encompass a wider range of data and functionalities, meeting the comprehensive needs of medical stores. Ultimately, this system is designed to optimize pharmacy operations, improve patient care, and enhance overall efficiency in the pharmacy setting. The system empowers pharmacists by providing easy access to medication information, enabling them to efficiently search for medicines and place orders. These systems will become more sophisticated, incorporating innovative technologies such as artificial intelligence, machine learning, and data analytics. They will streamline the entire pharmacy workflow, from inventory management and prescription processing to patient communication and medication adherence. Advanced algorithms will help pharmacists identify potential drug interactions and optimize medication dosages based on individual patient factors.

References

- [1] Michael J. Folk, Bill Zoellick, Greg Riccardi: File Structures-An Object Oriented Approach with C++, 3rd Edition, Pearson Education, 1998.
- [2] K.R. Venugopal, K.G. Srinivas, P.M. Krishnaraj: File Structures Using C++, Tata McGraw-Hill, 2008.
- [3] Scot Robert Ladd: C++ Components and Algorithms, BPB Publications, 1993.
- [4] Raghu Ramakrishnan and Johannes Gehrke: Database Management Systems, 3rd Edition, McGraw Hill, 2003.