VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JNANA SANGAMA", BELAGAVI - 590 018

A MINI PROJECT REPORT

on

# "Pharmacy Management System"

*Submitted by*


| | |
|---|---|
| Srivatsa Nayak | 4SF19IS106 |
| Swaroop | 4SF19IS111 |


BACHELOR OF ENGINEERING

in

INFORMATION SCIENCE & ENGINEERING

*Under the Guidance of*


**Mrs. Harinakshi C,**

Assistant Professor,

Department of ISE,


at


# SAHYADRI

College of Engineering and Management

Adyar, Mangaluru - 575 007


2021 - 22

# 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 **Srivatsa Nayak (4SF19IS106) and Swaroop (4SF19IS111)** 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 2021-22. 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** | **Signature of the Guide2** | **Signature of the HOD** |
| Mrs. Harinakshi C | Ms. Jayapadmini Kanchan | Dr. Shamanth Rai |

## 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 **Mrs. Harinakshi C,** 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.

**Srivatsa Nayak (4SF19IS106)**

**Swaroop (4SF19IS111)**

Dept. of ISE, SCEM, Mangaluru

# Abstract

The purpose of Pharmacy Management System is to automate the existing manual system by the help of computerized equipments so that their valuable data can be stored for a longer period with easy accessing and manipulation of the same. Pharmacy Management System, as it describes, can assist the user to concentrate on their other activities rather to concentrate on record keeping. Users can maintain computerized records without redundant entries. The aim of the project is to automate the existing manual system with computerized equipments so that their data can be stored and accessed easily at any point of time. Also the software provides different interface to different kinds of users to work on data like storing and accessing data more specifically and efficiently, conserving a lot of human effort and hence time.

# 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, **Mrs. Harinakshi C**, 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. Shamanth Rai**, Head and Associate Professor, Department of Information Science & Engineering for his invaluable support and guidance.

We sincerely thank **Dr. Rajesha S**, Principal, Sahyadri College of Engineering and Management and **Dr. D. L. Prabhakara**, Director, Sahyadri Educational Institutions, 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.

**Srivatsa Nayak (4SF19IS106)**

**Swaroop (4SF19IS111)**

# Table of Contents

# List of Figures

# Chapter 1

# Introduction

The "Pharmacy Management 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 Purpose

The main purpose of "Pharmacy Management 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.

## 1.2 Scope

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.

## 1.3 Overview

Since 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.

# Chapter 2

# Requirements Specification

## 2.1 Hardware Specification

- Processor : Intel(R) Core(TM) i5-8250U CPU @ 1.60GHz 1.80 GHz
- RAM : 8GB
- Hard Disk : 1TB
- Input Device : Standard keyboard and Mouse
- Output Device : Monitor

## 2.2 Software Specification

- Programming Language : C++
- IDE : Visual Studio Code

# Chapter 3

# System Design

## 3.1 Architecture Diagram

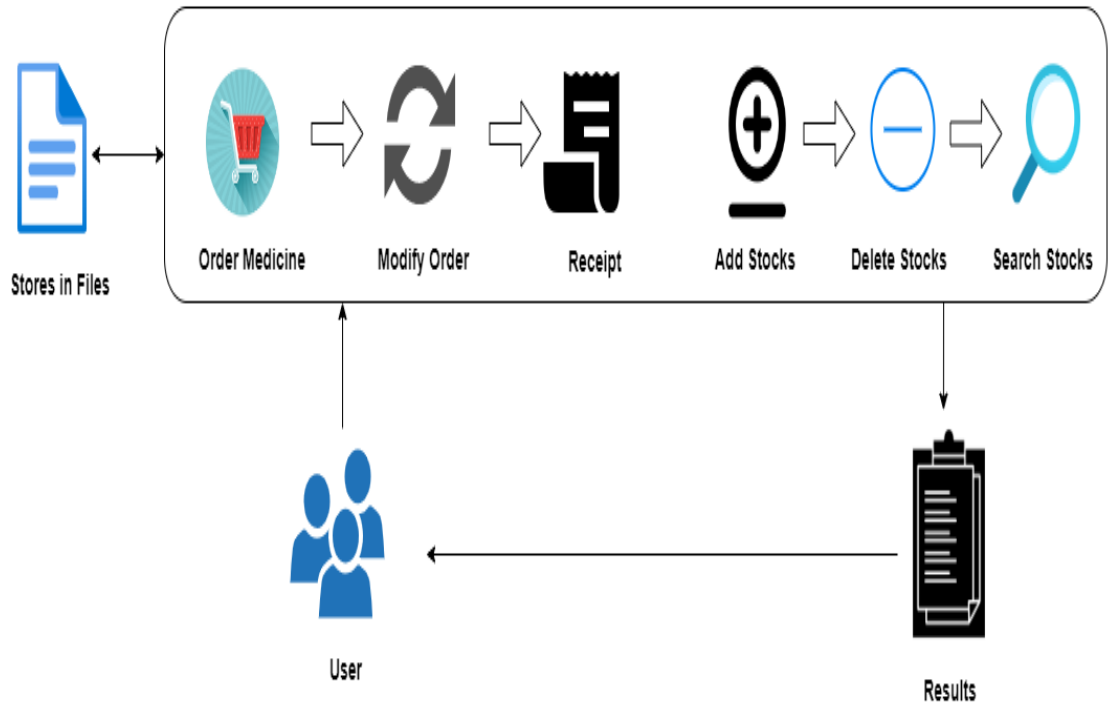


Figure 3.1: Architecture Diagram for Pharmacy Management System

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 inset, 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.

# Chapter 4

# Implementation

## 4.1 Code for Pharmacy Management System

### Code for taking the order:

The below figure is the code for taking the orders. It displays the different type of medicines available.

```
void medicineType::take_order()
{
    system("cls");
    int i;
    int choice, quantity, price, None;

    cout << "\nAdd Order Details\n";
    cout << "_____________ \n\n";

    node *temp;
    temp = new node;

    cout <<"************************\n";
                cout<<"DRUGS ID"<<"\tDRUGS TYPE"<<"   \t\tDRUGS NAME"<<"           DRUGS PRICE(Rupe)"<<endl;
                cout <<"************************\n";
                cout<<"0001"<<"\t"<<"\tOTC"<<"\t\t"<<"    Probiotics"<<"            Rupe 2.00"<<endl;
                cout<<"0002"<<"\t"<<"\tOTC"<<"\t\t"<<"    Vitamin C(500mg)"<<"      Rupe 3.00"<<endl;
                cout<<"0003"<<"\t"<<"\tOTC"<<"\t\t"<<"    Acid Free C(500mg)"<<"       Rupe 1.00"<<endl;
                cout<<"0004"<<"\t"<<"\tOTC"<<"\t\t"<<"    Women'S Multivate"<<"     Rupe 4.00"<<endl;
                cout<<"0005"<<"\t"<<"\tOTC"<<"\t\t"<<"    Marino Tablet"<<"     Rupe 1.00"<<endl;
                cout<<"0006"<<"\t"<<"\tOTC"<<"\t\t"<<"    Maxi Cal Tablet"<<"       Rupe 5.00"<<endl;
                cout<<"0007"<<"\t"<<"\tOTC"<<"\t\t"<<"    Amino Zinc Tablet"<<"     Rupe 7.00"<<endl;
                cout<<"0008"<<"\t"<<"\tOTC"<<"\t\t"<<"    Burnex"<<"            Rupe 4.00"<<endl;
                cout<<"0009"<<"\t"<<"\tOTC"<<"\t\t"<<"    Fabuloss 5"<<"            Rupe 3.00"<<endl;
                cout<<"0010"<<"\t"<<"\tOTC"<<"\t\t"<<"    Royal Propollen"<<"       Rupe 5.00"<<endl;
                cout<<" "<<endl;
    temp = new node;
    cout << "Type Order no: ";
    cin >> temp->reciept_number;
    cout << "Enter Customer Name: ";
    cin >> temp->customerName;
```

Figure 4.1: Code for taking the order

### Code for modifying the order:

The below figure is the code for modifying the orders. The user can modify the order if any changes are required.

```
void medicineType::modify()
{
    system("cls");
    int i, ch, sid;
    bool found;
    found = false;
    temp = start_ptr;
    cout << "Enter Receipt Number To Modify: ";
    cin >> sid;
    if (temp == NULL && sid == 0)
    {
        cout << "NO RECORD TO MODIFY..!" << endl;
    }

    else
    {
        while (temp != NULL && !found)
        {
            if (temp->reciept_number == sid)
            {
                found = true;
            }
            else
            {
                temp = temp->next;
            }
            if (found)
            {
                cout << "Change  Order Number: ";
                cin >> temp->reciept_number;
                cout << "Change Customer Name: ";
```

Figure 4.2: Code for modifying the order

### Code for printing the order receipt:

The below figure is the code for printing the receipt. It prints the receipt after an order has been made.

```
void medicineType::order_list()

{

    int i, num, num2;
    bool found;
    system("cls");
    node *temp;

    temp = start_ptr;
    found = false;

    cout << " Enter the Reciept Number To Print The Reciept\n";
    cin >> num2;
    cout << "\n";
    cout << "================================================================================" << endl;
    cout << "\t\tHere is the Order list\n";
    cout << "================================================================================" << endl;

    if (temp == NULL)
    {
        cout << "\tThere is no Order to show\n\t\t\tSo The List is Empty\n\n\n";
    }
    while (temp != NULL && !found)
    {
        if (temp->reciept_number == num2)
        {
            found = true;
        }
        else
```

Figure 4.3: Code for modifying the order

### Code for Daily Summary:

The below figure is the code for Daily summary. It shows the daily summary of the sales made along with the receipt number.

```
void medicineType::daily_summary()
{
    int i, num;
    system("cls");
    node *temp;

    temp = start_ptr;

    if (temp == NULL)
    {
        cout << "\t\t\tThere is no Order to show\n\t\t\tSo The List is Empty\n\n\n";
    }
    else
    {
        cout << "\n";                    (const char [76])"================================================================================"
        cout << "================================================================================" << endl;
        cout << " \t\tHere is the Daily Summary of All Orders \n"; // print all receipt
        cout << "================================================================================" << endl;

        while (temp != NULL)

        {

            ofstream fout;
            char fname[20];
            char rec[80];

            cout << "Enter a unique file name: ";
            cin >> fname;

            fout.open(fname, ios::app);
```

Figure 4.4: Code for Daily Summary

### Code for Adding Stocks:

The below figure is the code for Adding stocks. The user can add any new stocks which arrives to the pharmacy.

```
// Adding stocks
void medicineType::read_data()
{
    cout << "Medicine ID:";
    cin >> d_ID;
    cout << "Medicine Type:";
    cin >> d_TYPE;
    cout << "Medicine Name:";
    cin >> d_NAME;
    cout << "Medicine Price:";
    cin >> d_PRICE;
}
void medicineType::pack()
{
    string temp;
    buffer.erase();
    buffer += d_ID + "|" + d_TYPE + "|" + d_NAME + "|" + to_string(d_PRICE) + "$" + "\n";
}
void medicineType::write_to_file()
{
    int pos;
    fstream file;
    file.open("Add_stock.txt", ios::out | ios::app);
    pos = file.tellp();
    file << buffer;
    file.close();
    Name_list[++count] = d_NAME;
    Address_list[count] = pos;
    sort_index();
}
```

Figure 4.5: Code for modifying the order

### Code for Searching Stocks:

The below figure is the code for searching stocks. Here the user can search for stocks to see if they are available or not.

```
void medicineType::search(string key)
{
    int pos = 0, t;
    string buffer;
    buffer.erase();
    pos = search_index(key);
    if (pos == -1)
        cout << endl
             << "Medicine not found!!!" << endl;
    else if (pos >= 0)
    {
        read_from_file(pos);
        t = pos;
        while (Name_list[++t] == key && t <= count)
            read_from_file(t);
        t = pos;
        while (Name_list[--t] == key && t >= 0)
            read_from_file(t);
    }
}
```

Figure 4.6: Code for Searching Stocks

### Code for Daily Summary:

The below figure is the code for Daily Summary. Here the pharmacist will be able to view all the orders taken in a day.

```
void medicineType::search(string key)
{
    int pos = 0, t;
    string buffer;
    buffer.erase();
    pos = search_index(key);
    if (pos == -1)
        cout << endl
             << "Medicine not found!!!" << endl;
    else if (pos >= 0)
    {
        read_from_file(pos);
        t = pos;
        while (Name_list[++t] == key && t <= count)
            read_from_file(t);
        t = pos;
        while (Name_list[--t] == key && t >= 0)
            read_from_file(t);
    }
}
```

Figure 4.7: Code for Searching Stocks

# Chapter 5

# Results and Discussion

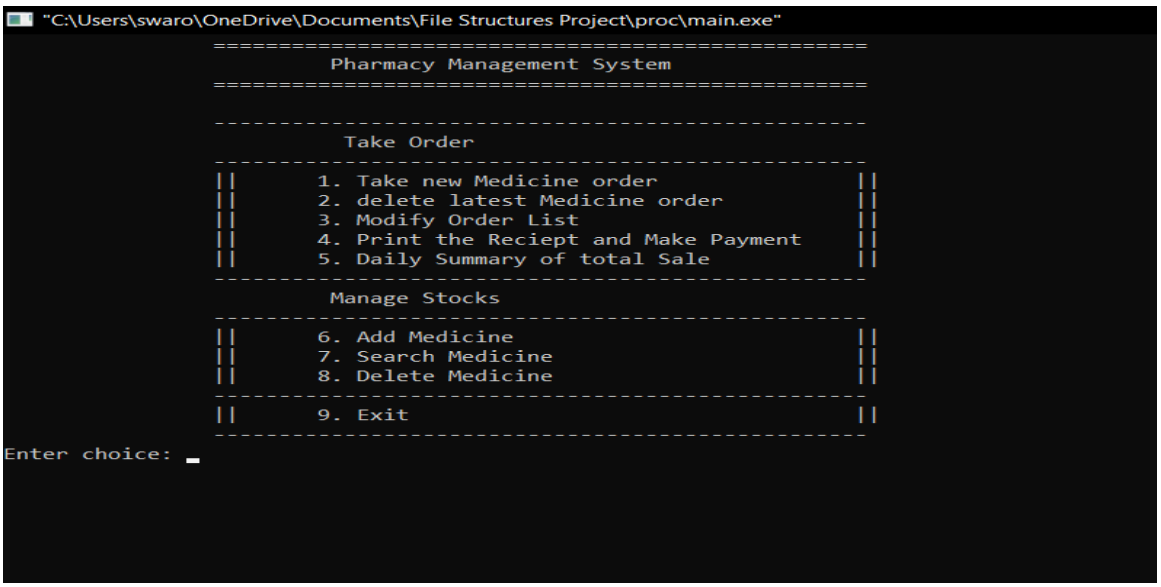


Figure 5.1: Main Menu

This is the Main Menu where the pharmacist can choose between the 9 available options according to the type of service needed.

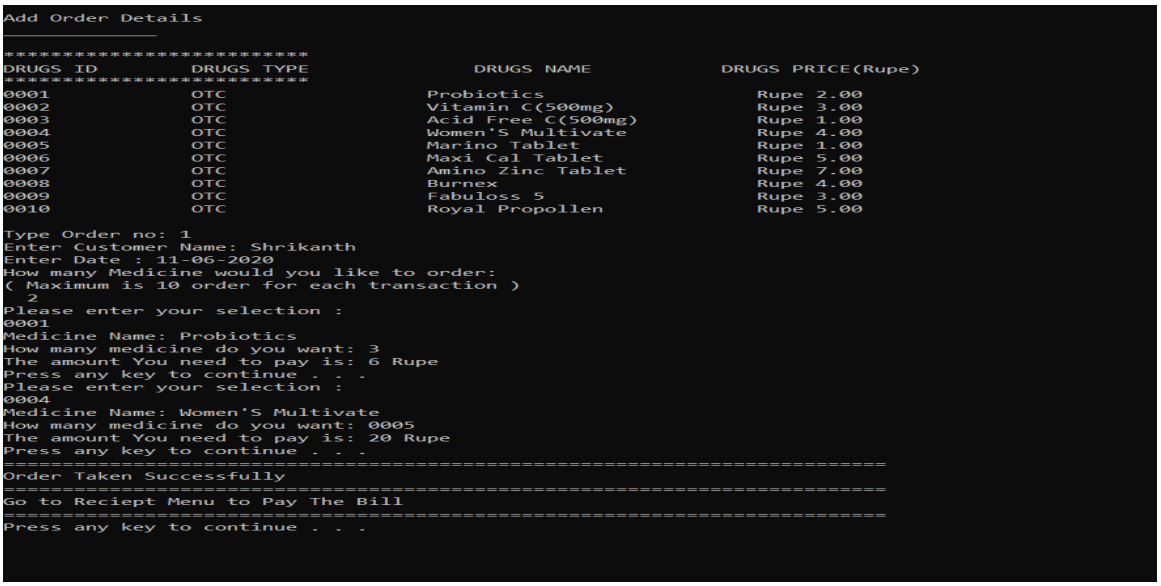


Figure 5.2: Taking Order

Here the pharmacist can order the type of medicine that is required by typing the name, quantity of the medicine.

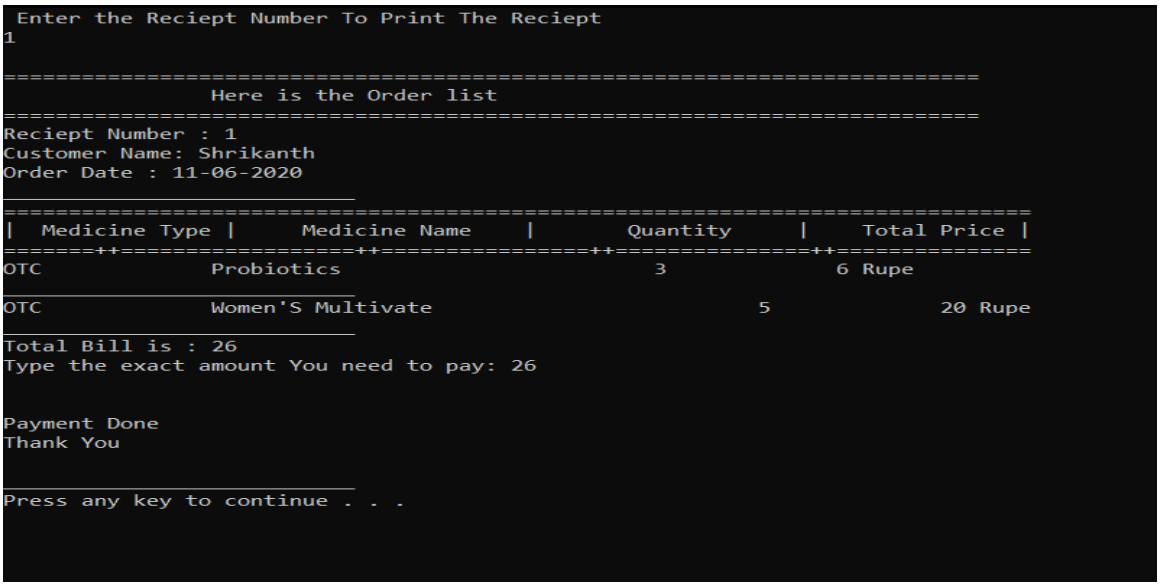


Figure 5.3: Order Receipt

Here the pharmacist will be able to view the receipt of the ordered medicine along with date of order and the customer name.

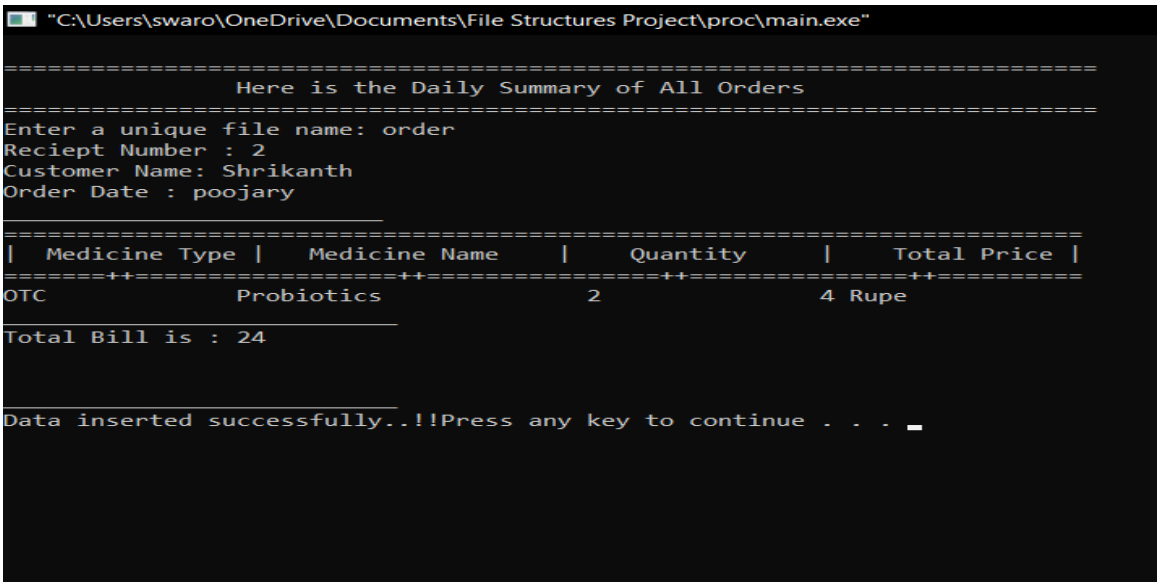


Figure 5.4: Daily Summer

The above figure is the Daily Summary page. Here the pharmacist will be able to view all the orders taken in a day.

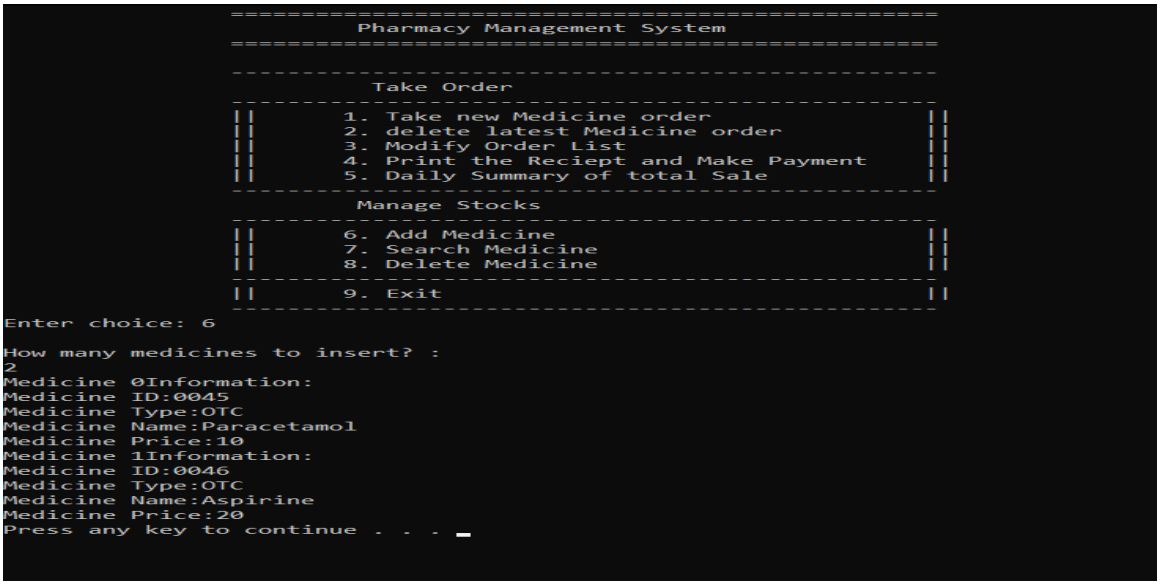


Figure 5.5: Adding Stocks

Here the pharmacist can add medicines to the existing stocks. The pharmacist should enter the type, name and price of the medicine before inserting it.

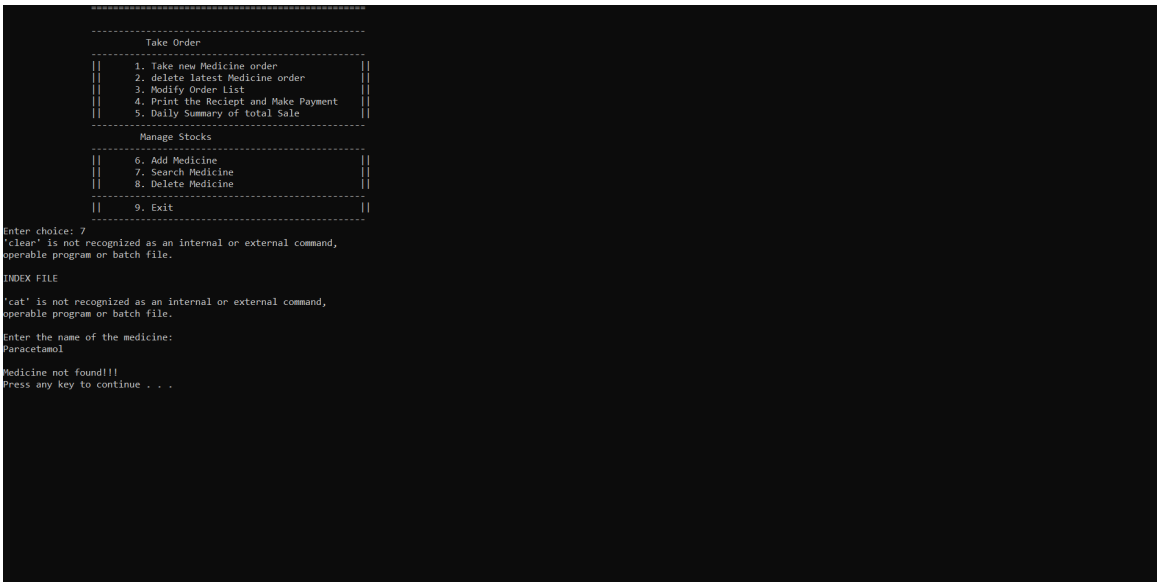


Figure 5.6: Search Medicine

Here the pharmacist can search for available medicines. If the medicine is not present then it displays medicine not found message.

# Chapter 6

# Conclusion

The "Pharmacy Management System" 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. Here we conclude that the Pharmacy Management System is developed to satisfy the complete needs of the medical stores.

# 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 Ramakrishan and Johannes Gehrke: Database Management Systems, 3rd Edition, McGraw Hill, 2003.