LOVELY
PROFESSIONAL
UNIVERSITY

Software Requirements Specification (SRS)

ORGANIZING AND MAINTAINING THE MEDICATION USE PROCESS WITHIN PHARMACIES SRS

C++

SUBMIT TO

Uminder Kaur

“In partial fulfilment of the requirements for the Master of Computer Application”.

PREPARED BY		
STUDENT’S NAME	ROLL NUMBER	REGISTRATION NO.
EKHLAKH AHMAD	RD2215B67	12209166

Lovely Professional University,
Punjab(India)

Table of Contents

1. Introduction

Computers have great relevant on storing data's securely and ease access on them in short period of time. Pharmacy management system is being build. Pharmacy management system is robust, integrated technology. Pharmacy management system deals with the maintenance of drugs and consumables in the pharmacy unit. The set-up of this pharmacy management system will ensure availability of sufficient quantity of drugs and consumable materials for the patient.

In general, The Pharmacy management system is based on computer technology that gives service for users, managed by the pharmacist who give implementation of function relatively in effective times as well as will design for removing time wasting, saving resources, easy data access of the medicine, security on data input and data access by removing almost manual based system.

1.1. Purpose

- The pharmacy management system is built for the sake of ensuring effective and clear data saving and manipulating as well as work on the pharmacy medical products.
- And almost the resources are wise used since most actions are done on the pharmacy system.
- Some of the resources minimized include paper.
- The other thing is for storing data in secure way.

1.2. Scope

With the development of specific and potent synthetic drugs, the emphasis of the pharmacist's responsibility has moved substantially towards the utilization of scientific knowledge in the proper use of modern medicines and the protection of the public against dangers that are inherent in their use.

Pharmacists are employed in regulatory control and drug management, community pharmacy, hospital pharmacy, the pharmaceutical industry, academic activities, training of other health workers, and research.

1.3. Objectives

It is the user-friendly application for Pharmacist which reduces the burden and helps to manage all sections of Pharmacy like Medicine management and Billing etc., which improve the processing efficiency. It deals with the automating tasks of maintaining of Bills. In Pharmacy, Billing management it the key process. Including safe data store about medicine as well as fast searching, delete and update of medicines. The pharmacy management system is easy for use so the user can do pharmacy actions without ambiguities.

1.4. Overview

The pharmacy management system is built in order to replace manual based system to computerize. System to be efficient, useful and affordable on implementing tasks that is order by the pharmacy manager.

2. Current System

The current pharmacy system was manually base system which is almost all works on the pharmacy organization is accomplished by papers. Among this medicine data search in order to by, audit, and other related works. And the other one is data security; the data can be accessed anyone who entered to the pharmacy house as friends other Humans without the volunteer of the pharmacist.

The pharmacists work in tedious situation because of the upper reasons. Not efficient on arrange medicine on the shelf meaning arrangement method is difficult to take in mind.

In current system almost all pharmacies do not use computerized system but use computer for giving bills only for the sold medicine to the user.

The pharmacy system will implement by the pharmacy unit of the organization. At present, system is being utilized. This system requires the pharmacist to manually monitor each drug that is available in the pharmacy shelf.

3. Proposed System

The pharmacy management system is design based on computer science students in order to illuminate the problem of the current which provided by the system involved all the pharmaceutical employees of the pharmacy.

On the other hand, PMS enables dispensation process. It stores all the physicians' prescription of the patients.

4. Functional Requirements

There are functions done by the system such as: store the necessary information of drugs, prepare bill for the medicine, give week reports, easily searching of medicine, working in two languages, Update, delete and save data of medicine.

4.1. Generate report:

The pharmacy management system generates report weakly on information about the drugs and it exports the information as output documents.

4.2. Store the necessary information of the drugs:

The pharmacy system stores the details information about each medicine including Actual name, formula of medicine and how it is importance and for which diseases is required. Since the information for each drug were required in some cases like the use of drugs, when use drugs and for whom is given.

4.3. Searching Medicine and other data:

The pharmacy has easily searched of medicine which shows in which shelf is put and the behaviour of the medicine. The searching process is based on the name of the give data or the identification of the item.

4.4. Alerting pharmacy data in the data:

Changing medicines to another because of medicine outdated, modifying the saved medicine data for incorrect data, deleting of data of the pharmacy can be done on the system.

4.5. Support multi language:

The system supposed work on primarily in English and Amharic, so the user can user one of these languages.

- ✓ Store Medicine data.
- ✓ Search Medicine data effectively.
- ✓ Update, delete, and Edit medicine information.
- ✓ Generate report on medicine.
- ✓ Prepare bill for the medicine.
- ✓ Gives navigation or information for pharmacy organization.
- ✓ Give alert for the user when medicine item is low.
- ✓ Changing login password.
- ✓ Have the system document how to use the system

5. Non-functional Requirements:

5.1. Usability:

- Which have the instruction menu how to use it which self-directive application then can be used the system without ambiguity.

5.2. Reliability:

The pharmacy system is available based on the user needs, can work properly, and transactions efficiently including safe data management of the pharmacy. The pharmacist manager control over the system by login to the pharmacy system. Any user can't use the system but the guest user can see on general properties of the pharmacy and medicines without password. As result data is protected and controlled by only the administrator.

5.3. Performance:

- The pharmacy management system operates its function in small amount of time which is less than two seconds and can be accessed by one user at a time or concurrently.
- For a service to be dependable, data servers must be accessible.

5.4. User interface:

- The user interface is friendly which is easy to use.
- Users should log in before accessing the data.
- E-mail and a phone number, if a user is required.

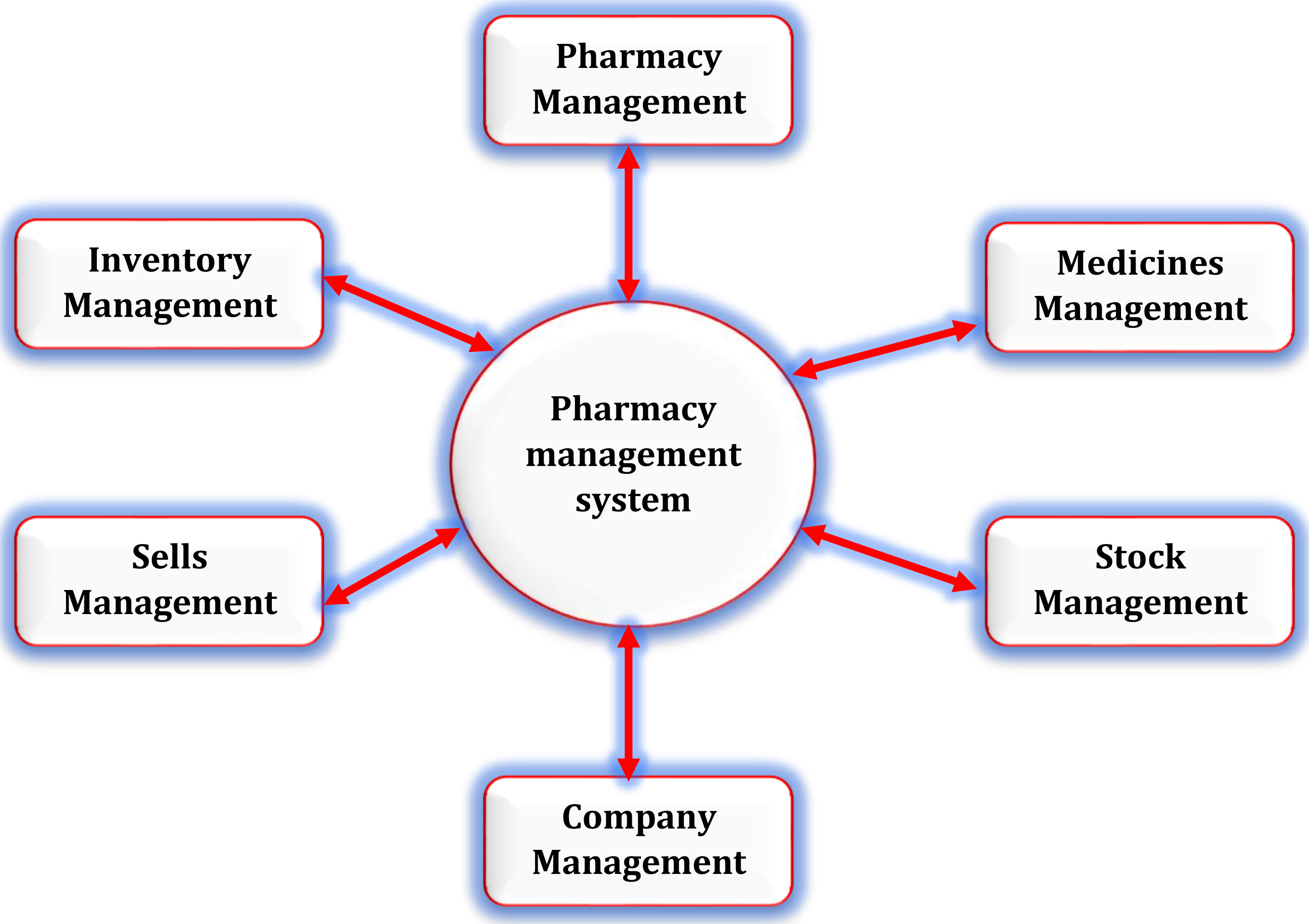
5.5. Operation:

The pharmacy management system is operated and controlled by the pharmacy manager for safe work.

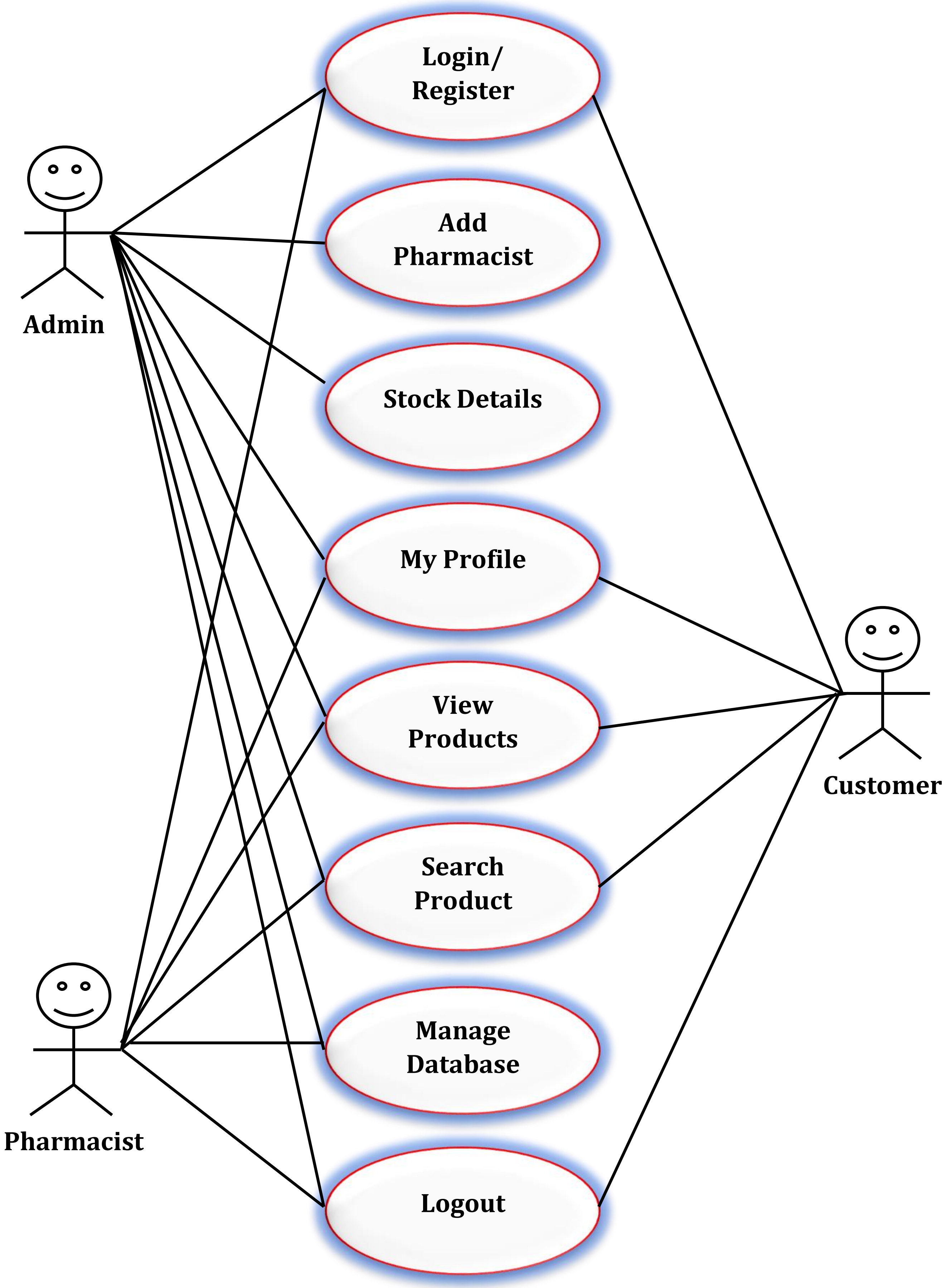
5.6. Supportability:

This pharmacy system operates in any version of windows operating system. Such as windows XP, Windows-7, 8, 10 and 11

6. Data flow diagram(DFD):



7. Use case diagram:



PROJECT

```
// ORGANIZING AND MAINTAINING THE MEDICATION USE PROCESS WITHIN PHARS..ACIES USING C++//


// Developed By:  EKHLAKH AHMAD, ANKIT KR. YADAV, AND DINANSHU YADAV  //


// LOVELY PROFESSIONAL UNIVERSITY //


//Standard Library


#include <iostream>
#include <stdlib.h>
#include <string>
#include <cctype>
#include <cmath>
#include <cstdio>
#include <fstream>
#include <iomanip>
#define max 10


using namespace std;
//the header file


class medicineType        //base class
```

```
{
public:

void take_order();//to  take_order
void delete_order(); //to delete the order
void modify(); //to modify the order
void order_list(); //to display the order_list
void daily_summary(); //to display daily_summary
void exit();  //function to exit system
medicineType();//constuctor

};

medicineType::medicineType ()
{

}                    //constructor for class CarType



struct node //constract node
{
int reciept_number;
string customerName;
string date;
int quantity[10];
string type = {"OTC"};
int x, menu2[10];
double amount[10];
```

```
string medicineName[10]={"Probiotics","Vitamin C(500mg)","Acid Free C(500mg)","Women'S Multivate","Marino Tablet","Maxi Cal Tablet",

"Amino Zinc Tablet","Burnex","Fabuloss 5","Royal Propollen"};

double Medicine[10] = {2.00,3.00,1.00,4.00,1.00,5.00,7.00,4.00,3.00,5.00};

double total;


node *prev;

node *next;

node *link;


}*q, *temp;                              //pointer declaration



node *start_ptr = NULL;

node *head = NULL;

node *last = NULL;


int main()    // Main function

{


system("COLOR 0");                 //Color to change background

medicineType medicine;

int menu;

do

{

system("cls");

cout<<"\t\t\t    PhaRs..acy Management System \n";

cout<<"\t\t=====================================================\n\n";
```

```
cout<<"\t\t--------------------------------------------------\n";
cout<<"\t\t||\t1. Take new Medicine order \t\t ||\n";
cout<<"\t\t||\t2. Delete latest Medicine order\t\t ||\n";
cout<<"\t\t||\t3. Modify Order List \t\t\t ||\n";
cout<<"\t\t||\t4. Print the Reciept and Make Payment \t ||\n";
cout<<"\t\t||\t5. Daily Summary of total Sale \t\t ||\n";
cout<<"\t\t||\t6. Exit\t\t\t\t\t ||\n";
cout<<"\t\t--------------------------------------------------\n";
cout<<"Enter choice: ";

cin>>menu;

switch (menu)
{
case 1:
{
medicine.take_order();  //function add
break;
}       //end case 1



case 2:
{
medicine.delete_order();        //function delete
system("PAUSE");
break;
}       //end case 2
```

```
case 3:
{
medicine.modify();         //function modify
system("PAUSE");
break;
}         //end case 3

case 4:
{
medicine.order_list();     //function order
system("PAUSE");
break;
}         //end case 4
case 5:
{
medicine.daily_summary();     //function daily_summary
system("PAUSE");
break;
}         //end case 5
case 6:
{
medicine.exit();     //function exit
goto a;
break;
}         //end case 6


default:
```

```
{
cout<<"You enter invalid input\nre-enter the input\n"<<endl;
break;
}//end defeault
}//end Switch
}while(menu!=6);//end do
a://goto
cout<<"thank you"<<endl;
system ("PAUSE");
return 0;
}//end  main function



void medicineType::take_order()                //function to 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(RS..)"<<endl;

cout
<<"************************************************************************************\n
";

cout<<"0001"<<"\t"<<"\tOTC"<<"\t\t"<<"   Probiotics"<<"                    RS.
2.00"<<endl;

cout<<"0002"<<"\t"<<"\tOTC"<<"\t\t"<<"   Vitamin C(500mg)"<<"
        RS. 3.00"<<endl;

cout<<"0003"<<"\t"<<"\tOTC"<<"\t\t"<<"   Acid Free C(500mg)"<<"     RS.
1.00"<<endl;

cout<<"0004"<<"\t"<<"\tOTC"<<"\t\t"<<"   Women'S Multivate"<<"
        RS. 4.00"<<endl;

cout<<"0005"<<"\t"<<"\tOTC"<<"\t\t"<<"   Marino Tablet"<<"
        RS. 1.00"<<endl;

cout<<"0006"<<"\t"<<"\tOTC"<<"\t\t"<<"   Maxi Cal Tablet"<<"             RS.
5.00"<<endl;

cout<<"0007"<<"\t"<<"\tOTC"<<"\t\t"<<"   Amino Zinc Tablet"<<"
        RS. 7.00"<<endl;

cout<<"0008"<<"\t"<<"\tOTC"<<"\t\t"<<"   Burnex"<<"
        RS. 4.00"<<endl;                              //1353fn

cout<<"0009"<<"\t"<<"\tOTC"<<"\t\t"<<"   Fabuloss 5"<<"                  RS.
3.00"<<endl;

cout<<"0010"<<"\t"<<"\tOTC"<<"\t\t"<<"   Royal Propollen"<<"             RS.
5.00"<<endl;

cout<<" "<<endl;


temp = new node;

cout << "Type Order no: ";

cin >> temp->reciept_number;

cout<< "Enter Customer Name: ";

cin>> temp->customerName;
```

```
cout<<"Enter Date : ";
cin>>temp->date;
cout << "How many Medicine would you like to order:"<< endl;
cout<<"( Maximum is 10 order for each transaction ) \n";
cout << "  " ;
cin >> temp->x;
if (temp->x >10)
{
cout << "The Medicine you order is exceed the maximum amount of order !";
system("pause");
}
else{
for (i=0; i<temp->x; i++)
{

cout << "Please enter your selection : "<<endl;
cin>> temp->menu2[i];
cout<< "Medicine Name: " <<temp->medicineName[temp->menu2[i]-1]<<endl;
cout << "How many medicine do you want: ";
cin >> temp->quantity[i];
temp->amount[i] = temp->quantity[i] * temp->Medicine[temp->menu2[i]-1];
cout << "The amount You need to pay is: " << temp->amount[i]<<" RS.."<<endl;
system("PAUSE");

}
cout<<"============================================================
==================="<<endl;
cout << "Order Taken Successfully"<<endl;
cout<<"================================================================
======================"<<endl;
cout << "Go to Reciept Menu to Pay The Bill"<<endl;
cout<<"================================================================
======================"<<endl;
system ("PAUSE");

temp->next=NULL;
if(start_ptr!=NULL)
{
temp->next=start_ptr;
}
start_ptr=temp;
system("cls");
}
}//End function take_order



void medicineType::order_list()                //Function to display receipt
{
int i, num, num2;
bool found;              //variable to search
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) //Invalid receipt code

{

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

{

temp = temp -> next;

}

if (found)     //print the receipt

{

cout <<"Reciept Number : "<<temp->reciept_number;

cout <<"\n";

cout<<"Customer Name: "<<temp->customerName<<endl;
```

```
cout<<"Order Date : "<<temp->date<<endl;


cout<<"_______________________________________________________________________"<<endl;


cout << "=============================================================================" << endl;

cout << "|  Medicine Type |    Medicine Name    |          Quantity     |   Total Price |" << endl;

cout << "=======++==================++================++==============++===============" << endl;

for (i=0;i<temp->x;i++)

{

cout << temp->type <<"  \t\t";

cout<<temp->medicineName[temp->menu2[i]-1]<<"\t\t\t  ";

cout<<temp->quantity[i] <<"\t\t";

cout<< temp->amount[i]<<" RS..."<<endl;

cout<<"___________________________________________________________________________"<<endl;

}


temp->total = temp->amount[0]+temp->amount[1]+temp->amount[2]+temp->amount[3]+temp->amount[4]+temp->amount[5]+temp->amount[6]+temp->amount[7]

+temp->amount[8]+temp->amount[9];

cout<<"Total Bill is : "<<temp->total;

cout<<"\n";

cout << "Type the exact amount You need to pay: ";

cin >> num;
```

```
cout <<"\n";

cout <<"\n";

cout<<"Payment Done\nThank You\n";

cout
<<"\n_______________________________________________________________________________\n";

}



}

}        //End function order_list



void medicineType::delete_order()   //function to delete_order

{

system("cls");

int i, num, count;

cout<<"Enter the data you want to delete \n";

cin>>num;

node *q;

node *temp;

bool found;


if(start_ptr == NULL)

cerr<<"Can not delete from an empty list.\n";

else

{

if(start_ptr->reciept_number == 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
{
found = false;
temp = start_ptr;
q = start_ptr->next;

while((!found) && (q != NULL))
{
if(q->reciept_number != num)
{
temp = q;
q = q-> next;
}
else
found = true;
}

if(found)
{
temp->next = q->next;
count--;

if(last == q)
last = temp;
delete q;
cout<<"The Reciept is Deleted Successfully"<<endl;
}
else
cout<<"Item to be deleted is not in the list."<<endl;
}
}
}      //End function delete_order



void medicineType::modify()        //function to modify order
{
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: ";
cin>> temp->customerName;
cout<<"Change Date : ";
cin>>temp->date;
cout << "How many New Medicine would you like to Change:"<< endl;
cout<<"( Maximum is 10 order for each transaction ) \n";
cout << "  " ;
cin >> temp->x;
if (temp->x >10)
{
cout << "The Medicine you order is exceed the maximum amount of order !";
```

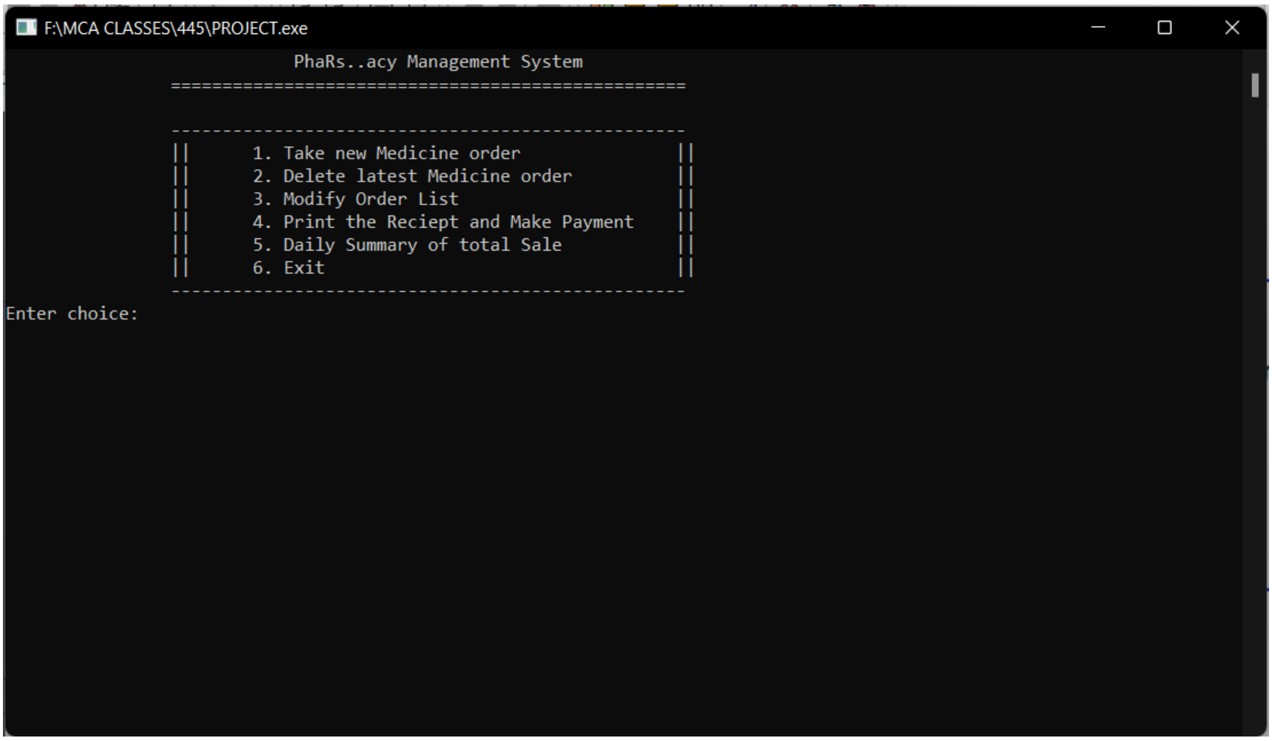
```
system("pause");
}
else{
for (i=0; i<temp->x; i++)
{

cout << "Please enter your selection to Change: "<<endl;
cin>> temp->menu2[i];
cout<< "Change Medicine Name: " <<temp->medicineName[temp->menu2[i]-1]<<endl;
cout << "How many New medicine do you want: ";
cin >> temp->quantity[i];
temp->amount[i] = temp->quantity[i] * temp->Medicine[temp->menu2[i]-1];
cout << "The amount You need to pay After Modify  is: " << temp->amount[i]<<" RS.."<<endl;
system("PAUSE");
}
temp = temp->next;
system("cls");

}

cout<<"RECORD MODIFIED....!"<<endl;
}
else
{
if(temp != NULL && temp->reciept_number != sid)
{
cout<<"Invalid Reciept Number...!"<<endl;
```

```
}
}
}
}
}                    //End modify function




void medicineType::daily_summary()                    //Function to display Daily Summary
{
int i,num;
system("cls");
node *temp ;

temp=start_ptr;


if(temp == NULL) //Invalid receipt code
{
cout << "\t\t\tThere is no Order to show\n\t\t\tSo The List is Empty\n\n\n";
}
else
{
cout<<"\n";
cout<<"==============================================================================="<<endl;
cout <<" \t\tHere is the Daily Summary of All OrdeRs. \n"; //print all receipt
```

```
cout<<"========================================================
==================="<<endl;


while(temp!=NULL)
{


cout <<"Reciept Number : "<<temp->reciept_number;
cout <<"\n";
cout<<"Customer Name: "<<temp->customerName<<endl;


cout<<"Order Date : "<<temp->date<<endl;


cout<<"________________________________________________________________________"<<en
dl;


cout <<
"============================================================
=============" << endl;
cout << "|  Medicine Type |   Medicine Name    |    Quantity     |    Total Price |" <<
endl;
cout <<
"=======++===================++================++==============
=++==========" << endl;
for (i=0;i<temp->x;i++)
{
cout << temp->type <<"  \t\t";
cout<<temp->medicineName[temp->menu2[i]-1]<<"\t\t";
cout<<temp->quantity[i] <<"\t\t";
cout<< temp->amount[i]<<" RS.."<<endl;
cout<<"_________________________________________________________________________"<<e
ndl;
}


temp->total = temp->amount[0]+temp->amount[1]+temp->amount[2]+temp->amount[3]+temp->amount[4]+temp->amount[5]+temp->amount[6]+temp->amount[7]

+temp->amount[8]+temp->amount[9];

cout<<"Total Bill is : "<<temp->total;


cout <<"\n";

cout <<"\n";

cout <<"\n________________________________________________________________________________\n";


temp=temp->next;

}

}

}                //End daily summary

void medicineType::exit() //Function to exit

{

cout<<"\nYou choose to exit.\n"<<endl;

}        //end function exit
```

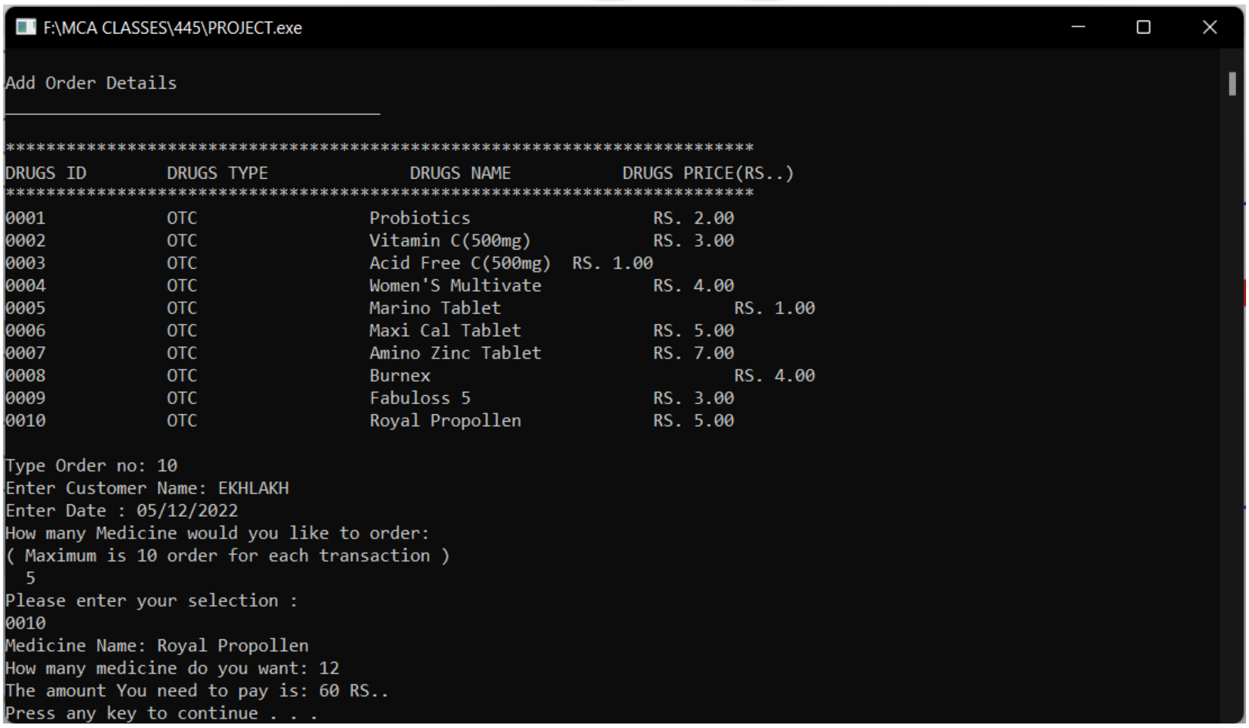
/////////////THE END OF PROGRAM///////////////

OUTPUT



```
                            PhaRs..acy Management System
                ==================================================

                --------------------------------------------------
                ||      1. Take new Medicine order               ||
                ||      2. Delete latest Medicine order          ||
                ||      3. Modify Order List                     ||
                ||      4. Print the Reciept and Make Payment    ||
                ||      5. Daily Summary of total Sale           ||
                ||      6. Exit                                  ||
                --------------------------------------------------
Enter choice:
```

EXECUTE



```
Add Order Details
_____________________________________

**************************************************************************
DRUGS ID        DRUGS TYPE              DRUGS NAME           DRUGS PRICE(RS..)
**************************************************************************
0001            OTC                 Probiotics                  RS. 2.00
0002            OTC                 Vitamin C(500mg)            RS. 3.00
0003            OTC                 Acid Free C(500mg)  RS. 1.00
0004            OTC                 Women'S Multivate           RS. 4.00
0005            OTC                 Marino Tablet                       RS. 1.00
0006            OTC                 Maxi Cal Tablet             RS. 5.00
0007            OTC                 Amino Zinc Tablet           RS. 7.00
0008            OTC                 Burnex                              RS. 4.00
0009            OTC                 Fabuloss 5                  RS. 3.00
0010            OTC                 Royal Propollen             RS. 5.00

Type Order no: 10
Enter Customer Name: EKHLAKH
Enter Date : 05/12/2022
How many Medicine would you like to order:
( Maximum is 10 order for each transaction )
  5
Please enter your selection :
0010
Medicine Name: Royal Propollen
How many medicine do you want: 12
The amount You need to pay is: 60 RS..
Press any key to continue . . .
```