MINI PROJECT REPORT

Submitted to the faculty of Engineering and Technology

B. Tech CSE

VI Semester

(Autonomous Batch)

DEPARTRMENTAL STORE BILLING SYSTEM

by

ANNE SAI VENKATA NAGA SAKETH
B16CS082

Under the Guidance of

B. HANMANTHU

Assistant Professor

Department of CSE

Department of Computer Science & Engineering

Kakatiya Institute of Technology & Science,

Warangal

2018-2019

KAKATIYA INSTITUTE OF TECHNOLOGY & SCIENCE
Warangal – 506 015, Telangana, INDIA. (An AUTONOMOUS INSTITUTE under Kakatiya University, Warangal)
కాకతీయ సాంకేతిక విజ్ఞాన శాస్త్ర విద్యాలయం, వరంగల్ – ౫౦౬ ౦౧౫.
Estd-1980
KITSW

C E R T I F I C A T E

This is to certify that **ANNE SAI VENKATA NAGA SAKETH** bearing Roll No. **B16CS082** of the VI semester B. Tech Computer Science and Engineering (Autonomous) has satisfactorily completed the mini project entitled "**DEPARTMENT STORE BILLING SYSTEM**".

Supervisor
B.HANMANTHU
Assistant Professor

Project Co-Ordinator
Dr. S. Narasimha Reddy
Associate Professor

Head of the Department
Dr. P.Niranjan

ACKNOWLEDGEMENT

I extend our sincere and heartfelt thanks to our esteemed guide, **B. HANMANTHU** sir and for his exemplary guidance, monitoring and constant encouragement throughout the course at crucial junctures and for showing us the right way.

I am grateful to respected coordinator **S.NARASIMHA REDDY** sir for permitting me to utilize all the necessary facilities of the Institute.

I would like to extend thanks to our respected Head of department, **Dr.P.NIRANJAN REDDY** sir for allowing us to use the facilities available. I would like to thank other faculty members also.

Last but not the least,I would like to thank our friends and family for the support and encouragement they have given us during the course of my work.

ANNE SAI VENKATA NAGA SAKETH
(B16CS082)

ABSTRACT

This project helps us to maintain regarding different products and it also helps us to maintain the records of the items bought and distributed for different customers. Using this project we can generate the report of the products sold in a particular period of time.

This project provides application software which is easy, comfortable and effective to carry out activities regarding stationery products for the college which is an automated system instead of making manual entries and calculation of bill and storing in books which consumes a great amount of time.

Cashier will login into his account and can modify no of items of the products. If the Products is not available then he can add that product, he can also see the Transaction details, he can also update the product details. Manager shall login to his account and can add and modify the details of cashiers and also check the customer details along with the date of purchase and the bill amount.

LIST OF FIGURES

TABLE OF CONTENTS

1. INTRODUCTION

The project is on Departmental Store Billing. Departmental Store is the place where customers come to purchase their daily necessity products and pay for them. So there is a need to calculate how many products are sold and to generate the bill for the customer. In our project we have 2 users. First is the Cashier who will store, edit, view and delete products in database and also generate bill. Second one is the Manager who will decide the discount on the products and can see the report of any product

1.1 Objective

"To make software fast in processing, with good user interface so that user can change it and it should be used for a long time without error and maintenance"

1.2 Work Flow

1. The product will come in the store.
2. Cashier will enter the information of the product in database.
3. The customer will come and take the basket with him/her and choose the product
4. The bill cashier will check the products with the ID and generate a bill
5. Customer will pay for the products.
6. All the products will be packed and delivered to the customer.

1.3 Proposed System

1. This system provides a software application to carry out the tasks related to stationery and printed materials of department of the college.
2. This system contains bills, and all the transactions associated with the departmental store so that it can be referred later when needed.
3. This application provides an easy user interface which helps the users to learn the usage of the application easily and quickly.

1.4 Scope

1. Calculate the bill.
2. Give the bill to the customer.
3. Store how many products are sold.
4. Store products and their prices and with other information.
5. Set the rates of taxes and commission on the products.
6. Can see the report of the product in a fix period of time.

2. LITERATURE REVIEW

2.1. System analysis

2.1.1. Feasibility Study

The feasibility sudy is performed to determine whether the proposed system is viable considering the Technical, Operational and Economical factors. After going through feasibility study we can have a clear-cut view of the system's benefits and drawbacks

Preliminary investigation examine project feasibility, the livelihood the system will be useful to the general stores. The main objective of the feasibility study is to test the Technical, operational and economic feasibility for adding new modules and debugging old running system. All system is feasible if they are unlimited resources and infinite time. There are aspects in the feasibility study portion of the preliminary investigation.

- Technical feasibility
- Operation feasibility
- Economic feasibility

2.1.1.1. Technical feasibility

The technical issue usually rose during the feasibility stage of the investigation includes the following;

- Does the necessary technology exist to do what is suggested?
- Do the proposed equipment's have the technical capacity to hold the data required to use the new system?
- Will the proposed system provide adequate response to inquiries, regardless of the number or location of users?
- Can the system be upgraded if developed?
- Are there technical guarantees of accuracy, reliability, ease of access and data security?

The database's purpose is to create, establish and maintain a workflow among various entities in order to facilitate all concerned users in their various capacities or roles. Permission to the users would be granted based on the roles specified. Therefore" it provides the technical guarantee of accuracy, reliability and security.

2.1.1.2. Operational Feasibility

Proposed projects are beneficial only if they can be turned out into information system. That will meet the general store's operating requirements. Operational feasibility aspects of the project are to be taken as an important part of the project implementation. Some of the important tissues raised are to test the operational feasibility of a project includes the following

- Is there sufficient support for the management from the users?
- Will the system be used and work properly if it is being developed and implemented?
- Will there be any resistance from the user that will undermine the possible application benefits?

This system is targeted to be in accordance with the above-mentioned issues. Beforehand, the management issues and user requirements have been taken into consideration. So there is no question of resistance from the users that can undermine the possible application benefits. The well-planned design would ensure the optimal utilization of the computer resources and would help in the improvement of performance status.

2.1.1.3. Economic Feasibility

As the necessary hardware and software are available in the market at a low cost, the initial investment is the only cost incurred and does not need any further enhancements. Hence it is economically feasible. The system is feasible in all respects and hence it encourages taking up the system design.

3. DESIGN

3.1 Software Requirements Specification

This section describes the intended purpose, requirements and nature of a software developed. Software requirements specification (SRS) is a description of a software system to be developed, it's defined after business requirements specification. The SRS lays out functional and non-functional requirements and may include a set of use cases that describe user interactions that the software must provide.

3.1.1 Purpose

The purpose of this document is to build an application which is used to make the billing process an easy task.

3.1.2 Requirements to Develop Java GUI Application on PC

Hardware Requirements

- Microsoft Windows 7/8/10 (32-bit or 64-bit)
- 2 GB RAM minimum, 8 GB RAM recommended
- 1 GB of available disk space minimum, 4 GB Recommended (500 MB for IDE)
- 1366 x 768 minimum screen resolution
- JDK 8

Software Requirements

- Netbeans(IDE)
- MySQL(Database)
- Java(Programming Language)

4. TECHNOLOGIES USED

4.1 Java

Java is a general-purpose computer-programming language that is concurrent, class-based, object-oriented, and specifically designed to have as few implementation dependencies as possible. It is intended to let application developers "write once, run anywhere" (WORA), meaning that compiled Java code can run on all platforms that support Java without the need for recompilation. Java applications are typically compiled to "bytecode" that can run on any Java virtual machine (JVM) regardless of the underlying computer architecture. The language derives much of its original features from Smalltalk, with a syntax similar to C and C++, but it has fewer low-level facilities than either of them. As of 2018, Java was according to Github one of the most popular programming languages in use, particularly for client-server web applications, with a reported 9 million developers.

Java was originally developed by a Canadian James Gosling at Sun Microsystems (which has since been acquired by Oracle) and released in 1995 as a core component of Sun Microsystems' Java platform. The original and reference implementation Java compilers, virtual machines, and class libraries were originally released by Sun under proprietary licenses. As of May 2007, in compliance with the specifications of the Java Community Process, Sun had relicensed most of its Java technologies under the GNU General Public License. Meanwhile, others have developed alternative implementations of these Sun technologies, such as the GNU Compiler for Java (bytecode compiler), GNU Classpath (standard libraries), and IcedTea-Web (browser plugin for applets).

The latest version is Java SE 12, released in March 2019. Since Java 9 is no longer supported, Oracle advises its users to "immediately transition" to Java 12. Oracle released the last public update for the legacy Java 8 LTS, which is free for commercial use, in January 2019. Java 8 will be supported with public updates for personal use up to at least December 2020. Oracle and others "highly recommend that you uninstall older versions of Java" because of serious risks due to unresolved security issues.

4.2 Java JVM and Bytecode

One design goal of Java is portability, which means that programs written for the Java platform must run similarly on any combination of hardware and operating system

with adequate run time support. This is achieved by compiling the Java language code to an intermediate representation called Java bytecode, instead of directly to architecture-specific machine code. Java bytecode instructions are analogous to machine code, but they are intended to be executed by a virtual machine (VM) written specifically for the host hardware. End users commonly use a Java Runtime Environment (JRE) installed on their own machine for standalone Java applications, or in a web browser for Java applets.

Standard libraries provide a generic way to access host-specific features such as graphics, threading, and networking.

The use of universal bytecode makes porting simple. However, the overhead of interpreting bytecode into machine instructions made interpreted programs almost always run more slowly than native executable. Just-in-time (JIT) compilers that compile byte-codes to machine code during runtime were introduced from an early stage. Java itself is platform-independent and is adapted to the particular platform it is to run on by a Java virtual machine for it, which translates the Java bytecode into the platform's machine language.

4.3 SQL

The name SQL stands for Structural Query Language. SQL is a data access language, like any other language, it is used for communication. SQL communicates with database manager. The database manager could be Oracle, Informix, DB2 and SQL database. SQL is easy to learn. Despite the fact that SQL is a computer programming language, it is much simpler than traditional programming language like COBOL, BASIC, FORTRAN or API. This is due to the fact that SQL is a non-procedural language.

SQL is one of the Oracle facilities. It is important to understand in each case its differences, purpose and place in the Oracle family.

- SQL is the language used to access a relational database, including Oracle.
- SQL May be used with each of the Oracle tools, where access to the database is required.

5. IMPLEMENTATION

Implementation is the stage of the project when the theoretical design is turned out into a working system. Thus it can be considered to be the most critical stage in achieving a successful new system and in giving the use, confidence that the new system will work and be effective.

The implementation stage involves careful planning, investigation of the existing system and it's constraints on implementation, designing of methods to achieve changeover and evaluation of changeover methods

This project consists of two modules, they are --

1. **Manager Module.**
   - 1.1 Manager_Add_cash
   - 1.2 Manager_edit_cash
   - 1.3 Manager_remove_cash
   - 1.4 Manager_check_customer_details
   - 1.5 Manager_change_pass
   - 1.6 Manager_view_cash
2. **Cashier Module.**
   - 2.1 Cashier_Add_ product
   - 2.2 Cashier_edit_ product
   - 2.3 Cashier_remove_product
   - 2.4 Cashier_view_product
   - 2.5 Cashier_generate_bill
   - 2.6 Cashier_check_product_report
   - 2.7 Cashier_change_pass

MANAGER MODULE

1.1 MANAGER_ADD_CASHIER

```
public class Man_add_cash extends javax.swing.JFrame {
 public Man_add_cash() {
    initComponents();
 }

 private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    String name,email,password,phone,id,address;
    int sal;
    id = jTextField1.getText();
    name = jTextField2.getText();
    email = jTextField3.getText();
    password = jTextField4.getText();
    sal = Integer.parseInt(jTextField5.getText());
    address = jTextField6.getText();
    phone = jTextField7.getText();
    AddCashier(id,name,email,password,sal,address,phone);
 }

 private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
     Manager_main mm = new Manager_main();
    mm.setVisible(true);
 }
private void AddCashier(String id,String name,String email,String password,int sal,String
address,String phone)
{
 try{
    Connection con=MY_DB.database.getConnection();
    PreparedStatement pstmt=null;
    pstmt = con.prepareStatement("select * from cashier where p_id=?");
    pstmt.setString(1,id);
    ResultSet rs = pstmt.executeQuery();
    if(rs.next())
```

```
{
    JOptionPane.showMessageDialog(this,"Cashier Details With the ID Already
Exixt, Please Edit If Needed");
}
else
{
pstmt = con.prepareStatement("insert into cashier values (?,?,?,?,?,?,?)");
pstmt.setString(1,id);
pstmt.setString(2,name);
pstmt.setString(3,email);
pstmt.setString(4,password);
pstmt.setInt(5,sal);
pstmt.setString(6,address);
pstmt.setString(7,phone);
int n = pstmt.executeUpdate();
if(n!=0)
{
    JOptionPane.showMessageDialog(this,"Cashier Details Added Successfully");
    jTextField1.setText("");
    jTextField2.setText("");
    jTextField3.setText("");
    jTextField4.setText("");
    jTextField5.setText("");
    jTextField6.setText("");
    jTextField7.setText("");
}
else
{
    JOptionPane.showMessageDialog(this,"Cashier Details Not Added");
}
}
}
catch(Exception e)
{
```

```
            System.out.println("Exception raised:"+e);
        }
    }
}
```

1.2 MANAGER_EDIT_CASHIER

```
public class Man_edit_cash extends javax.swing.JFrame {
    public Man_edit_cash() {
        initComponents();
    }
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        String cname,cemail,cpassword,cadd,cphone,cid;
        int csal;
        cname = jTextField1.getText();
        cemail = jTextField2.getText();
        cpassword = jTextField3.getText();
        cadd = jTextField5.getText();
        cphone = jTextField6.getText();
        cid = jTextField7.getText();
        csal = Integer.parseInt(jTextField4.getText());
        UpdateDetails(cid,cname,cemail,cpassword,cadd,cphone,csal);
    }
 private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
          String c_id = jTextField7.getText();
        getdetails(c_id);
    }
 private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
        Manager_main mm = new Manager_main();
        mm.setVisible(true);
    }
private void UpdateDetails(String id,String name,String email,String password,String
add,String phone,int sal)
{
    try{
        Connection con=MY_DB.database.getConnection();
```

```
        PreparedStatement pstmt=con.prepareStatement("update cashier set
p_name=?,p_email=?,p_pass=?,p_sal=?,p_add=?,p_phone=? where p_id=?");
        pstmt.setString(1,name);
        pstmt.setString(2,email);
        pstmt.setString(3,password);
        pstmt.setInt(4,sal);
        pstmt.setString(5,add);
        pstmt.setString(6,phone);
        pstmt.setString(7,id);
        int n = pstmt.executeUpdate();
        if(n!=0)
        {
           JOptionPane.showMessageDialog(this,"Cashier Details Updated Successfully");
           jTextField1.setText("");
           jTextField2.setText("");
           jTextField3.setText("");
           jTextField4.setText("");
           jTextField5.setText("");
           jTextField6.setText("");
           jTextField7.setText("");
        }
        else
        {
           JOptionPane.showMessageDialog(this,"Cashier Details Not Updated");
        }
        }
        catch(Exception e)
        {
           System.out.println("Exception raised:"+e);
        }
    }
private void getdetails(String id)
{
    String cname,cemail,cpassword,cadd,cphone,cid;
```

```
        int csal;
    try{
        Connection con=MY_DB.database.getConnection();
        PreparedStatement pstmt=con.prepareStatement("select * from cashier where
p_id=?");
        pstmt.setString(1,id);
        ResultSet rs=pstmt.executeQuery();
        if(rs.next())
        {
            JOptionPane.showMessageDialog(this,"Cashier Details Retrieved Successfully");
        cname = rs.getString(2);
        cemail = rs.getString(3);
        cpassword = rs.getString(4);
        csal = Integer.parseInt(rs.getString(5));
        cadd = rs.getString(6);
        cphone = rs.getString(7);
        String c_sal = Integer.toString(csal);
        jTextField1.setText(cname);
        jTextField2.setText(cemail);
        jTextField3.setText(cpassword);
        jTextField5.setText(cadd);
        jTextField6.setText(cphone);
        jTextField4.setText(c_sal);
        }
        else
        {
            JOptionPane.showMessageDialog(this,"Cashier Details Not Retrieved");
        }
        }
        catch(Exception e)
        {
            System.out.println("Exception raised:"+e);
        }
    }
```

```
}
```

1.3 MANAGER_REMOVE_CASHIER

```
public class Man_rem_cash extends javax.swing.JFrame {
    public Man_rem_cash() {
        initComponents();
    }
    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        jtf_name.setText("");
        jtf_pid.setText("");
    }
    private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
        String name=jtf_name.getText();
        String pid=jtf_pid.getText();
        removecash(name,pid);
    }
    private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
        Manager_main mm = new Manager_main();
        mm.setVisible(true);
    }
    private void removecash(String name, String pid) {
        try
        {
            Connection con=MY_DB.database.getConnection();
            PreparedStatement pstmt=null;
            if((name!=null)&&(pid.isEmpty()))
            {
                pstmt=con.prepareStatement("delete from cashier where p_name=?");
                pstmt.setString(1,name);
                int x=pstmt.executeUpdate();
                if(x!=0)
                {
                    JOptionPane.showMessageDialog(this,"Cashier Access Removed
Successfully");
```

```
            }
            else
            {
               JOptionPane.showMessageDialog(this,"Please try again with a valid Name or
ID");
            }
         }
         else if((pid!=null)&&(name.isEmpty()))
         {
            pstmt=con.prepareStatement("delete from cashier where p_id=?");
            pstmt.setString(1,pid);
            int x=pstmt.executeUpdate();
            if(x!=0)
            {
               JOptionPane.showMessageDialog(this,"Cashier Access Removed
Successfully");
            }
            else
            {
               JOptionPane.showMessageDialog(this,"Please try again with a valid Name or
ID");
            }
         }
         else
         {
            JOptionPane.showMessageDialog(this,"Please enter the details and try again");
         }
      }
      catch(Exception e)
      {
         System.out.println("Exception raised : "+e);
      }
   } }
```

1.4 MANAGER_CHECK_CUSTOMER_DETAILS

```
public class Man_check_cust_det extends javax.swing.JFrame {
    int totalbill = 0;
    public Man_check_cust_det() {
        initComponents();
    }
    @SuppressWarnings("unchecked")
    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        int year,month,date;
        year = Integer.parseInt(jTextField1.getText());
        month = Integer.parseInt(jTextField2.getText());
        date = Integer.parseInt(jTextField3.getText());
        getcustdetails(year,month,date);
    }
    private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
        Manager_main mm = new Manager_main();
        mm.setVisible(true);
    }
    private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
        int year,month,date;
        year = Integer.parseInt(jTextField1.getText());
        month = Integer.parseInt(jTextField2.getText());
        date = Integer.parseInt(jTextField3.getText());
        deleteallcustdetails(year,month,date);
    }
    private void jRadioButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        jTextField1.disable();
        jTextField2.disable();
        jTextField3.disable();
        getalldetails();
    }
    private void jRadioButton2ActionPerformed(java.awt.event.ActionEvent evt) {
        jTextField1.enable();
```

```
        jTextField2.enable();
        jTextField3.enable();
    }
    private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {
        JOptionPane.showMessageDialog(this,"Total Revenue generated for the peroid is::"+totalbill);
    }
    private void getcustdetails(int year, int month, int date) {
         totalbill = 0;
         try{
             DefaultTableModel model1 = (DefaultTableModel)jTable1.getModel();
             model1.setRowCount(0);
        Connection con=MY_DB.database.getConnection();
        PreparedStatement pstmt=null;
        pstmt = con.prepareStatement("select * from customer where year=? and month=? and date=?");
        pstmt.setInt(1,year);
        pstmt.setInt(2,month);
        pstmt.setInt(3,date);
        ResultSet rs = pstmt.executeQuery();
        String cust_name,cust_phone,cust_bill;
        boolean flag = false;
        while(rs.next())
        {
            flag = true;
            cust_name = rs.getString(1);
            cust_phone= rs.getString(2);
            cust_bill = rs.getString(3);
            int custbill = Integer.parseInt(cust_bill);
            totalbill = totalbill + custbill;
            model1.insertRow(model1.getRowCount(),newObject[] {cust_name,cust_phone,cust_bill,year,month,date});
        }
        if(flag!=true)
```

```
        {
            JOptionPane.showMessageDialog(this,"Customer Details Do Not Exist");
        }
        }
        catch(Exception e)
        {
            System.out.println("Exception raised:"+e);
        }
    }
    private void deleteallcustdetails(int year, int month, int date) {
        try{
        Connection con=MY_DB.database.getConnection();
        PreparedStatement pstmt=null;
        pstmt = con.prepareStatement("delete from customer where year=? and month=? and
date=?");
        pstmt.setInt(1,year);
        pstmt.setInt(2,month);
        pstmt.setInt(3,date);
        int n = pstmt.executeUpdate();
        if(n!=0)
        {
            JOptionPane.showMessageDialog(this,"All Customer Details Deleted on the
Respective Day");
        }
        else
        {
            JOptionPane.showMessageDialog(this,"Customer Details Not Deleted");
        }
        }
        catch(Exception e)
        {
            System.out.println("Exception raised:"+e);
        }
    }
```

```
    private void getalldetails() {
        totalbill = 0;
        try{
            DefaultTableModel model1 = (DefaultTableModel)jTable1.getModel();
            model1.setRowCount(0);
        Connection con=MY_DB.database.getConnection();
        PreparedStatement pstmt=null;
        pstmt = con.prepareStatement("select * from customer");
        ResultSet rs = pstmt.executeQuery();
        String cust_name,cust_phone,cust_bill;
        int year,month,date;
        boolean flag = false;
        while(rs.next())
        {
            flag = true;
            cust_name = rs.getString(1);
            cust_phone= rs.getString(2);
            cust_bill = rs.getString(3);
            year = rs.getInt(4);
            month = rs.getInt(5);
            date = rs.getInt(6);
            int custbill = Integer.parseInt(cust_bill);
            totalbill = totalbill + custbill;
            model1.insertRow(model1.getRowCount(),newObject[]
{cust_name,cust_phone,cust_bill,year,month,date});
        }
        if(flag!=true)
        {
            JOptionPane.showMessageDialog(this,"Customer Details Do Not Exist");
        }
        }
        catch(Exception e)
        {
            System.out.println("Exception raised:"+e);        }
```

```
    }
}
```

1.5 MANAGER_CHANGE_PASSWORD

```
public class Man_change_pass extends javax.swing.JFrame {
    public Man_change_pass() {
        initComponents();
    }
String userid;
    public Man_change_pass(String id) {
         initComponents();
         userid = id;
    }
    @SuppressWarnings("unchecked")
    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        String oldpass,newpass,cnfnewpass;
        oldpass = jTextField1.getText();
        newpass = jTextField2.getText();
        cnfnewpass = jTextField3.getText();
        if(newpass.equalsIgnoreCase(cnfnewpass))
        {
            getuserdet(userid,oldpass,newpass,cnfnewpass);
        }
        else
        {
            JOptionPane.showMessageDialog(this,"New Passwords are not Same, Please try
Again");
        }
    }
    private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
        Manager_main mm = new Manager_main();
        mm.setVisible(true);
    }
```

```
    private void getuserdet(String userid,String oldpass, String newpass, String cnfnewpass)
{
        try{
       Connection con=MY_DB.database.getConnection();
       PreparedStatement pstmt = null;
       pstmt = con.prepareStatement("select * from managers where m_email=?");
       pstmt.setString(1,userid);
       ResultSet rs=pstmt.executeQuery();
       if(rs.next())
       {
          String ooldpass = rs.getString("m_pass");
          if(ooldpass.equals(oldpass))
          {
          pstmt = con.prepareStatement("update managers set m_pass=? where
m_email=?");
          pstmt.setString(1, newpass);
          pstmt.setString(2, userid);
          int n = pstmt.executeUpdate();
          if(n!=0)
          {
             JOptionPane.showMessageDialog(this,"Password Updated");
          }
          else
          {
             JOptionPane.showMessageDialog(this,"Password not updated");
          }
          }
       }
       else
       {
          JOptionPane.showMessageDialog(this,"Please Enter correct Old Password");
       }
       }
       catch(Exception e)
```

```
        {
            System.out.println("Exception raised:"+e);
        }
    }
}
```

1.6 MANAGER_VIEW_CASHIER

```
public class Manager_view_cash extends javax.swing.JFrame {
    public Manager_view_cash() {
        initComponents();
    }
    @SuppressWarnings("unchecked")
    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        getdetails();
    }
    private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {
        Manager_main mm = new Manager_main();
        mm.setVisible(true);
    }
private void getdetails()
{
     String cname,cemail,cadd,cphone,cid;
        int csal;
        DefaultTableModel model1 = (DefaultTableModel)jTable1.getModel();
            model1.setRowCount(0);
    try{
        Connection con=MY_DB.database.getConnection();
        PreparedStatement pstmt=con.prepareStatement("select * from cashier");
        ResultSet rs=pstmt.executeQuery();
        while(rs.next())
        {
        cid = rs.getString(1);
        cname = rs.getString(2);
        cemail = rs.getString(3);
```

```
        csal = Integer.parseInt(rs.getString(5));
        cadd = rs.getString(6);
        cphone = rs.getString(7);
        String c_sal = Integer.toString(csal);
         model1.insertRow(model1.getRowCount(),newObject[]
{cid,cname,cemail,c_sal,cadd,cphone});
        }
        }
        catch(Exception e)
        {
           System.out.println("Exception raised:"+e);
        }
    }
}
```

CASHIER MODULE

2.1 CASHIER_ADD_PRODUCT

```
public class Manager_view_cash extends javax.swing.JFrame {
    public Manager_view_cash() {
        initComponents();
    }
    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        getdetails();
    }
    private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {
        Manager_main mm = new Manager_main();
        mm.setVisible(true);
    }
private void getdetails()
{
     String cname,cemail,cadd,cphone,cid;
        int csal;
        DefaultTableModel model1 = (DefaultTableModel)jTable1.getModel();
            model1.setRowCount(0);
    try{
        Connection con=MY_DB.database.getConnection();
        PreparedStatement pstmt=con.prepareStatement("select * from cashier");
        ResultSet rs=pstmt.executeQuery();
        while(rs.next())
        {
        cid = rs.getString(1);
        cname = rs.getString(2);
        cemail = rs.getString(3);
        csal = Integer.parseInt(rs.getString(5));
        cadd = rs.getString(6);
        cphone = rs.getString(7);
        String c_sal = Integer.toString(csal);
         model1.insertRow(model1.getRowCount(),new Object[]
{cid,cname,cemail,c_sal,cadd,cphone});
```

```
        }
        }
        catch(Exception e)
        {
            System.out.println("Exception raised:"+e);
        }
    }
}
```

2.2 CASHIER_EDIT_PRODUCT

```
public class Cashier_edit_prod extends javax.swing.JFrame {
    public Cashier_edit_prod() {
        initComponents();
    }
    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
      String prod_id = jTextField1.getText();
        getdetails(prod_id);
    }
    private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
        String prodname,prodshort,proddealer,prodexp,prodid;
        int prodcp,prodsp,proddisc,prodquan;
        prodid = jTextField1.getText();
        prodname = jTextField2.getText();
        prodshort = jTextField3.getText();
        proddealer = jTextField4.getText();
        prodexp = jTextField5.getText();
        prodcp = Integer.parseInt(jTextField6.getText());
        prodsp = Integer.parseInt(jTextField7.getText());
        proddisc = Integer.parseInt(jTextField8.getText());
        prodquan = Integer.parseInt(jTextField9.getText());

UpdateDetails(prodname,prodshort,proddealer,prodexp,prodid,prodcp,prodsp,proddisc,prodq
uan);
    }
```

```
    private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
        Cashier_main cm =new Cashier_main();
        cm.setVisible(true);
    }
private void getdetails(String id)
{
     String prodname,prodshort,proddealer,prodexp;
        int prodcp,prodsp,proddisc,prodquan;
    try{
        Connection con=MY_DB.database.getConnection();
        PreparedStatement pstmt=con.prepareStatement("select * from products where prod_id=?");
        pstmt.setString(1,id);
        ResultSet rs=pstmt.executeQuery();
        if(rs.next())
        {
            JOptionPane.showMessageDialog(this,"Product Details Retrieved Successfully");
        prodname = rs.getString(2);
        prodshort = rs.getString(3);
        proddealer = rs.getString(4);
        prodexp = rs.getString(9);
        prodcp = Integer.parseInt(rs.getString(5));
        prodsp = Integer.parseInt(rs.getString(6));
        proddisc = Integer.parseInt(rs.getString(7));
        prodquan = Integer.parseInt(rs.getString(8));
        String prod_cp = Integer.toString(prodcp);
        String prod_sp = Integer.toString(prodsp);
        String prod_disc = Integer.toString(proddisc);
        String prod_quan = Integer.toString(prodquan);
        jTextField2.setText(prodname);
        jTextField3.setText(prodshort);
        jTextField4.setText(proddealer);
        jTextField5.setText(prodexp);
        jTextField6.setText(prod_cp);
```

```
        jTextField7.setText(prod_sp);
        jTextField8.setText(prod_disc);
        jTextField9.setText(prod_quan);
        }
        else
        {
           JOptionPane.showMessageDialog(this,"Product Details Not Retrieved");
        }
        }
        catch(Exception e)
        {
           System.out.println("Exception raised:"+e);
        }
    }
private void UpdateDetails(String prodname,String prodshort,String proddealer,String prodexp,String prodid,int prodcp,int prodsp,int proddisc,int prodquan)
{
    try{
        Connection con=MY_DB.database.getConnection();
        PreparedStatement.pstmt=con.prepareStatement("update products set prod_name=?,prod_shortcut=?,prod_dealer=?,prod_cp=?,prod_sp=?,prod_disc=?,prod_quantity=?,prod_exp=? where prod_id=?");
        pstmt.setString(1,prodname);
        pstmt.setString(2,prodshort);
        pstmt.setString(3,proddealer);
        pstmt.setInt(4,prodcp);
        pstmt.setInt(5,prodsp);
        pstmt.setInt(6,proddisc);
        pstmt.setInt(7,prodquan);
        pstmt.setString(8,prodexp);
        pstmt.setString(9,prodid);
        int n = pstmt.executeUpdate();
        if(n!=0)
        {
```

```
            JOptionPane.showMessageDialog(this,"Product Details Updated Successfully");
            jTextField1.setText("");
            jTextField2.setText("");
            jTextField3.setText("");
            jTextField4.setText("");
            jTextField5.setText("");
            jTextField6.setText("");
            jTextField7.setText("");
            jTextField8.setText("");
            jTextField9.setText("");
        }
        else
        {
            JOptionPane.showMessageDialog(this,"Product Details Not Updated");
        }
        }
        catch(Exception e)
        {
            System.out.println("Exception raised:"+e);
        }
    }
}
```

2.3 CASHIER_REMOVE_PRODUCT

```
        public class Cashier_rem_prod extends javax.swing.JFrame {
            public Cashier_rem_prod() {
                initComponents();
            }
            private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
                String prodid = jTextField1.getText();
                removeprod(prodid);
            }
            private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
                Cashier_main cm =new Cashier_main();
```

```
        cm.setVisible(true);
    }
private void removeprod(String prodid)
{
   try
      {
         Connection con=MY_DB.database.getConnection();
         PreparedStatement pstmt=null;
         if(prodid!=null)
         {
            pstmt=con.prepareStatement("delete from products where prod_id=?");
            pstmt.setString(1,prodid);
            int x=pstmt.executeUpdate();
            if(x>0)
            {
               JOptionPane.showMessageDialog(this,"Product Removed Successfully
from Database");
            }
            else
            {
               JOptionPane.showMessageDialog(this,"Please try again with a valid
Product ID");
            }
         }
      }
      catch(Exception e)
      {
         System.out.println("Exception raised : "+e);
      }
}
}
```

2.4 CASHIER_VIEW_PRODUCT

```
public class Cash_view_prod extends javax.swing.JFrame {
```

```
    public Cash_view_prod() {
        initComponents();
    }
    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        String s = null;
        if(jRadioButton1.isSelected())
        {
            s = "highest";
        }
        else if(jRadioButton2.isSelected())
        {
            s = "lowest";
        }
        getdetails(s);
    }
    private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
        Cashier_main cm = new Cashier_main();
        cm.setVisible(true);
    }
    private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
        String pid = jTextField1.getText();
        int stock = Integer.parseInt(jTextField2.getText());
        Addstock(pid,stock);
    }
private void getdetails(String orderby)
{
     String pname,pid,pshort,pdeal;
        int pcp,psp,pd,pquant;
        DefaultTableModel model1 = (DefaultTableModel)jTable1.getModel();
            model1.setRowCount(0);
    try{
        Connection con=MY_DB.database.getConnection();
        PreparedStatement pstmt = null;
        if(orderby.equalsIgnoreCase("highest"))
```

```
        {
           pstmt=con.prepareStatement("select * from products order by prod_quantity
desc");
        }
        else if(orderby.equalsIgnoreCase("lowest"))
        {
           pstmt=con.prepareStatement("select * from products order by prod_quantity
asc");
        }
        ResultSet rs=pstmt.executeQuery();
        while(rs.next())
        {
        pid = rs.getString(1);
        pname = rs.getString(2);
        pshort = rs.getString(3);
        pdeal = rs.getString(4);
        pcp = Integer.parseInt(rs.getString(5));
        psp = Integer.parseInt(rs.getString(6));
        pd = Integer.parseInt(rs.getString(7));
        pquant = Integer.parseInt(rs.getString(8));
        String p_cp = Integer.toString(pcp);
        String p_sp = Integer.toString(psp);
        String p_d = Integer.toString(pd);
        String p_quant = Integer.toString(pquant);
         model1.insertRow(model1.getRowCount(),new                    Object[]
{pid,pname,pshort,pdeal,p_cp,p_sp,p_d,p_quant});
        }
        }
        catch(Exception e)
        {
           System.out.println("Exception raised:"+e);
        }
    }
    private void Addstock(String pid, int stock) {
```

```
 try{
Connection con=MY_DB.database.getConnection();
PreparedStatement pstmt = null;
pstmt = con.prepareStatement("select * from products where prod_id=?");
pstmt.setString(1, pid);
ResultSet rs=pstmt.executeQuery();
int instock = 0;
if(rs.next())
{
instock = rs.getInt("prod_quantity");
JOptionPane.showMessageDialog(this,"Product In Stock::"+instock+" items");
int newquant = instock + stock;
pstmt=con.prepareStatement("update products set prod_quantity=? where prod_id=?");
pstmt.setInt(1,newquant);
pstmt.setString(2,pid);
int r=pstmt.executeUpdate();
if(r!=0)
{
}
else
{
   JOptionPane.showMessageDialog(this,"Product Quantity Not updated");
}
}
else
{
   JOptionPane.showMessageDialog(this,"Please Enter Valid Details");
}
}
catch(Exception e)
{
   System.out.println("Exception raised:"+e);
}
```

```
    }
}
```

2.5 CASHIER_GENERATE_BILL

```
public class Cashier_gen_bill extends javax.swing.JFrame {
    int i = 0;
    int billcount = 0;
    double totalbill=0,totalquantity=0;
    String custname = "";
    public Cashier_gen_bill() {
        initComponents();
    }
    private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
        String prodid,prodshort;
        String custname,custphone;
        prodshort = jTextField1.getText();
        prodid = jTextField2.getText();
        custname = jTextField4.getText();
        custphone = jTextField5.getText();
        int quan = Integer.parseInt(jTextField3.getText());
        Date tod = new Date();
        int mo,year,day;
        mo = tod.getMonth()+1;
        year = tod.getYear()+1900;
        day = tod.getDate();
        getdetails(prodid,prodshort,quan,mo,year,day,custname,custphone);
    }
    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        Cashier_main cm = new Cashier_main();
        cm.setVisible(true);
    }
    private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
        displaytemp(totalbill,totalquantity,custname);
    }
```

```
private void getdetails(String prodid,String prodshort,int quan,int mo,int year,int day,
String custname, String custphone)
{
    try
      {
        DefaultTableModel model1 = (DefaultTableModel)jTable1.getModel();
        Connection con=MY_DB.database.getConnection();
        PreparedStatement pstmt=null;
        int quant,latestquan;
        double cost,discount;
        int custbill = 0;
        boolean custdetflag = false;
        boolean resflag = false;
        if((prodid!=null)&&(prodshort.isEmpty()))
        {
           pstmt=con.prepareStatement("select * from products where prod_id=?");
           pstmt.setString(1,prodid);
           ResultSet rs = pstmt.executeQuery();
           while(rs.next())
           {
              resflag = true;
              getreportdet(prodid,prodshort,year,mo,quan);
              String productname = rs.getString(2);
              quant = rs.getInt("prod_quantity");
              if(quant < 10)
              {
                 JOptionPane.showMessageDialog(this,"Product Quantity Less in
Stock, Please Add Stock");
              }
              cost = rs.getDouble("prod_sp");
              discount = rs.getDouble("prod_disc");
              if(quant>quan)
              {
                 latestquan = quant - quan;
```

```
                double bill = calculatecost(cost,discount,quan);
                String singleprodbill = Double.toString(bill);
                totalbill = totalbill + bill;
                totalquantity = totalquantity + quan;
                custbill = (int)totalbill;
                model1.insertRow(model1.getRowCount(),newObject[]
{productname,quan,discount,singleprodbill});

            if(updatequant(latestquan,prodshort,prodid))
            {
                storecustdet(custname,custphone,custbill,year,mo,day);
                if(billcount == 0)
                {
                    String cname = jTextField4.getText();
                    this.custname = cname;
                    addtofile(productname,quan,discount,singleprodbill,cname);
                    billcount = billcount+1;
                }
                else if(billcount > 0)
                {
                    this.custname = custname;
                    addtooldfile(productname,quan,discount,singleprodbill,custname);
                }
            }
            }
            else
            {
                JOptionPane.showMessageDialog(this,"Required Quantity Not in
Stock");
            }
        }
        if(resflag!=true)
        {
```

```
            JOptionPane.showMessageDialog(this,"Please Try Again With a Valid
Product ID");
          }
        }
        else if((prodshort!=null)&&(prodid.isEmpty()))
        {
          pstmt=con.prepareStatement("select * from products where
prod_shortcut=?");
          pstmt.setString(1,prodshort);
          ResultSet rs = pstmt.executeQuery();
          while(rs.next())
          {
            resflag = true;
            getreportdet(prodid,prodshort,year,mo,quan);
            quant = rs.getInt("prod_quantity");
            if(quant < 10)
            {
              JOptionPane.showMessageDialog(this,"Product Quantity Less in
Stock, Please Add Stock");
            }
            cost = rs.getDouble("prod_sp");
            discount = rs.getDouble("prod_disc");
            String productname = rs.getString(2);
            if(quant>quan)
            {
            latestquan = quant - quan;
            double bill = calculatecost(cost,discount,quan);
            String singleprodbill = Double.toString(bill);
            totalbill = totalbill + bill;
            totalquantity = totalquantity + quan;
            custbill = (int)totalbill;
            model1.insertRow(model1.getRowCount(),new Object[]
{productname,quan,discount,singleprodbill});
            if(updatequant(latestquan,prodshort,prodid))
```

```
                {
                    storecustdet(custname,custphone,custbill,year,mo,day);
                    if(billcount == 0)
                    {
                        String cname = jTextField4.getText();
                        this.custname = cname;
                        addtofile(productname,quan,discount,singleprodbill,cname);
                        billcount = billcount+1;
                    }
                    else if(billcount > 0)
                    {
                        this.custname = custname;
                        addtooldfile(productname,quan,discount,singleprodbill,custname);
                    }
                }
                }
                else
                {
                    JOptionPane.showMessageDialog(this,"Required Quantity Not in Stock");
                }
            }
            if(resflag!=true)
            {
                JOptionPane.showMessageDialog(this,"Please Try Again With a Valid Product ShortCut");
            }
        }
    }
    catch(Exception e)
    {
        System.out.println("Exception raised : "+e);
    }
}
```

```
    private void storecustdet(String custname, String custphone, int custbill,int year,int mo,int day) {
        generatecustid();
        String ival = Integer.toString(i);
        try{
           if(custname.isEmpty())
           {
              custname = "cust_name"+ival;
              jTextField4.setText(custname);
           }
           if(custphone.isEmpty())
           {
              custphone = "0000000000";
              jTextField5.setText(custphone);
           }
        Connection con=MY_DB.database.getConnection();
        PreparedStatement pstmt=null;
        pstmt = con.prepareStatement("select * from customer where year=? and month=? and date=? and cust_name=? and cust_phone=?");
        pstmt.setString(4, custname);
        pstmt.setString(5, custphone);
        pstmt.setInt(1, year);
        pstmt.setInt(2, mo);
        pstmt.setInt(3, day);
        ResultSet rs = pstmt.executeQuery();
        if(rs.next())
        {
        pstmt = con.prepareStatement("update customer set cust_bill=? where year=? and month=? and date=? and cust_name=? and cust_phone=?");
        pstmt.setInt(1,custbill);
        pstmt.setString(5, custname);
        pstmt.setString(6, custphone);
        pstmt.setInt(2,year);
```

```
pstmt.setInt(3,mo);
pstmt.setInt(4,day);
int n = pstmt.executeUpdate();
if(n!=0)
{
   System.out.println("Customer Details Updated Successfully");
}
else
{
   JOptionPane.showMessageDialog(this,"Customer Details Not Updated");
}
}
else
{
 pstmt = con.prepareStatement("insert into customer values (?,?,?,?,?,?)");
pstmt.setString(1,custname);
pstmt.setString(2,custphone);
pstmt.setInt(3,custbill);
pstmt.setInt(4,year);
pstmt.setInt(5,mo);
pstmt.setInt(6,day);
int n = pstmt.executeUpdate();
if(n!=0)
{
   System.out.println("Customer Details Added Successfully");
}
else
{
   JOptionPane.showMessageDialog(this,"Customer Details Not Added");
}
}
}
catch(Exception e)
{
```

```
            System.out.println("Exception raised:"+e);
        }
    }
    private boolean updatequant(int latestquan, String prodshort, String prodid) {
        boolean flag = false;
        try{
        Connection con=MY_DB.database.getConnection();
        PreparedStatement pstmt=null;
        if((prodshort!=null)&&(prodid.isEmpty()))
        {
            pstmt = con.prepareStatement("update products set prod_quantity=? where prod_shortcut=?");
            pstmt.setInt(1, latestquan);
                pstmt.setString(2,prodshort);
                int n = pstmt.executeUpdate();
                if(n!=0)
                {
                    flag = true;
                }
        }
        else if((prodid!=null)&&(prodshort.isEmpty()))
        {
            pstmt = con.prepareStatement("update products set prod_quantity=? where prod_id=?");
            pstmt.setInt(1, latestquan);
                pstmt.setString(2,prodid);
                int n = pstmt.executeUpdate();
                if(n!=0)
                {
                    flag = true;
                }
        }
        else if((prodid.isEmpty())&&(prodshort.isEmpty()))
        {
```

```
            JOptionPane.showMessageDialog(this,"Product Quantity Not updated");
            flag = false;
        }
        }
        catch(Exception e)
        {
            System.out.println("Exception raised:"+e);
        }
        return flag;
    }
    private void getreportdet(String prodid, String prodshort, int year, int mo,int quan) {
        String productid = "";
        try{
        Connection con=MY_DB.database.getConnection();
        PreparedStatement pstmt=null;
        if((prodshort!=null)&&(prodid.isEmpty()))
        {
            pstmt = con.prepareStatement("select * from products where
prod_shortcut=?");
            pstmt.setString(1,prodshort);
            ResultSet rs = pstmt.executeQuery();
            if(rs.next())
            {
                productid = rs.getString(1);
            }
        }
        else if((prodid!=null)&&(prodshort.isEmpty()))
        {
            productid = prodid;
        }
        addforreport(productid,year,mo,quan);
        }
        catch(Exception e)
        {
```

```
            System.out.println("Exception raised:"+e);
        }
    }
    private void addforreport(String productid, int year, int mo,int quan) {
        int latestquant,origquant;
         try{
        Connection con=MY_DB.database.getConnection();
        PreparedStatement pstmt=null;
        pstmt = con.prepareStatement("select * from checkreports where year=? and month=? and prod_id=?");
        pstmt.setInt(1,year);
        pstmt.setInt(2,mo);
        pstmt.setString(3,productid);
        ResultSet rs = pstmt.executeQuery();
        if(rs.next())
        {
            origquant = rs.getInt(4);
            latestquant = origquant + quan;
            pstmt = con.prepareStatement("update checkreports set quant_sold=? where year=? and month=? and prod_id=?");
            pstmt.setInt(1, latestquant);
            pstmt.setInt(2, year);
            pstmt.setInt(3, mo);
            pstmt.setString(4, productid);
            int n = pstmt.executeUpdate();
            if(n!=0)
            {
            }
        }
        else
        {
            pstmt = con.prepareStatement("insert into checkreports values(?,?,?,?)");
            pstmt.setInt(1, year);
```

```
            pstmt.setInt(2, mo);
            pstmt.setString(3,productid);
            pstmt.setInt(4,quan);
            int n = pstmt.executeUpdate();
            if(n!=0)
            {
            }
        }
        }
        catch(Exception e)
        {
            System.out.println("Exception raised:"+e);
        }
    }
    private double calculatecost(double cost, double discount, int quan) {
        double bill = 0,afterdisc = 0;
        afterdisc = ((discount/100)*cost);
        cost = cost - afterdisc;
        bill = cost * quan;
        return bill;
    }
    private void displaytemp(double totalbill, double totalquantity,String custname) {
        DefaultTableModel model1 = (DefaultTableModel)jTable1.getModel();
        model1.insertRow(model1.getRowCount(),new Object[] {"","","",""});
        model1.insertRow(model1.getRowCount(),newObject[]
{"TOTAL",(int)totalquantity,"",(int)totalbill});
        String strtotalbill = Double.toString(totalbill);
        int inttotalquantity = (int)totalquantity;
        String emptystr = "TOTAL";
        double emptystr1 = 0.0;
        addtooldfile(emptystr,inttotalquantity,emptystr1,strtotalbill,custname);
        openfile(custname);
    }
```

```
private void addtofile(String productname, int quan, double discount, String singleprodbill,String cname) {
        Date tod = new Date();
       int mo,year,day;
       mo = tod.getMonth()+1;
       year = tod.getYear()+1900;
       day = tod.getDate();
       int hours = tod.getHours();
       int min = tod.getMinutes();
        try{
           String filename = ""+year+""+mo+""+day+cname;
         FileWriter.fw=new
FileWriter("C:\\Users\\Admin\\Desktop\\Bill\\"+filename+".txt",true);
          fw.write("\t\t\t\tDEPARTMENTAL STORE\n\n");
          fw.write("ADDRESS: SIDDHARTHA NAGAR, KAZIPET, WARANGAL, 506004.\n");
          fw.write("PHONE: 9618141098\n");
          fw.write("DATE: "+day+"-"+mo+"-"+year+"\n");
          fw.write("TIME: "+hours+":"+min+"\n");
          fw.write("CUSTOMER NAME: "+cname+"\n");
          fw.write("\t\t\t\tBILL\n\n");
          fw.write("PRODUCT NAME\t PRODUCT QUANTITY\t PRODUCT DISCOUNT\t PRODUCT COST\n");
          fw.write("-------------------------------------------------------------------------------\n");
          fw.write(""+productname+"\t\t\t"+quan+"\t\t\t"+discount+"%\t\t\t "+singleprodbill+"\n");
          fw.close();
         }
        catch(Exception e){System.out.println(e);}
    }
   private void addtooldfile(String productname, int quan, double discount, String singleprodbill, String cname) {
            Date tod = new Date();
```

```
        int mo,year,day;
        mo = tod.getMonth()+1;
        year = tod.getYear()+1900;
        day = tod.getDate();
        try{
            String filename = ""+year+""+mo+""+day+cname;
          FileWriter.fw=new
FileWriter("C:\\Users\\Admin\\Desktop\\Bill\\"+filename+".txt",true);
          fw.write(""+productname+"\t\t\t"+quan+"\t\t\t"+discount+"%\t\t\t
"+singleprodbill+"\n");
          fw.close();
         }
        catch(Exception e){System.out.println(e);}
    }
   private void generatecustid() {
        try{
        Connection con=MY_DB.database.getConnection();
        Date tod = new Date();
        int mo,year,day;
        mo = tod.getMonth()+1;
        year = tod.getYear()+1900;
        day = tod.getDate();
        String cust_phone = "0000000000";
        PreparedStatement pstmt=con.prepareStatement("select count(*) from customer
where year=? and month=? and date=? and cust_phone=?");
        pstmt.setInt(1,year);
        pstmt.setInt(2,mo);
        pstmt.setInt(3,day);
        pstmt.setString(4,cust_phone);
        ResultSet rs = pstmt.executeQuery();
        if(rs.next())
        {
           i = rs.getInt(1)+1;
        }
```

```
            else
            {
                i = 1;
            }
            }
            catch(Exception e)
            {
                System.out.println("Exception raised:"+e);
            }
        }
        private void openfile(String fname) {
            Date tod = new Date();
            int mo,year,day;
            mo = tod.getMonth()+1;
            year = tod.getYear()+1900;
            day = tod.getDate();
            String filename = ""+year+""+mo+""+day+fname;
            try{
                File file = new File("C:\\Users\\Admin\\Desktop\\Bill\\"+filename+".txt");
            Desktop desktop = Desktop.getDesktop();
            if(file.exists()) desktop.open(file);
            }
            catch(Exception e)
            {
                System.out.println("Exception Raised:"+e);
            }
        }
    }
```

2.6 CASHIER_CHECH_PRODUCT_REPORTS

```
    public class Cashier_check_rep extends javax.swing.JFrame {
        public Cashier_check_rep() {
            initComponents();
        }
```

```
  private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
      String prodid;
      int fyear,tyear,fmonth,tmonth;
      prodid = jTextField3.getText();
      fyear = Integer.parseInt(jTextField1.getText());
      fmonth = Integer.parseInt(jTextField2.getText());
      tyear = Integer.parseInt(jTextField4.getText());
      tmonth = Integer.parseInt(jTextField5.getText());
      if((fyear == tyear)&&(fmonth == tmonth))
      {
         JOptionPane.showMessageDialog(this,"You Have Selected Same Year and
Month, You Will Have Constant Sales");
      }
      int initquant = getreportval1(prodid,fyear,fmonth);
      int finalquant = getreportval2(prodid,tyear,tmonth);
      int fquant=0,iquant=0;
      jLabel9.setText("QUANTITY SOLD IN THE PERIOD OF "+fmonth+"th
MONTH of "+fyear+" is "+initquant);
      jLabel11.setText("QUANTITY SOLD IN THE PERIOD OF "+tmonth+"th
MONTH of "+tyear+" is "+finalquant);
      if((initquant >= 100)||(finalquant >= 100))
      {
         iquant = (int)(initquant/10);
         fquant = (int)(finalquant/10);
      }
      else if((initquant >= 1000)||(finalquant >= 1000))
      {
         iquant = (int)(initquant/100);
         fquant = (int)(finalquant/100);
      }
      else if((initquant >= 10000)||(finalquant >= 10000))
      {
         iquant = (int)(initquant/1000);
         fquant = (int)(finalquant/1000);
```

```
    }
    else if((initquant >= 100000)||(finalquant >= 100000))
    {
        iquant = (int)(initquant/10000);
        fquant = (int)(finalquant/10000);
    }
    jProgressBar1.setValue(iquant);
    jProgressBar2.setValue(fquant);
    if(initquant > finalquant)
    {
        jLabel13.setText("Hmm! Decrease in Sales");
    }
    else if(initquant < finalquant)
    {
        jLabel13.setText("Yeah! Increase in Sales");
    }
    else if(initquant == finalquant)
    {
        jLabel13.setText("Constant Sales");
    }
}
private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    Cashier_main cm = new Cashier_main();
    cm.setVisible(true);
}
private void formWindowOpened(java.awt.event.WindowEvent evt) {
    this.getContentPane().setBackground(Color.orange);
}
private int getreportval1(String prodid, int fyear, int fmonth) {
     int initquant=0;
     try{
    Connection con=MY_DB.database.getConnection();
    PreparedStatement pstmt = null;
```

```
        pstmt = con.prepareStatement("select * from checkreports where year=? and
month=? and prod_id=?");
        pstmt.setInt(1,fyear);
        pstmt.setInt(2,fmonth);
        pstmt.setString(3,prodid);
        ResultSet rs = pstmt.executeQuery();
        if(rs.next())
        {
           initquant = rs.getInt(4);
        }
        else
        {
           JOptionPane.showMessageDialog(this,"No Details Found in the given
period");
        }
        }
        catch(Exception e)
        {
           System.out.println("Exception raised:"+e);
        }
         return initquant;
    }
    private int getreportval2(String prodid, int tyear, int tmonth) {
         int finalquant=0;
         try{
        Connection con=MY_DB.database.getConnection();
        PreparedStatement pstmt = null;
        pstmt = con.prepareStatement("select * from checkreports where year=? and
month=? and prod_id=?");
        pstmt.setInt(1,tyear);
        pstmt.setInt(2,tmonth);
        pstmt.setString(3,prodid);
        ResultSet rs = pstmt.executeQuery();
        if(rs.next())
```

```
        {
            finalquant = rs.getInt(4);
        }
        else
        {
            JOptionPane.showMessageDialog(this,"No Details Found in the given
period");
        }
        }
        catch(Exception e)
        {
            System.out.println("Exception raised:"+e);
        }
         return finalquant;
    }
}
```

2.7 CASHIER_CHANGE_PASSWORD

```
public class Cash_change_pass extends javax.swing.JFrame {
    public Cash_change_pass() {
        initComponents();
    }
String userid;
    public Cash_change_pass(String id) {
         initComponents();
         userid = id;
    }
    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        Cashier_main cm = new Cashier_main();
        cm.setVisible(true);
    }
    private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
        String oldpass,newpass,cnfnewpass;
        oldpass = jTextField1.getText();
```

```
        newpass = jTextField2.getText();
        cnfnewpass = jTextField3.getText();
        if(newpass.equalsIgnoreCase(cnfnewpass))
        {
           getuserdet(userid,oldpass,newpass,cnfnewpass);
        }
        else
        {
           JOptionPane.showMessageDialog(this,"New Passwords are not Same, Please try Again");
        }
    }
    private void getuserdet(String userid, String oldpass, String newpass, String cnfnewpass) {
         try{
        Connection con=MY_DB.database.getConnection();
        PreparedStatement pstmt = null;
        pstmt = con.prepareStatement("select * from cashier where p_email=?");
        pstmt.setString(1,userid);
        ResultSet rs=pstmt.executeQuery();
        if(rs.next())
        {
           String ooldpass = rs.getString("p_pass");
           if(ooldpass.equals(oldpass))
           {
           pstmt = con.prepareStatement("update cashier set p_pass=? where p_email=?");
           pstmt.setString(1, newpass);
           pstmt.setString(2, userid);
           int n = pstmt.executeUpdate();
           if(n!=0)
           {
              JOptionPane.showMessageDialog(this,"Password Updated");
           }
           else
```

```
        {
            JOptionPane.showMessageDialog(this,"Password not updated");
        }
        }
    }
    else
    {
        JOptionPane.showMessageDialog(this,"Please Enter correct Old Password");
    }
    }
    catch(Exception e)
    {
        System.out.println("Exception raised:"+e);
    }
} }
```

6. RESULTS

HOME PAGE

Image 6.1

MANAGER HOME PAGE ACTIVITIES

Image 6.2

MANAGER ADD CASHIER DETAILS

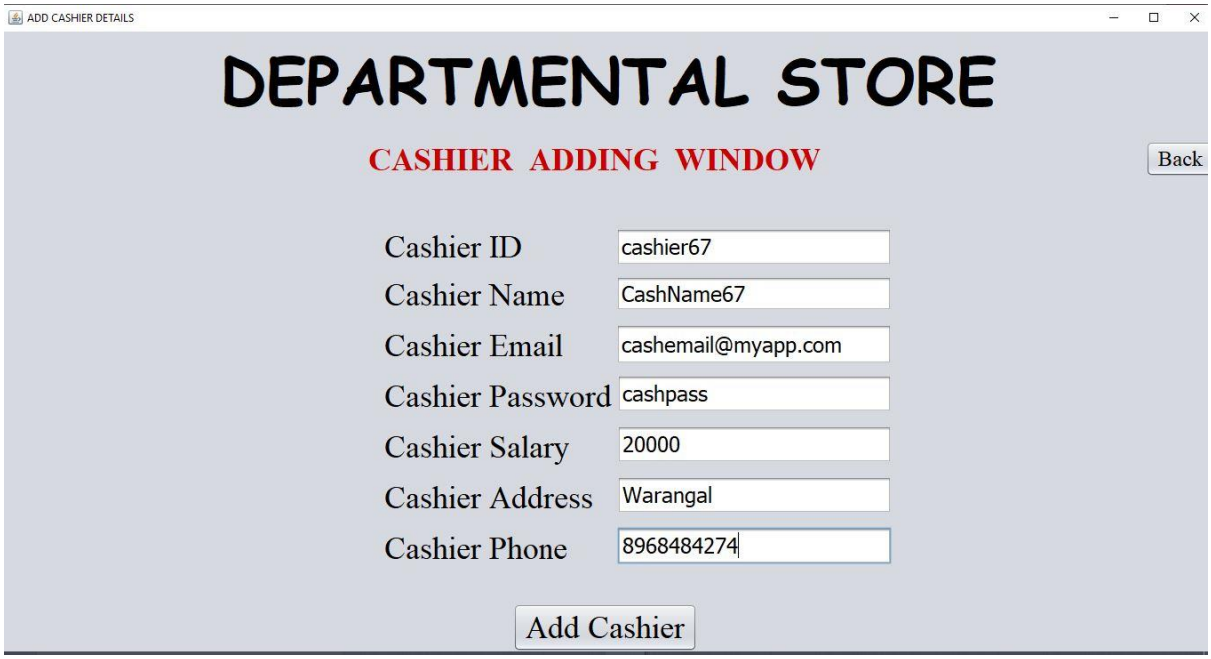

Image 6.3

MANAGER EDIT CASHIER DETAILS

Image 6.4

MANAGER VIEW CASHIER DETAILS

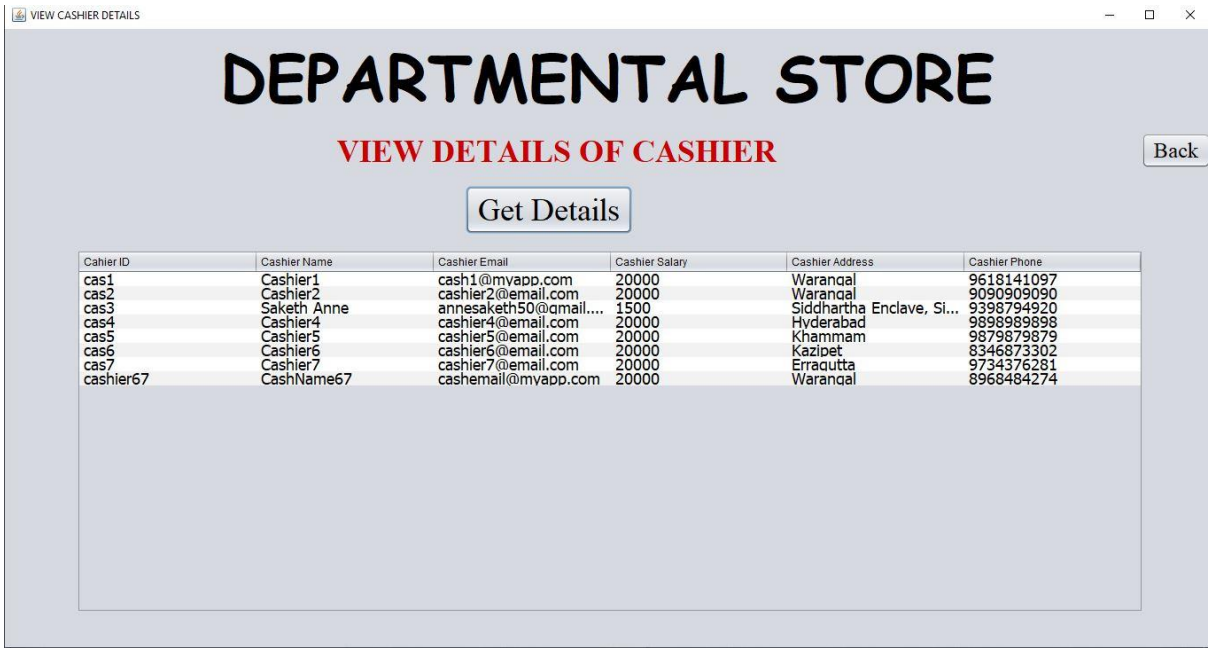

Image 6.5

MANAGER REMOVE CASHIER DETAILS

Image 6.6

MANAGER CHECK CUSTOMERS DETAILS

Image 6.7

CASHIER HOME PAGE ACTIVITIES

Image 6.8

CASHIER VIEW AND ADD STOCK PRODUCT DETAILS

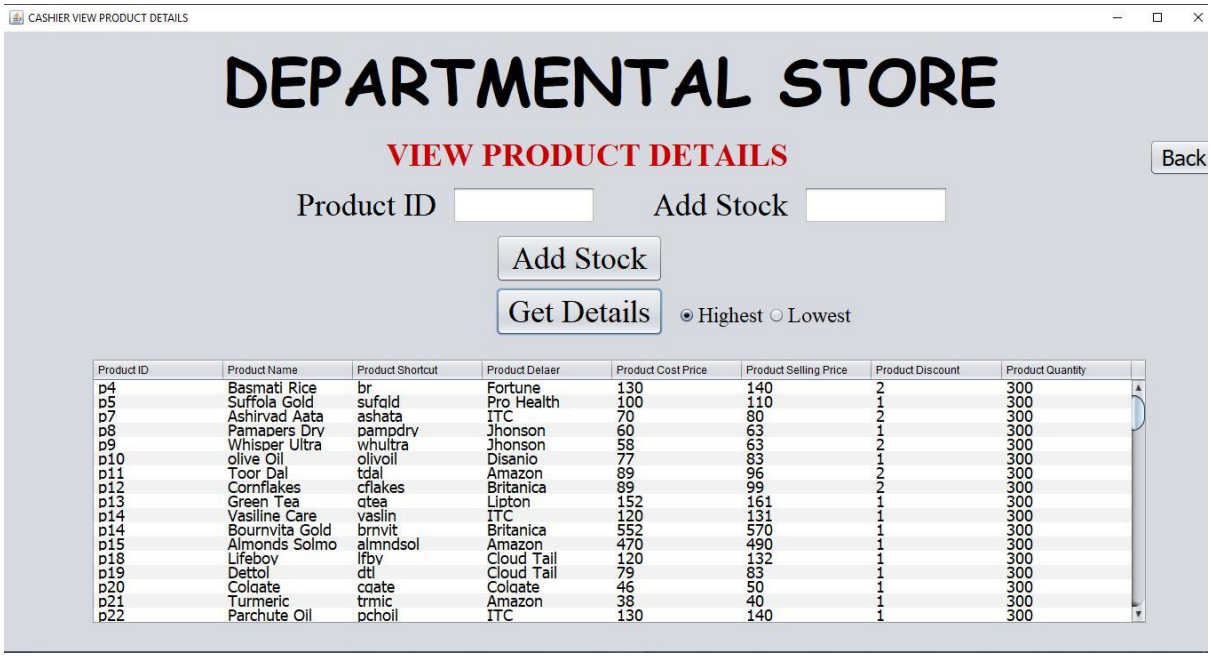

Image 6.9

CASHIER EDIT PRODUCT DETAILS

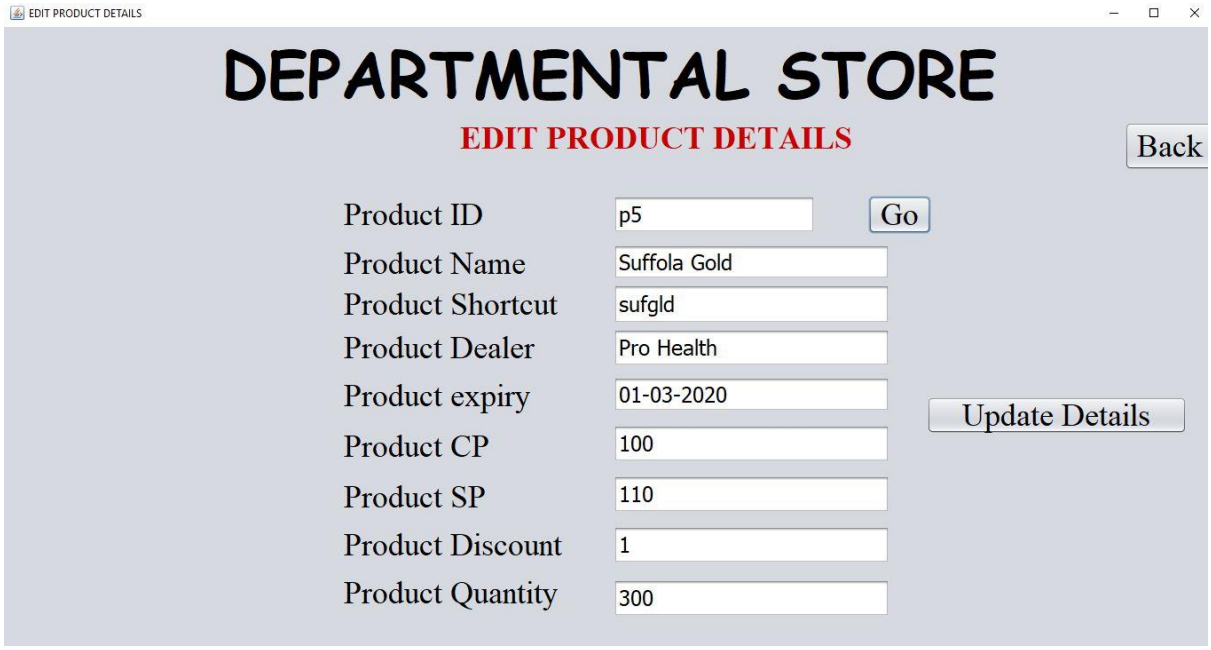

Image 6.10

CASHIER CHECK REPORT DETAILS

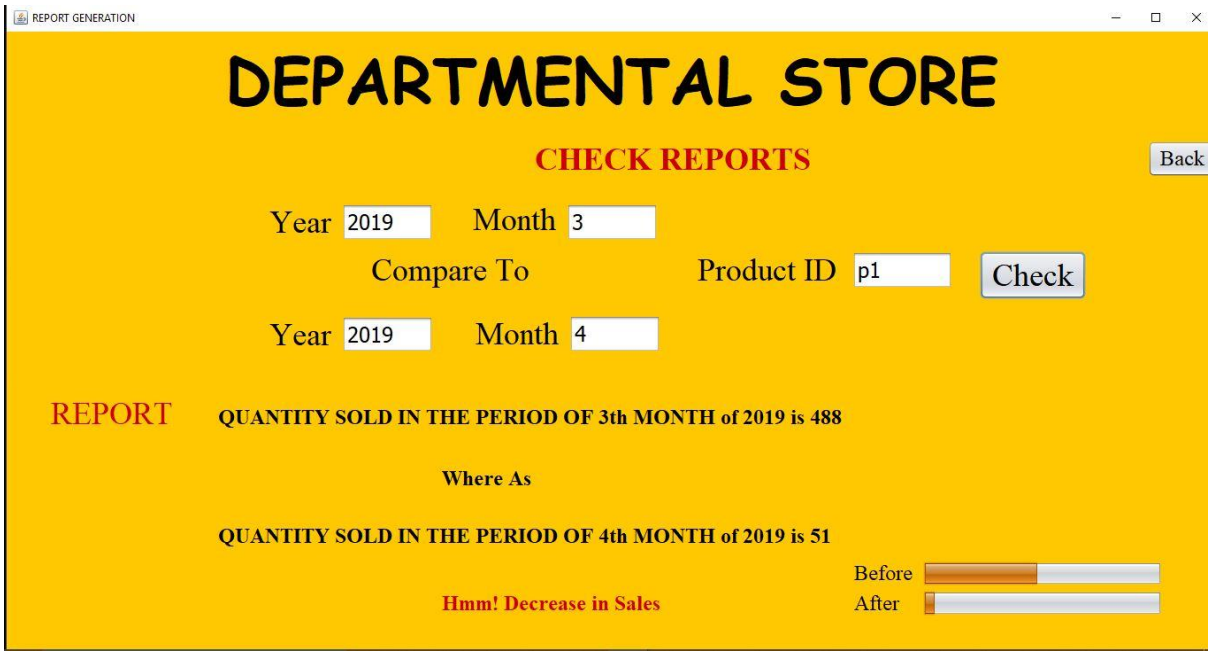

Image 6.11

CASHIER GENERATE BILL

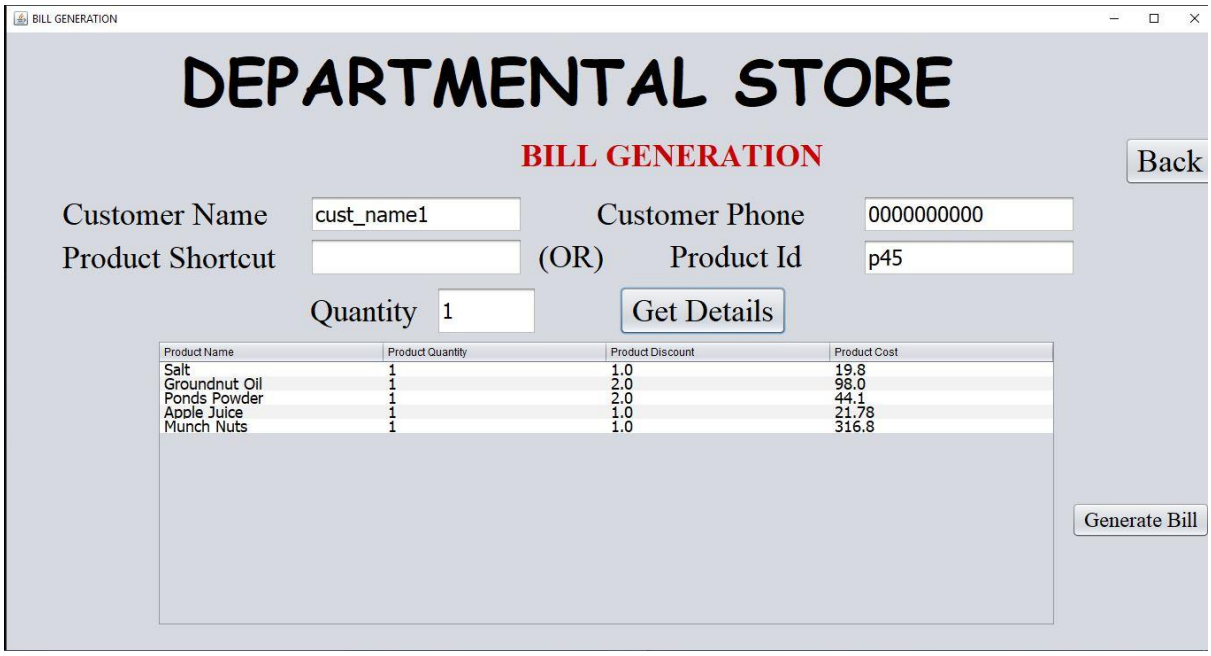

Image 6.12

BILL DOCUMENT

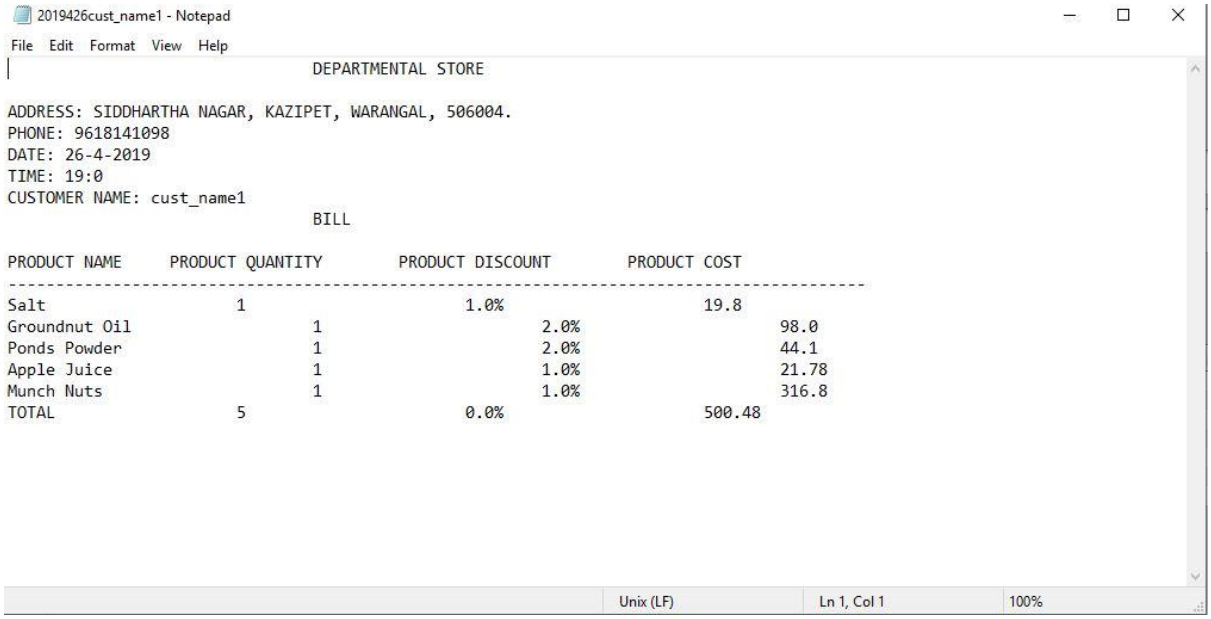

Image 6.13

7. TESTING

System testing is the stage before system implementation where the system is made error free and all the needed modifications are made. The system was tested with test data and necessary corrections to the system were carried out. All the reports were checked by the user and approved. The system was very user friendly.

Security Testing of the Project

Testing is vital for success of any software. No system is ever perfect. Testing is also carried in two phases. First phase is during the software engineering that is during the module creation. Second phase is after the completion of software. This is system testing which verifies that the whole set of programs handed together.

Test Plan

A test plan is a general document for the entire project, which defines the scope, approach to be taken, and schedule of testing, as well as identifying the test item for the entire testing process, and the personal responsible for the different activities of testing. This document describes the plan for testing, the knowledge management tool.

Major testing activities are:

- Test units
- Features to be test
- Approach for testing
- Test deliverables
- Schedule
- Personal allocation

8. CONCLUSION AND FUTURE SCOPE

Time and money are one of the most important factors to anyone. Implementing such software in the departmental store can surely be a profitable deal as this application helps to carry out tasks with ease and thereby reduces time and money on manpower and materials. It provides a great use in managing the data in a well-ordered manner. It is designed according to day-to-day need of the people.The project which I undertaken has helped me gain a better perspective on various aspects related to my course of study as well as particular knowledge of Java GUI based applications. I became familiar with software analysis, designing, implementation, testing and maintenance considered with my project.

BIBLIOGRAPHY

Textebooks:

- Java fx-Client programming on netbeans platform-Addisom-Wesley
- Netbeans IDE8-cook book

Websites

- www.stackoverflow.com
- www.wikipedia.com
- www.tutorialspoint.com
- www.javapoint.com