# CHAPTER-1 INTRODUCTION OF PROJECT

**1.1 Introduction:**

Product Billing System is a computerized systemI which helps its user to save time to generate bills. Supermarket is the place where customers come to purchase i their daily using products and pay for that. So there is a need to calculate how many products are sold and to generate the bill for i the customer. It reduces the risk of paper work such as file lost, file damaged and time consuming. It can help user to manage the transaction or record more effectively i and time-saving.

Telecommunication companies need an effective and accurate billing system to be able to assure their revenue. Billing systems process the usage of network equipment that is used during the service usage into a single Call Detaili Record (CDR).



## Figure 3.1: Software Logo

The billing process involves receiving billing records from various networks, determining the billing rates associated with the billingi records, calculating the cost for each billing record, aggregating these records periodically to generate invoices, sending invoices to the customer, and collecting payments receivedi from the customer. Billing system is very complex startingi from network elements that generate usage to the billing system to usage collection, mediation, rating, and invoicing.

In business IT, billingi software refers to programs that handlei the tracking of billable products and services delivered to a customer or set of customers. Some billing software also tracks work hours for billingi purposes. These types of programs automatei much of what used to be a time-consumingi process of preparing invoices or other documentation.

The modern digital structures providedi by billing software services and products are part of whati has propelled businesses into the new digital era, allowingi for more productivity and greater ease of business administration in general.

**PROBLEM STATEMENT:**

The problem occurred beforei having computerized system includes:

* File lost :-

When computerized system is not implementedi file is always lost because of human environment. Sometimesi due to some human error there may be a loss of records.

* File damaged :-

When a computerized systemi is not there file is always lost due to some accident like spilling of water by somei member on file accidentally. Besides some natural disaster like floods or fires may also damage the files.

* Difficult to search record :-

When there is no computerizedi system there is always a difficultyi in searching of records if the records are large in number.

* Space consuming :-

After the numberi of records become large the space for physicali storagei of file and records also increasesi if no computerizedi system is implemented. cords are large in number .

**1**.2 **Objective:**

The objective and the scope of my projecti Producti Billing Softwarei is to provide professional environment to modify the process of generatingi invoices for supermarket store. It simplifiesi the task and saves time. During implementationi an admin can only login. It means it supports “Administrativei Login” only and avoid any kind of chaos.

This software will generate the resultsi without any interferencei and will give 100% accuracy.

It includes the followingi points:-

* Improvement in control and performance:-

The system is developedi to cope up with the current issuesi and problemsi of library.

The system can add user, validatei user and is also bug free.

* Save cost :-

After computerizedi system is implementedi less human force will be required to maintain the institute thus reducingi the overall cost.

* Save time :-

User is able to searchi record by using few clicksi of mouse and few searchi keywords thus saving his valuable time.

**1.3 Features:**

* Securei
* Easyi to use
* Reliablei and accurate
* Saving of paper
* Timei Saving
* Chaos is avoided
* Datai is saved in Database.
* Useri friendly interface

**CHAPTER-2**

# SOFTWARE AND HARDWARE REQUIREMENTS

**2.1 Requirements and Specifications:**

In this chapter, we will discuss and analyzei about the developing process of Institute Management System including softwarei requirement specificationi (SRS) and comparisoni betweeni existing and proposed system. The functional and non- functional requirements are included in SRS parti to provide completei description and overview of system requirement before the developingi process is carriedi out. Besides that, existing vs proposedi provides a view of how the proposedi system will be more efficienti than the existing one.

**2.1.1 Software Requirements:**

The software requirementsi for this programi are basic operatingi system, one programming languagei and the software or complier to compilei the source program. The detailed description of softwarei requirements are:-

Operatingi System - Windows XP/Windows 7/Windows 8i or any other versions

Language Used - Core Javai

Software Used - My Eclipse Blue Editioni

Database Used - Microsoft SQL Server

**2.1.2 Hardware Requirements:**

The minimum hardwarei requirements for this program are processori with normal speed, he limitedi memory is required to run this program. As, this programi are written in C++. This program does not need higheri hardware specifications. The detailed hardware specifications for this programi are described below :-Processor - Processor: 800MHz Intel Pentium III or Equivalenti

Ram - Min 512 MbHardi Disk - Min 1 GB or morei

# CHAPTER-3 SOFTWARE REQUIREMENT ANALYSIS

**3.1 Introduction To Java:**

Java is a general-purpose [computer-programming](https://en.wikipedia.org/wiki/Programming_language) that is [concurrent,](https://en.wikipedia.org/wiki/Concurrent_computing) [classbased,](https://en.wikipedia.org/wiki/Class-based_programming) [object-oriented,](https://en.wikipedia.org/wiki/Object-oriented_programming) and specificallyi designed to have as few implementationi dependenciesi as possible. It is intended to let applicationi developers "w[rite once, run anywhere"](https://en.wikipedia.org/wiki/Write_once,_run_anywhere) (WORA), meaning that [compiled](https://en.wikipedia.org/wiki/Compiler)iJava code can run on all platforms that support Javai without the need for recompilation. Java applicationsi are typically compiled to [bytecode](https://en.wikipedia.org/wiki/Java_bytecode) that can run on any [Java virtual machine](https://en.wikipedia.org/wiki/Java_virtual_machine) (JVM) regardless of [computeri architecture.](https://en.wikipedia.org/wiki/Computer_architecture) As of 2016, Java is one of the most [popular programming languages in use,](https://en.wikipedia.org/wiki/Measuring_programming_language_popularity) particularly for client-server web applications, with a reported 9 millioni developers. Java was originallyi developed by [James Gosling](https://en.wikipedia.org/wiki/James_Gosling) at [Sun Microsystems](https://en.wikipedia.org/wiki/Sun_Microsystems) and released in 1995 as a core componenti of Sun Microsystems' [Java platform.](https://en.wikipedia.org/wiki/Java_(software_platform)) The language derivesi much of its [syntax](https://en.wikipedia.org/wiki/Syntax_(programming_languages)) from [C](https://en.wikipedia.org/wiki/C_(programming_language)) and [C++,](https://en.wikipedia.org/wiki/C%2B%2B) but it has fewer [low-level](https://en.wikipedia.org/wiki/Low-level_programming_language) facilities than either of them.

**Figure 3.**

**1**

**:**

**JAVA**



**CHARACTERISTICS OF JAVA:**

* Platform independenti: Java programs use the Java virtual machinei as abstraction and do not access the operatingi system directly. This makes Java programsi highly portable. A Java program can run unmodifiedi on all supported platforms, e.g.

Windowsi or Linux.

* Object-orientated programming languageii: Except the primitive data types, all elementsi in Javai are objects.
* Interpreted and compiled languagei: Java source code is transferredi into the bytecodei format which does not depend on the targeti platform. These bytecode instructions will be interpreted by the Java Virtual machine (JVM). The JVM contains a so called Hotspot-Compileri which translates performance critical bytecode instructionsi into native code instructions.
* Automatic memory managementi: Java managesi the memory allocation and deallocationi for creating new objects. The program does not have direct accessi to the memory. The so-called garbage collector automaticallyi deletes objects to which no active pointeri exists.

**3.2 Introduction To Microsoft SQL Server:-**

Microsoft SQL Server is a relational database management system developed by Microsoft. As a database server, it is a software product with the primary function of storing and retrieving data as requested by other software applications—which may run either on the same computer or on another computer across a network (including the Internet).

Microsoft markets at least a dozen different editions of Microsoft SQL Server, aimed at different audiences and for workloads ranging from small single-machine applications to large Internet-facing applications with many concurrent users.

* It is a software, developed by Microsoft, which is implemented from the specification of RDBMS. It is also an ORDBMS.
* It is platform dependent.
* It is both GUI and command based software.
* It supports SQL (SEQUEL) language which is an IBM product, non-procedural, common database and case insensitive language.
* It analyze the data through SQL Server Analysis Services (SSAS).
* It generate reports through SQL Server Reporting Services (SSRS).
* It carry out ETL operations through SQL Server Integration Services (SSIS)



* 1. **Introduction to Product Billing Software:**

The Project “Product Billing Software” deals with the automation of supermarket billing system. This software will help salespersons in managing the various types of Records pertaining to his/her customer. The product will help the user to work in a highly effective and efficient environment. The salespersons have been recording the customer information in the past and even in the present through their personal manual efforts. And indeed, it consumes their considerable time and energy that could be utilized in the better productive activities.

Telecommunication companies need an effective and accurate billing system to be able to assure their revenue. Billing systems process the usage of network equipment that is used during the service usage into a single Call Detail Record (CDR).

The billing process involves receiving billing records from various networks, determining the billing rates associated with the billing records, calculating the cost for each billing record, aggregating these records periodically to generate invoices, sending invoices to the customer, and collecting payments received from the customer.

* 1. **Problems with Existing System:**

There are lots of problems with the existing manual system of invoicing. These problems are:-

* **Expensive & time Consuming:-**The main problem of the existing system is it is very expensive. It will include the process like collecting of data & entering the data into the database takes too much time is expensive to conduct, for example, time & money is spent in printing forms.
* **Too much paper work:-**The process involves too much paper work & paper storage which is difficult as papers become bulky.
* **Errors during data entry:-**Errors are part of all human beings; it is very unlucky for humans to be 100 % efficient in data entry. This problem can be easily resolved with the help of Product Billing system.
* **Loss of Invoices-**The main problem of the existing system is the loss of invoices. The loss will be lead to error in accountancy details.

**3.5 Modules & Their Functionality:**

* Login
* Vendor Details
* Product Details
* Customer Details
* Generate Bill
* Customer Payment Details
* Purchased Items Details

The description of the different modules are explained in brief below:-

* **Login:** It is the login session for the administrator or exit the software. These module will help the administrator to login into software by providing correct password. And then, it is able to manipulate the details for different stakeholders like customers and vendors and also can generate the invoices.
* **Vendor Details:** It is used to add, view, update and delete the vendor details from database. In this module, information of each vendor including its personal details is added in the database according to the vendor id of the vendor.
* **Product Details:** It is used to add, view, update and delete the product details from database. In this module, information of each product including its name, price, quantity, net amount, g.s.t, total amount details is added in the database according to the product id of the product.
* **Customer Details:** It is used to add, view, update and delete the customer details from database. In this module, information of each customer including its personal details is added in the database according to the customer id of the customer.
* **Generate Bill:** This module is used to create the invoice which includes the list of product purchased by the customer. And then, the details of purchased items is stored into database and then it display the final bill with a print option where you can print the bill.
* **Customer Payment Details:** In this module, information of each customer payment record including customer name, net amount, g.s.t, total amount, date, time is added in the database according to the invoice number.
* **Purchased Items:** In this module, information of each customer purchased products record including product name, price, quantity, net amount, g.s.t, total is added in the database according to the invoice number

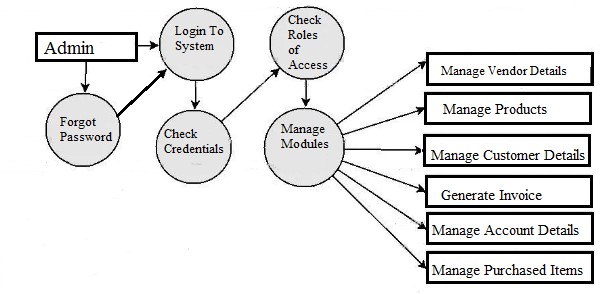
# CHAPTER-4 SYSTEM DESIGN

**4.1 Data Flow Diagram:**

A Data Flow Diagram(DFD) is a graphical representation that depicts information flow & the transforms that are applied as data move from input to output. The basic form of a data flow diagram, also known as a data flow graph or bubble chart. The data flow diagram may be used to represent a system or software at any level of abstraction.

As information moves through software, it is modified by a series of transformations. DFD may be partitioned into levels that represent increasing flow & functional detail. Therefore, the DFD provides a mechanism for functional modelling as well as information flow modelling. DFDs are very useful in understanding a system & can be effectively used during analysis. DFDs can be hierarchically organized, which helps in progressively partitioning & analyzing large systems.

Figure 4.1 Data Flow Diagram



**4.2 Use Case Diagram:**

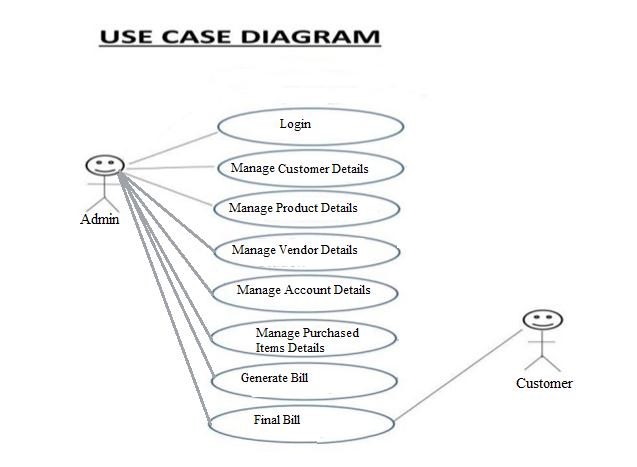
It shows various activities the users can perform on the system they model the dynamic aspects of system. It provide user perspective of the system. It includes:- **Actor:**-An actor is a user of system playing a particular role.

**Use case:**-Use case is particular activity a user can do on the system.

**Relationship:**-Relationship are simply make the link between the actors & use cases with a line.

Figure 4.2

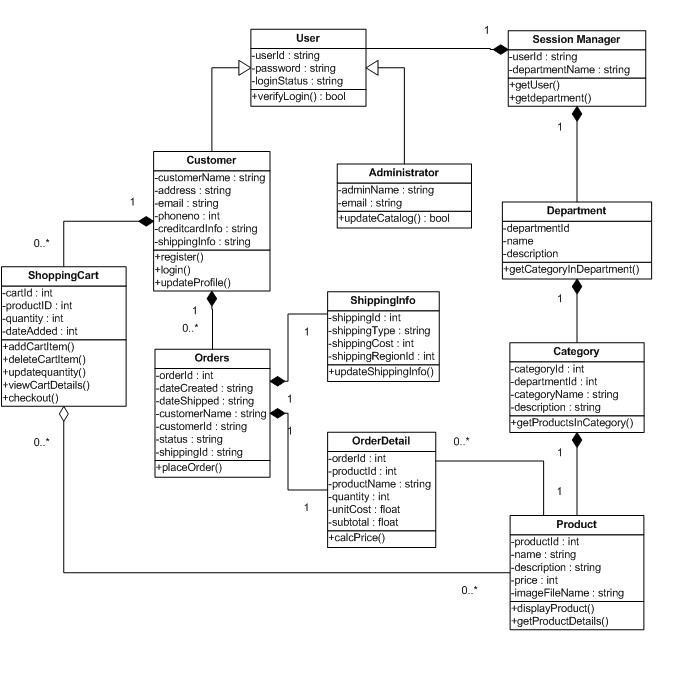
Case Diagram



**4.3 Class Diagram:**

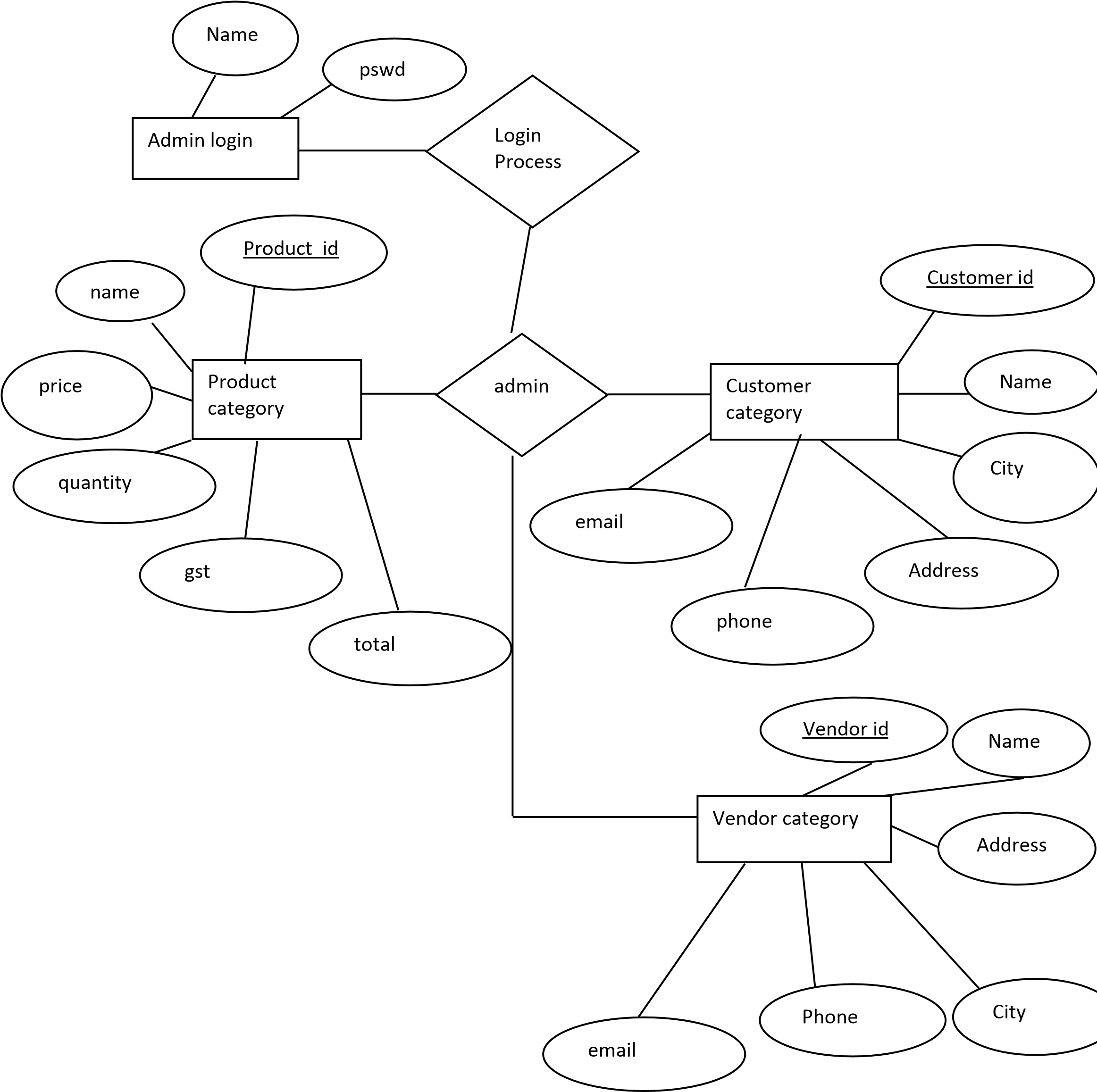
A class diagram describes the types of objects in the system & the various kinds of static relationships that exist among them i.e. A graphical representation of a static view on declarative static elements. A class is the description of a set of objects having similar attributes, operation, relationships & behaviour.

Figure 4.3 Class Diagram



**4.4 E-R Diagram:**

An E-R model stands for **Entity-Relationship Model** which describes the interrelated things of interest in a specific domain of knowledge. A basic ER model is composed of entity types & specifies relationships that can exist between instances of those entity types. It was developed for database design by **Peter Chen.**



**Figure 4.4: E-R Diagram**

# CHAPTER-5 SOURCE CODE OF THE PROJECT

**5.1 Database Tables:**

There are different types of tables are used in the Product Billing System project.

These are called Database Table. The different database Tables are:- **1. Login Details table:**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No.** | **Field Name** | **Type** | **Description** |
| 1. | Serial | varchar | Serial Number |
| 2. | Username | varchar | User Name |
| 3. | Password | Image | Password |

## Table 5.1: Login Details Table

**2. Vendor Details Table:-**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No.** | **Field Name** | **Type** | **Description** |
| 1. | Vendor id | Int | Vendor Id(primary key) |
| 2 | Name | Varchar | Name of vendor |
| 3 | Address | Varchar | Address |
| 4 | City | Varchar | City |
| 5 | Phone | Char | Phone |
| 6 | Email | Varchar | Email |

## Table 5.2: Vendor Details Table

1. **Customer Details Table:-**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No.** | **Field Name** | **Type** | **Description** |
| 1. | Customer id | Int | Customer Id(primary key) |
| 2 | Name | Varchar | Name of customer |
| 3 | Address | Varchar | Address |
| 4 | City | Varchar | City |
| 5 | Phone | Char | Phone |
| 6 | Email | Varchar | Email |

**Table 5.3: Customer Details Table**

1. **Product Details Table:-**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No.** | **Field Name** | **Type** | **Description** |
| 1. | Product id | Int | Customer Id(primary key) |
| 2 | Name | Varchar | Product Name |
| 3 | Vendor Name | Varchar | Vendor name |
| 4 | Price | Int | City |
| 5 | Quantity | Int | Phone |
| 6 | Net Amount | Int | Email |
| 7 | GST | Int | GST |
| 8 | Total | Int | Total |
| 9 | DateTime | datetime | Date And Time |

**Table 5.4: Product Details Table**

1. **Customer Payment Details Table:-**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No.** | **Field Name** | **Type** | **Description** |
| 1. | Billno | Int | Invoice No(primary key) |
| 2 | CustomerID | Varchar | Customer ID |
| 3 | Name | Varchar | Customer Name |
| 4 | TotalQuantity | Int | Total Quantity |
| 5 | Total | Int | Total |
| 6 | DateTime | datetime | Date And Time |

**Table 5.5: Customer Payment Details Table**

1. **Purchased Items Table:-**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No.** | **Field Name** | **Type** | **Description** |
| 1. | Billno | Int | Invoice No(primary key) |
| 2 | Name | Varchar | Product Name |
| 3 | Price | Int | Price |
| 4 | Quantity | Int | Quantity |
| 5 | NetAmt | Int | Net Amount |
| 6 | GST | Int | GST |
| 7 | Total | Varchar | Total |

## Table 5.6: Customer Payment Details Table

**5.2 Coding:**

**import** java.awt.\*; import java.awt.\*; import java.awt.event.\*; import javax.swing.\*; import java.sql.Connection; import java.sql.DriverManager; import java.sql.PreparedStatement; import java.sql.ResultSet; import java.text.DecimalFormat; import java.util.\*; public class AddProduct implements

ActionListener,Runnable,KeyListener

{

GregorianCalendar gc;

Thread datetimeThread;

JPanel panel;

Font headingFont; JLabel

headingLabel,productidLabel,nameLabel,priceLabel,quantityLabel,gstLab el,netamtLabel,vendoridLabel,vendornameLabel,vendoraddressLabel,vendo rcityLabel,vendorcontactLabel,totalLabel,dateLabel,timeLabel,vendorem ailLabel;

JTextField

productidText,nameText,priceText,quantityText,gstText,netamtText,vend orcityText,vendornameText,vendorcontactText,totalText,dateText,timeTe xt,vendoremailText;

|  |  |  |
| --- | --- | --- |
|  | JScrollPane addressScroll; | |
|  | JTextArea addressTextarea; | |
|  | JButton saveButton, resetButton; | |
|  | JComboBox vendoridCombo; | |
|  | JSeparator sp; | |
|  | public AddProduct() | |
|  | { |  |
|  |  | panel=new JPanel(); |
|  |  |  |
|  |  | headingLabel=new JLabel("Add Product Details"); |
|  |  | dateLabel=new JLabel("Date:"); |
|  |  | timeLabel=new JLabel("Time:"); |
|  |  | productidLabel=new JLabel("Product Id: "); |
|  |  | nameLabel=new JLabel("Enter Product Name : "); |
|  |  | priceLabel=new JLabel("Enter Product Price : "); |
|  |  | quantityLabel=new JLabel("Enter Quantity : "); |
|  |  | gstLabel=new JLabel("Enter G.S.T : "); |
|  |  | netamtLabel=new JLabel("Net Amount : "); |
|  |  | vendoridLabel=new JLabel("Vendor Id: "); |
|  |  | vendornameLabel=new JLabel("Enter Vendor's Name : "); |
| "); |  | vendoraddressLabel=new JLabel("Enter Vendor's Address : |
|  |  | vendorcityLabel=new JLabel("Enter Vendor's City : "); |
| "); |  | vendorcontactLabel=new JLabel("Enter Vendor's Contact : |
|  |  | vendoremailLabel=new JLabel("<html>Enter Email ID : |

<br>(Optional)</html>");

totalLabel=new JLabel("Total : ");

dateText=new JTextField(10); timeText=new JTextField(10); productidText=new JTextField(10); nameText=new JTextField(10); priceText=new JTextField(10); quantityText=new JTextField(10); gstText=new JTextField(10); netamtText=new JTextField(10); vendornameText=new JTextField(10);

vendornameText.setText(null); vendorcityText=new JTextField(10); vendorcontactText=new JTextField(10); vendoremailText=new JTextField(10);

totalText=new JTextField(10);

addressTextarea=new JTextArea(5,10); addressTextarea.setLineWrap(true);

addressTextarea.setWrapStyleWord(true);

addressScroll=new JScrollPane(addressTextarea);

saveButton=new JButton("Save Details"); saveButton.setFocusPainted(false); resetButton=new JButton("Reset");

resetButton.setFocusPainted(false);

vendoridCombo=new JComboBox(); vendoridCombo.insertItemAt("none", 0);

vendoridCombo.setSelectedIndex(0);

sp = new JSeparator(); sp.setBackground(Color.BLACK);

headingFont=new Font("Times New Roman",

Font.BOLD+Font.ITALIC, 50);

headingLabel.setBounds(10,10,500,50); headingLabel.setFont(headingFont);

sp.setBounds(0,80,1500,30);

dateLabel.setBounds(900,100,180,30); dateText.setBounds(950,100,100,30); timeLabel.setBounds(900,140,180,30); timeText.setBounds(950,140,100,30); productidLabel.setBounds(10,100,180,30); productidText.setBounds(190,100,180,30); nameLabel.setBounds(10,140,180,30); nameText.setBounds(190,140,180,30); priceLabel.setBounds(10,180,180,30); priceText.setBounds(190,180,180,30); quantityLabel.setBounds(10,220,180,30); quantityText.setBounds(190,220,180,30); netamtLabel.setBounds(10,260,180,30); netamtText.setBounds(190,260,180,30); gstLabel.setBounds(10,300,180,30); gstText.setBounds(190,300,180,30); totalLabel.setBounds(10,340,180,30); totalText.setBounds(190,340,180,30); saveButton.setBounds(350,440,120,30); resetButton.setBounds(500,440,120,30); vendoridLabel.setBounds(440,100,180,30); vendoridCombo.setBounds(630,100,180,30); vendornameLabel.setBounds(440,140,180,30);

vendornameText.setBounds(630,140,180,30); vendoraddressLabel.setBounds(440,180,180,30); addressScroll.setBounds(630,180,180,70); vendorcityLabel.setBounds(440,260,180,30); vendorcityText.setBounds(630,260,180,30); vendorcontactLabel.setBounds(440,300,180,30);

vendorcontactText.setBounds(630,300,180,30); vendoremailLabel.setBounds(440,340,180,30);

vendoremailText.setBounds(630,340,180,30);

priceText.addKeyListener(this); quantityText.addKeyListener(this); gstText.addKeyListener(this); saveButton.addActionListener(this); resetButton.addActionListener(this);

vendoridCombo.addActionListener(new ActionListener()

{

public void actionPerformed(ActionEvent evt)

{

if(vendoridCombo.getSelectedIndex()==0)

{

vendornameText.setText(""); addressTextarea.setText(""); vendorcityText.setText(""); vendorcontactText.setText("");

vendoremailText.setText("");

}

try

{

Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");

Connection

con=DriverManager.getConnection("jdbc:odbc:GaganExpress");

String tmp=(String)

vendoridCombo.getSelectedItem();

PreparedStatement

ps=con.prepareStatement("select name,address,city,phone,email from VendorDetails where vendorId=?");

ps.setString(1,tmp);

ResultSet rs=ps.executeQuery();

//insert, update, delete

if(rs.next())

{

String name=rs.getString("name");

String

vendoraddress=rs.getString("address");

String city=rs.getString("city");

String phone=rs.getString("phone");

String vendoremail=rs.getString("email");

vendornameText.setText(name);

addressTextarea.setText(vendoraddress.toString().trim());

vendorcityText.setText(city); vendorcontactText.setText(phone);

vendoremailText.setText(vendoremail.toString().trim());

}

}

catch(Exception ee)

{

}

}

});

panel.setLayout(null);

panel.add(headingLabel); panel.add(sp); panel.add(dateLabel); panel.add(dateText); panel.add(timeLabel); panel.add(timeText); panel.add(productidLabel); panel.add(productidText); panel.add(nameLabel); panel.add(nameText); panel.add(gstLabel); panel.add(gstText); panel.add(priceLabel); panel.add(priceText); panel.add(quantityLabel); panel.add(quantityText); panel.add(netamtLabel); panel.add(netamtText); panel.add(vendoridLabel); panel.add(vendoridCombo); panel.add(vendornameLabel); panel.add(vendornameText); panel.add(vendoraddressLabel); panel.add(addressScroll); panel.add(vendorcityLabel); panel.add(vendorcityText); panel.add(vendorcontactLabel); panel.add(vendorcontactText); panel.add(vendoremailLabel); panel.add(vendoremailText); panel.add(totalLabel);

panel.add(totalText);

panel.add(saveButton);

panel.add(resetButton);

try

{

Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");

Connection

con=DriverManager.getConnection("jdbc:odbc:GaganExpress");

PreparedStatement ps=con.prepareStatement("select \* from VendorDetails");

ResultSet rs=ps.executeQuery();

# CHAPTER-6 TESTING

**6.1 Introduction to Testing:**

Testing is the process of running a system with the intention of finding errors. Testing enhances the integrity of a system by declaring deviations in design & errors in the system. Testing aims at detecting error-prone areas. This helps in the prevention of errors in a system. Testing also adds value to the product by conforming to the user requirements.

The main purpose of testing is to detect errors &errorprone areas in a system. Testing must be thorough & well planned. A partially tested system is as bad as an untested system. And the price of an untested & under tested system is high.

The implementation is the final & important phase. It involves user-training, system testing in order to ensure successful running of the proposed system. The user tests the system & changes are made according to their needs. The testing involves the testing of the developed system using various kinds of data. While testin, errors are oted& correctness is the mode.

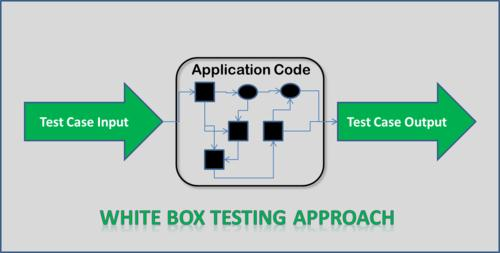
There are different types of testing methods are present. These are:-

* Static Testing
* Dynamic Testing.
* White-Box Testing
* Black-Box Testing
* Visual Testing
* Grey-Box Testing.

**6.1.1 White box Testing:**

White-box testing (also known as clear box testing, glass box testing, transparent box testing, and structural testing) is a method of testing [software](https://en.wikipedia.org/wiki/Software) that tests internal structures or workings of an application, as opposed to its functionality (i.e. [black-box testing)](https://en.wikipedia.org/wiki/Black-box_testing). In white-box testing an internal perspective of the system, as well as programming skills, are used to design test cases. The tester chooses inputs to exercise paths through the code and

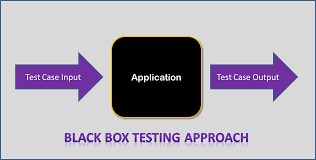
determine the expected outputs. Though this method of test design can uncover many errors or problems, it has the potential to miss unimplemented parts of the specification or missing requirements.



## Figure 6.1: White Box Testing

**6.1.2 Black Box Testing:**

Black box testing is a method of software testing that examines the functionality of an application without peering into its internal structures & workings. This method of test can be applied virtually to every level of software testing. It is sometimes referred to as specification-based testing. Specific knowledge of the application's code, internal structure and programming knowledge in general is not required.[[2]](https://en.wikipedia.org/wiki/Black-box_testing#cite_note-2) The tester is aware of *what* the software is supposed to do but is not aware of *how* it does it. For instance, the tester is aware that a particular input returns a certain, invariable output but is not aware of *how* the software produces the output in the first place.



## Figure 6.2: Black Box Testing

**6.2 Test Cases in Product Billing System:**

There are different types of test cases are present in the Product Billing system projects. These are:-

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Test Case** | **Result** | **Status** |
| 1 | Incorrect username or Password | Sign in page will not open | Pass |
| 2 | Correct username & password | Sign in page will be open | Pass |
| 3 | Data Stored in the  Database | Data show in database | Pass |
| 4 | Vendor not Found in Database | Can’t add Product | Pass |
| 5 | Details are all filled | Data is added into database | Pass |
| 6 | Details are not completely filled | Shows error | Pass |
| 7 | When Customer is not selected while creating invoice | Shows error | Pass |
| 8 | When no product is not selected while creating invoice | Shows error | Pass |
| 9 | When all details are filled while creating invoice. | Generates Invoice and also stores data in database. | Pass |

**Table 6.1: Test Cases in Product Billing Software**

# CHAPTER-7 OUTPUT SCREENS

**7.1 Output Screens:**

This chapter will include the output screens of the Product Billing System. These output screens include different frames of the software include about us login frame, forgot password, main software etc. It is very useful for understanding the project.

Figure

7.1

Splash Screen





Figure 7.2 Login Frame



Figure 7.3 Forgot Password Frame

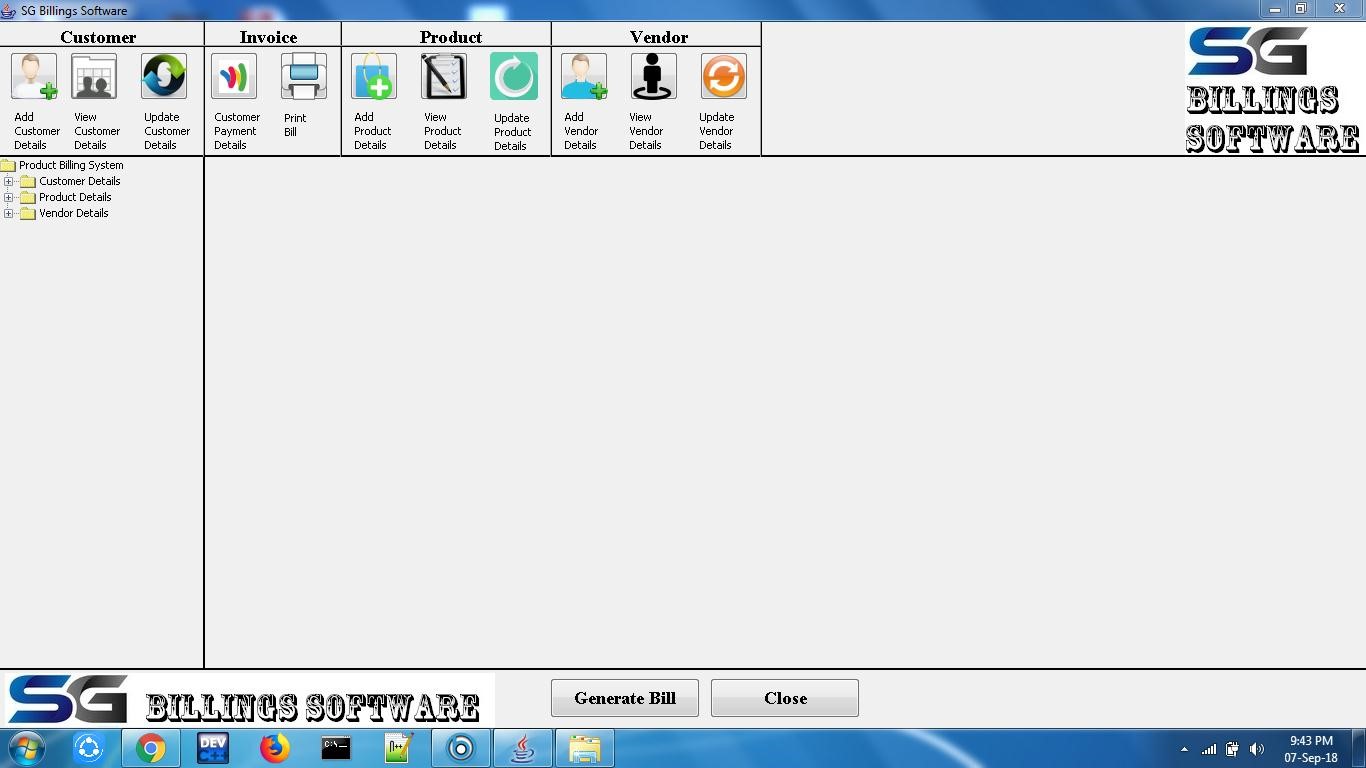


Figure 7.4 Main Software Frame

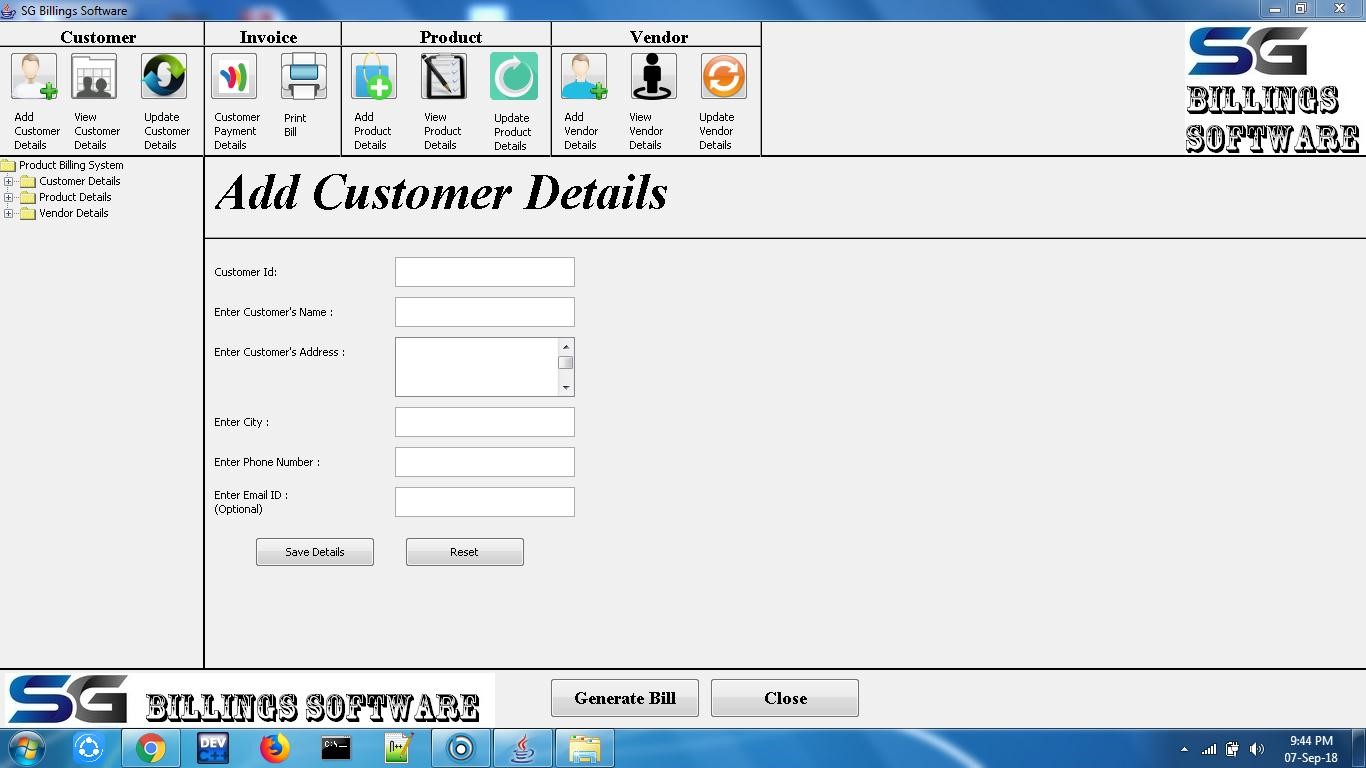


Figure 7.5 Add Customer Frame

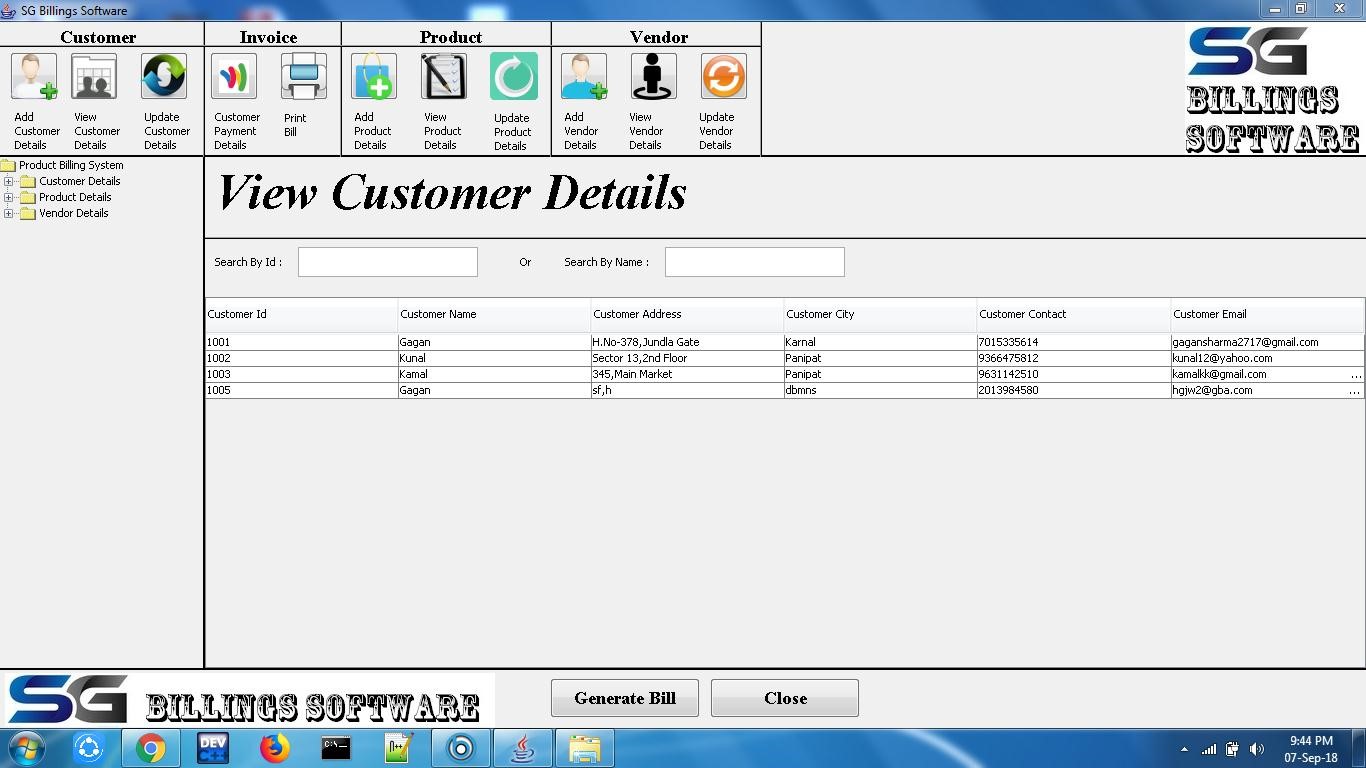


Figure 7.6 View Customer Frame

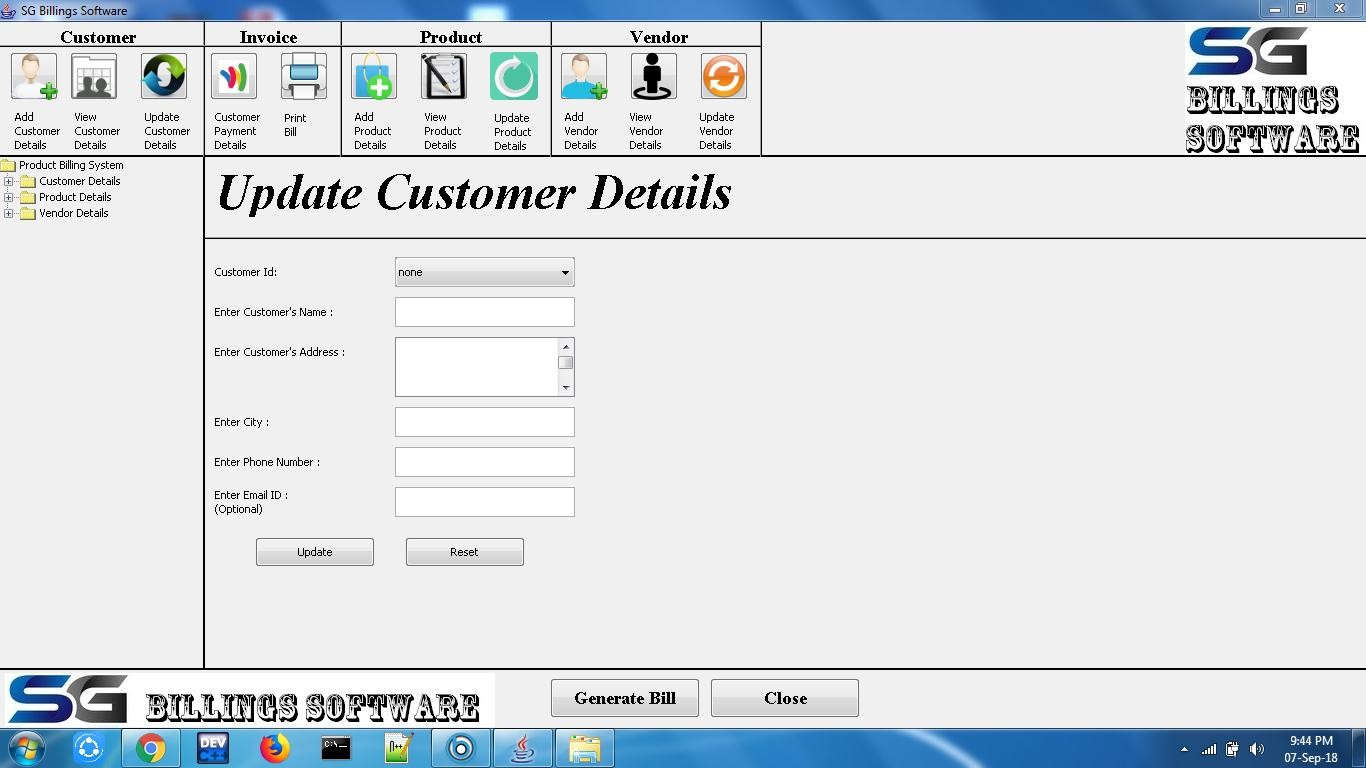


Figure 7.7 Update Customer Frame

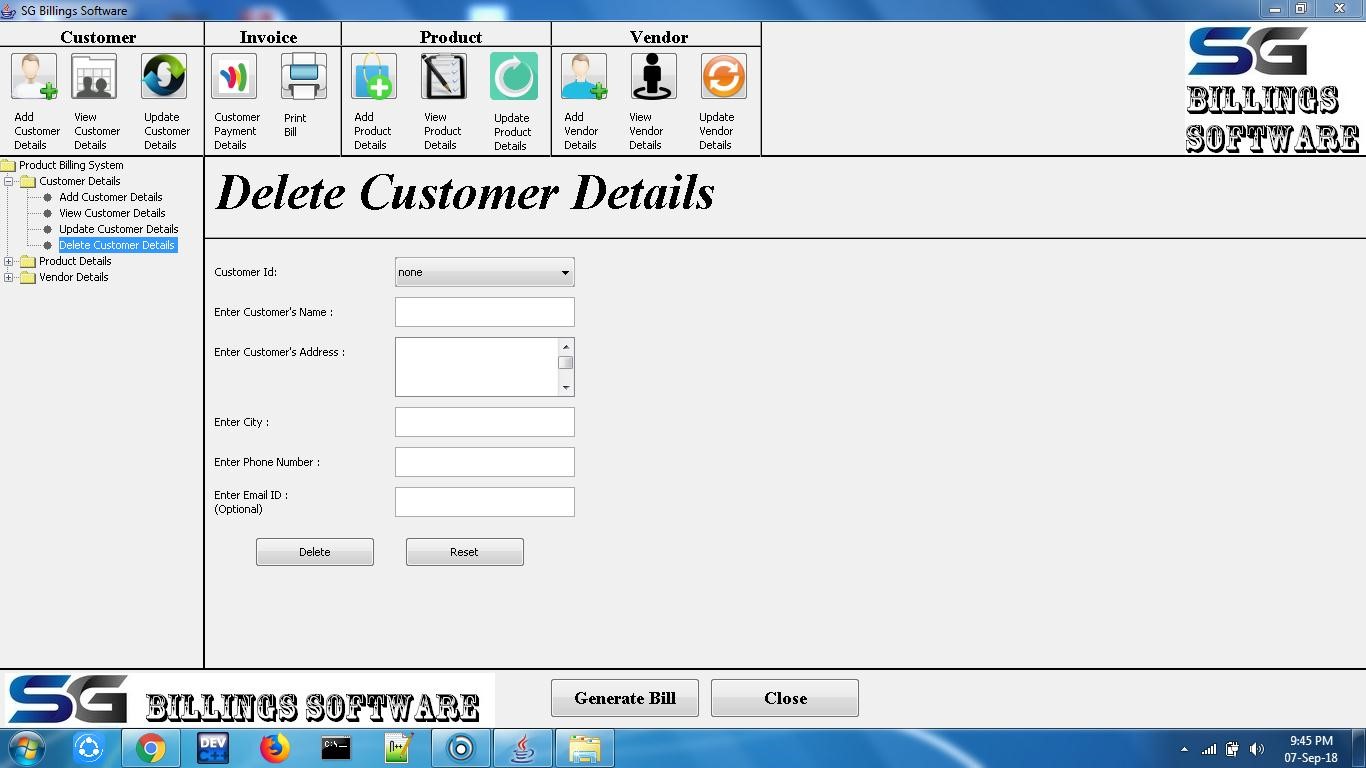


Figure 7.8 Delete Customer Frame

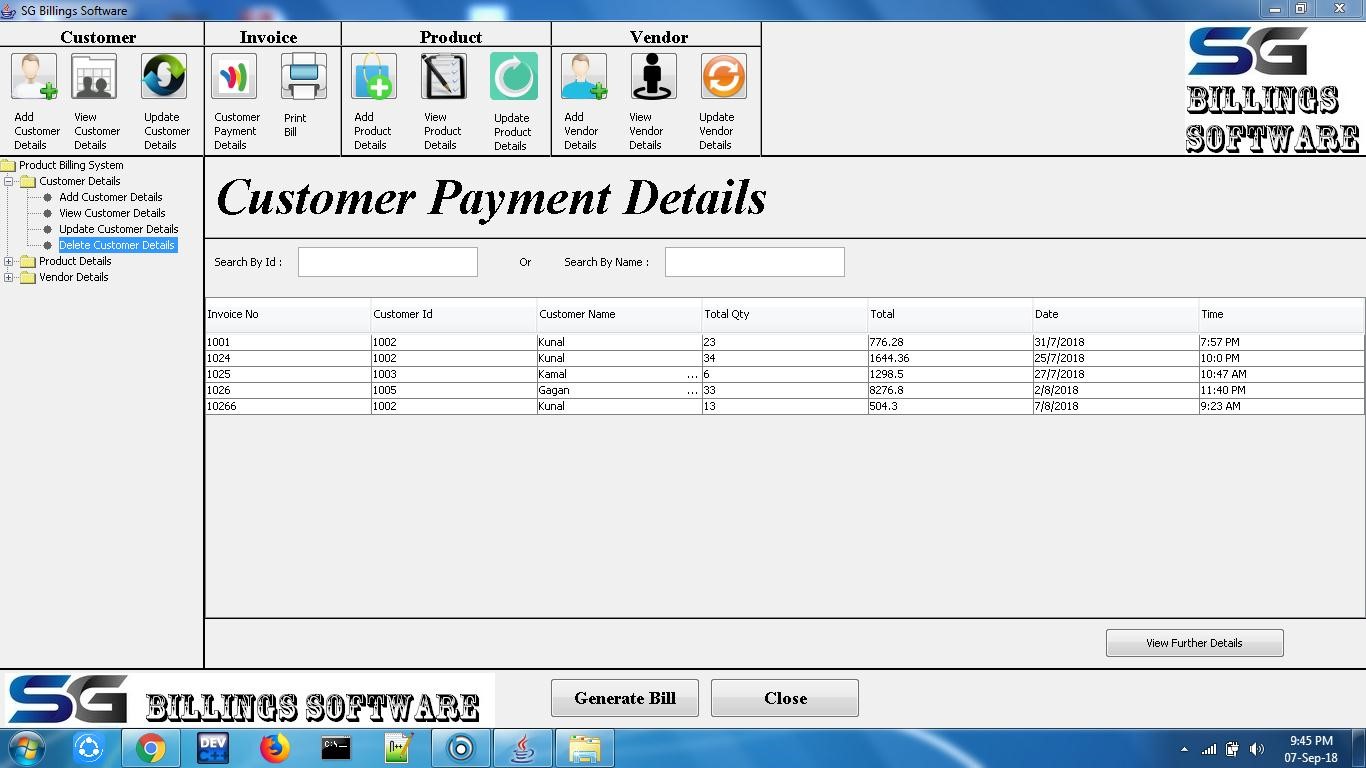


Figure 7.9 Customer Payment Details Frame

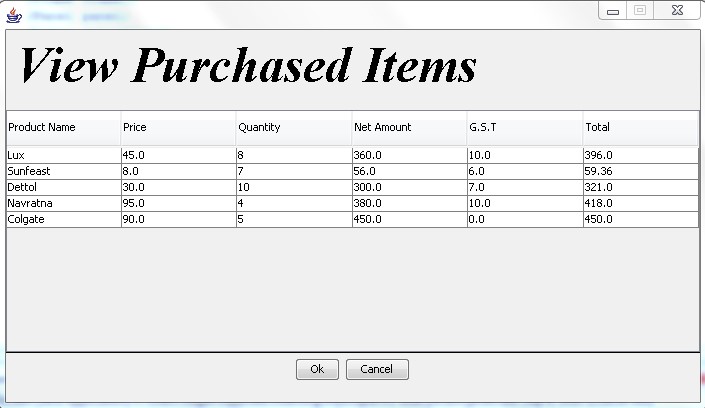


Figure 7.10 View Purchased Items Frame

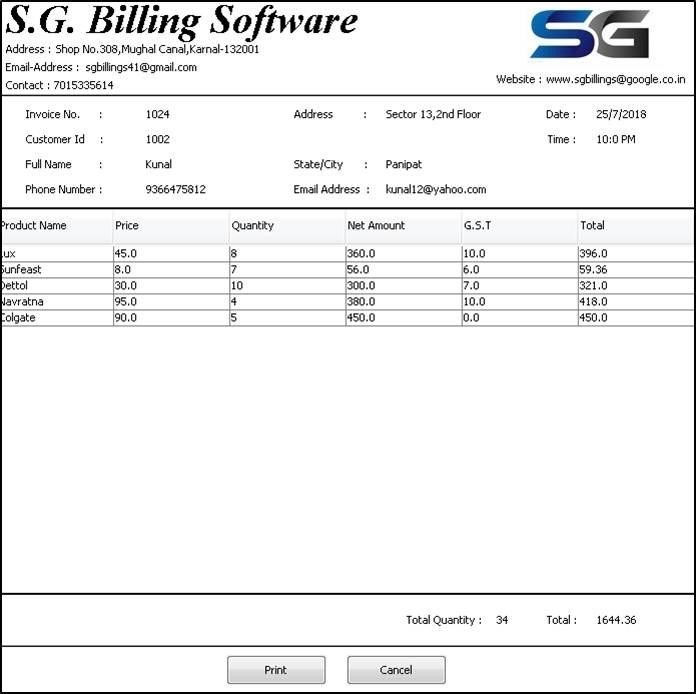


Figure 7.11 Invoice Frame

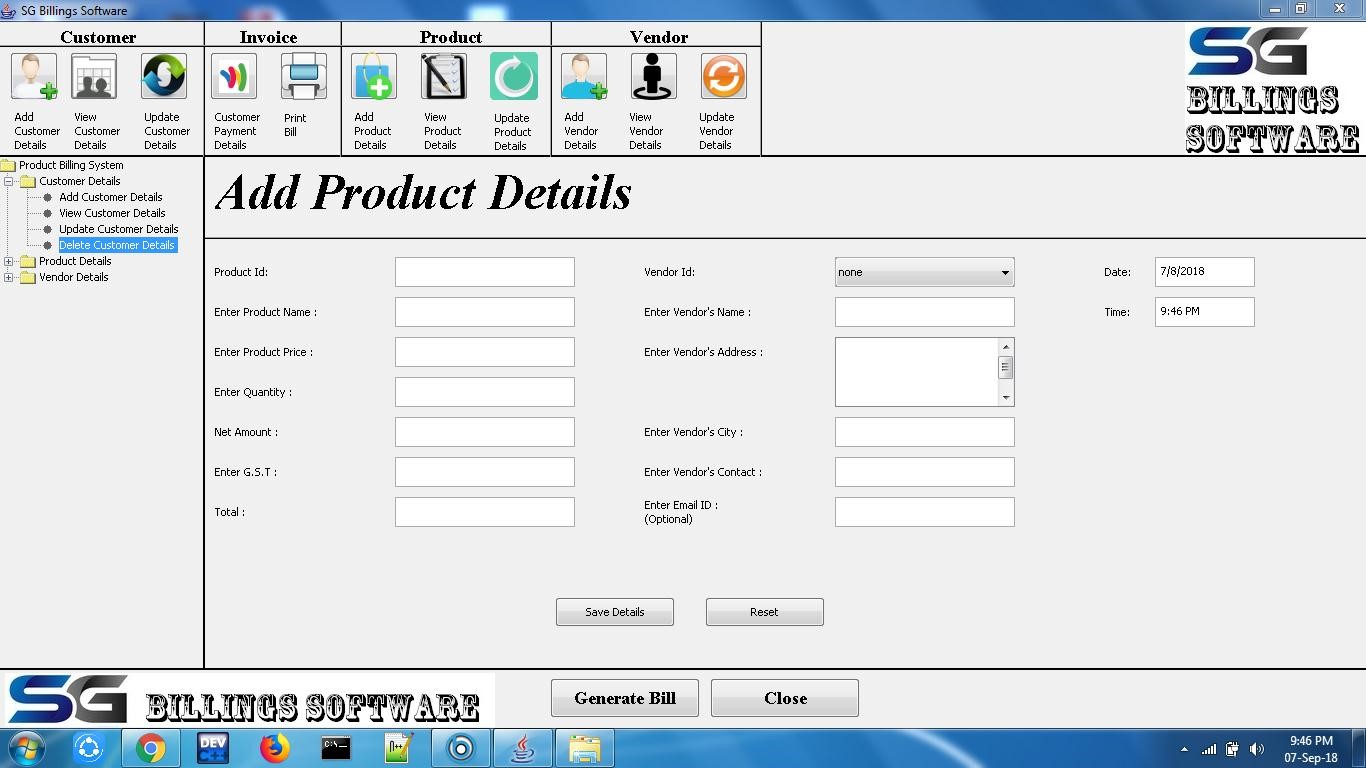


Figure 7.12 Add Product Frame

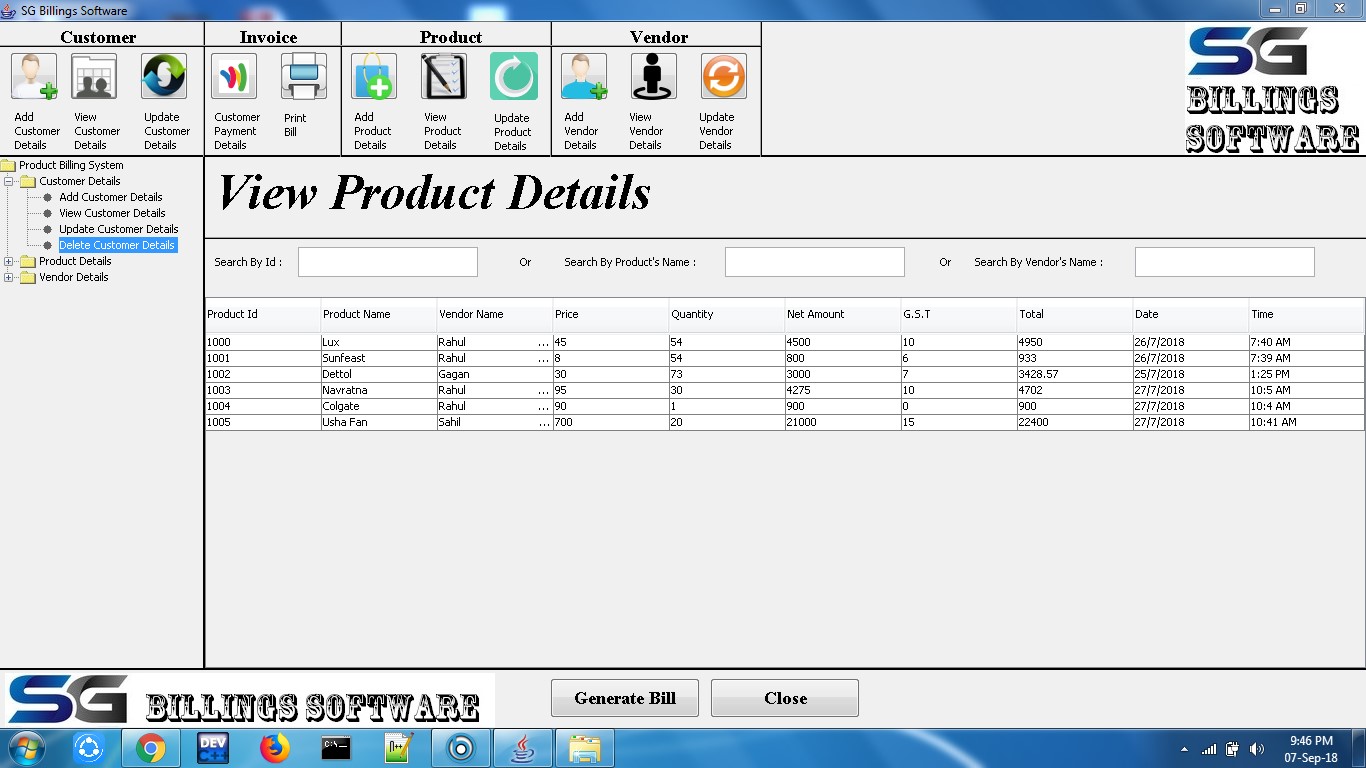


Figure 7.13 View Product Frame

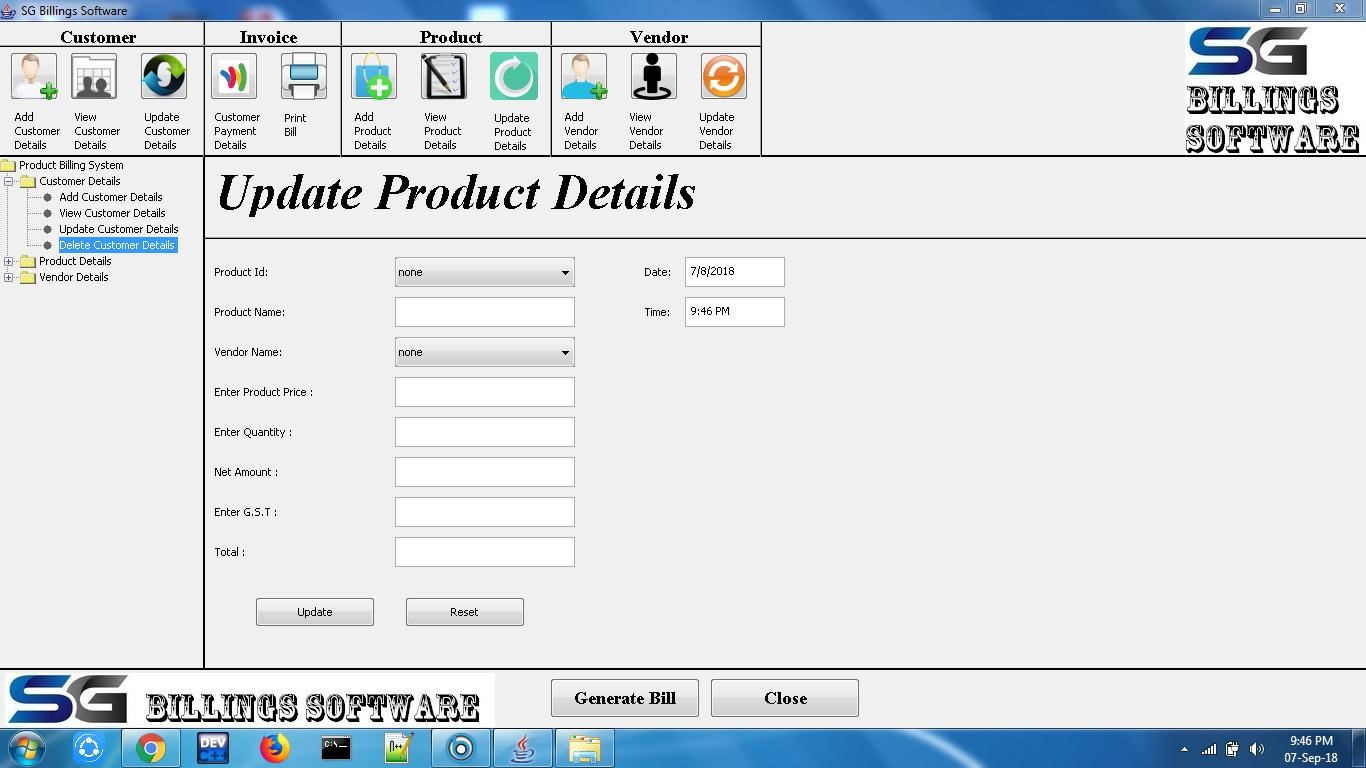


Figure 7.14 Update Product Frame

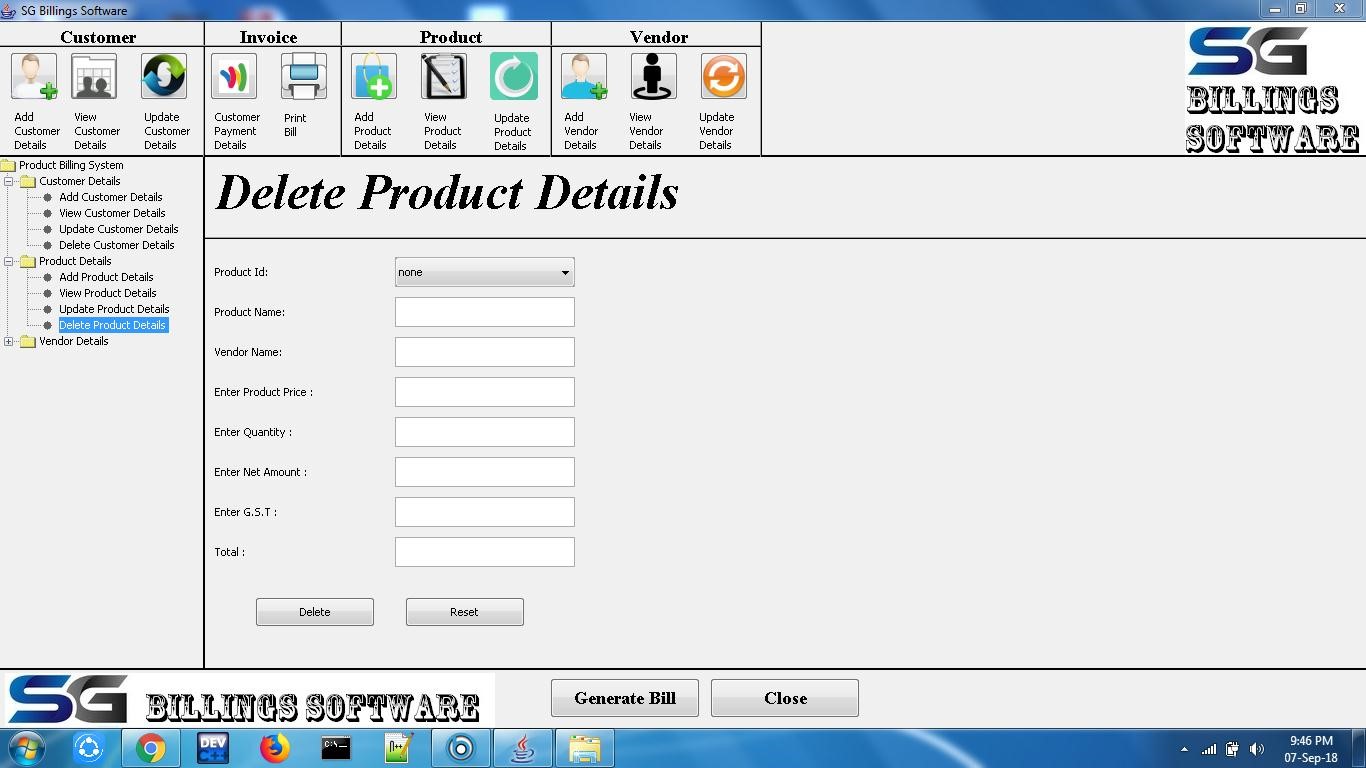


Figure 7.15 Delete Product Frame

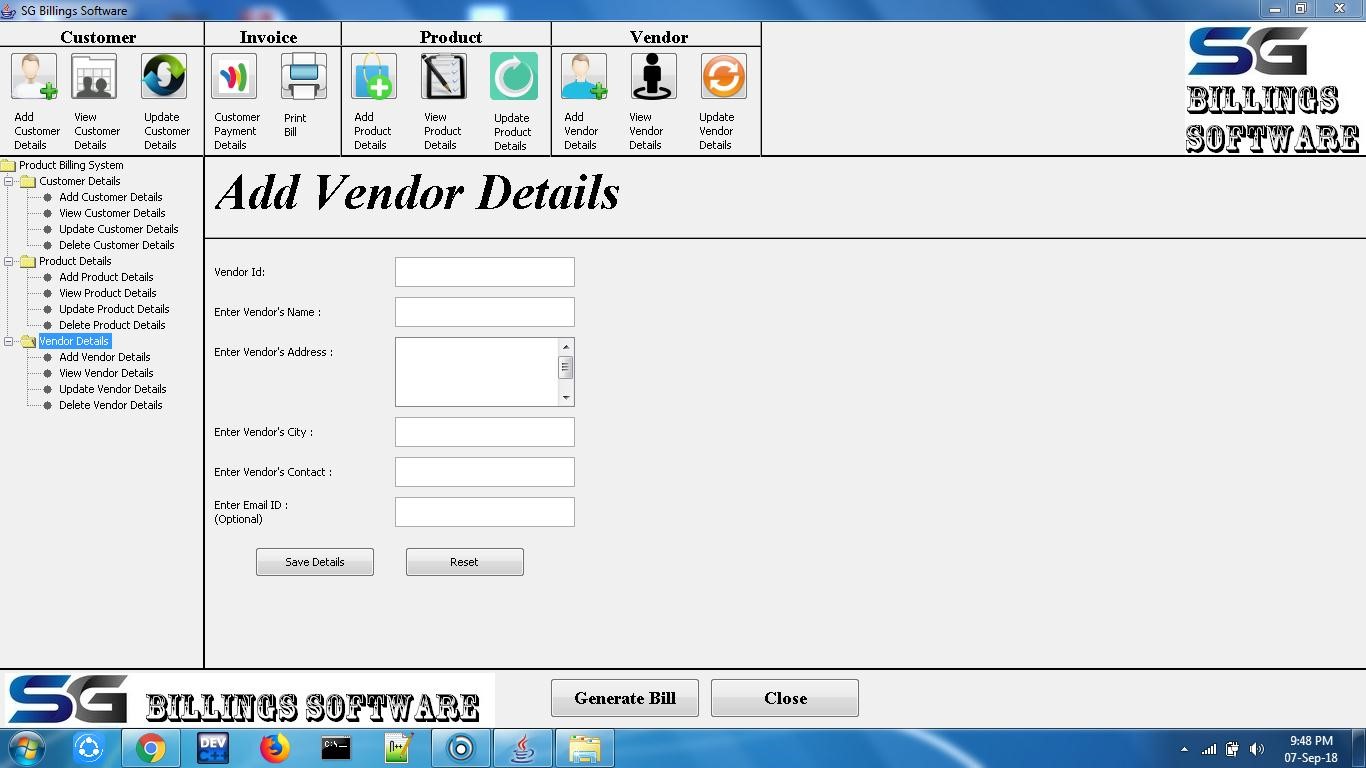


Figure 7.16 Add Vendor Frame

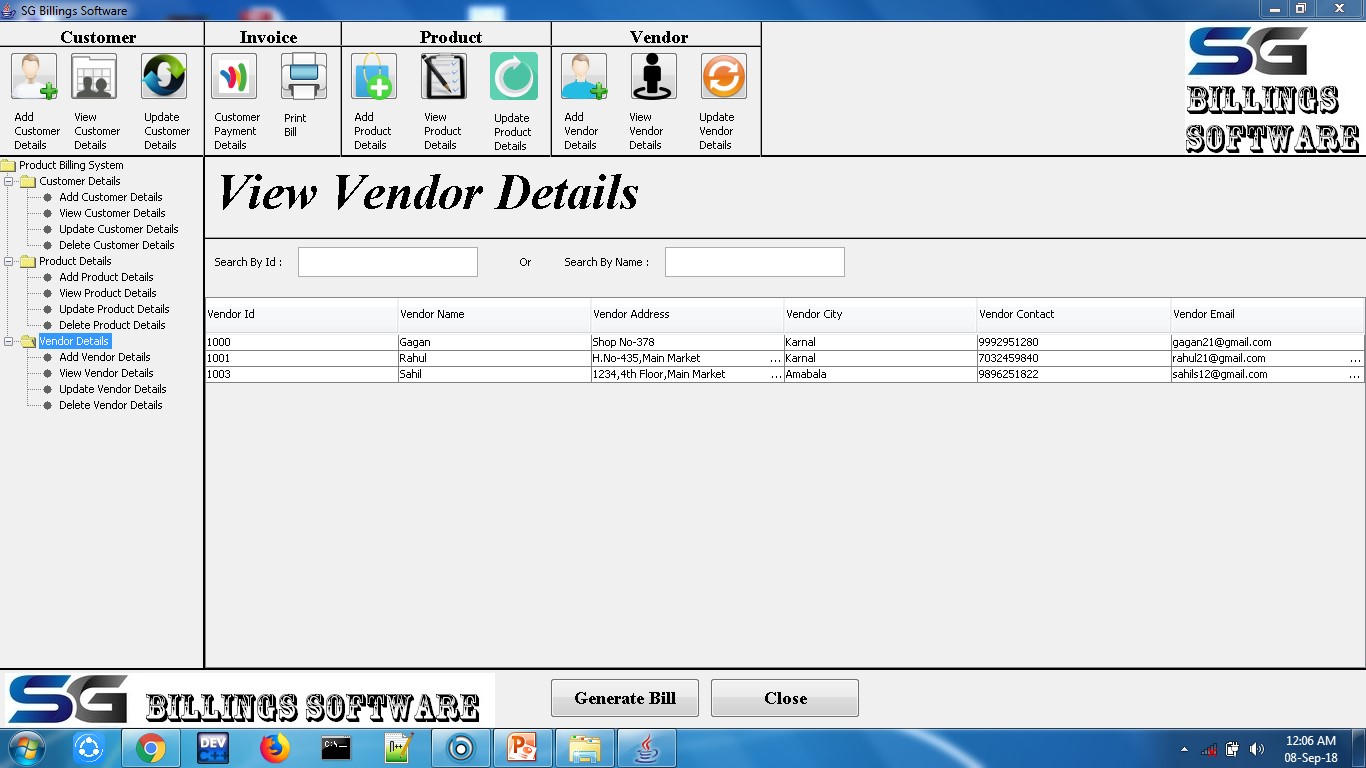


Figure 7.17 View Vendor Frame

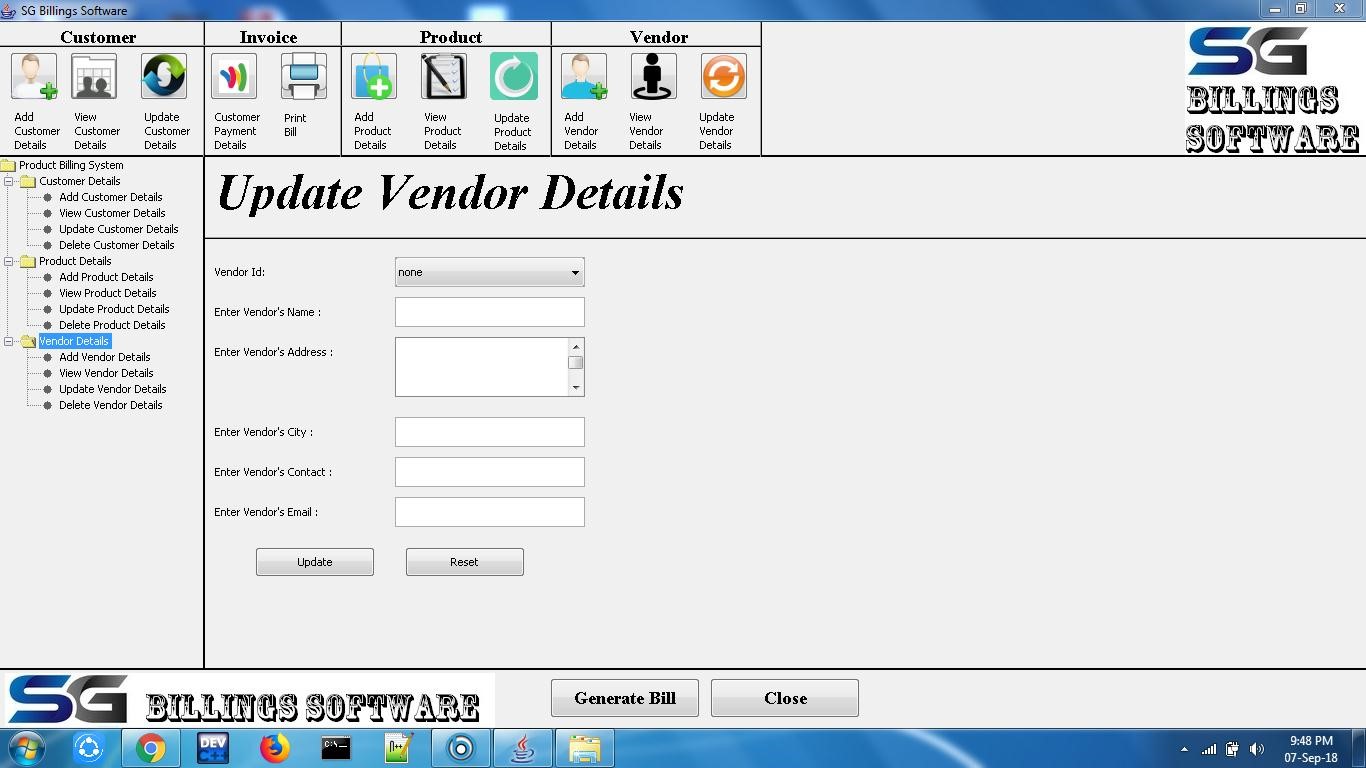


Figure 7.18 Update Vendor Frame

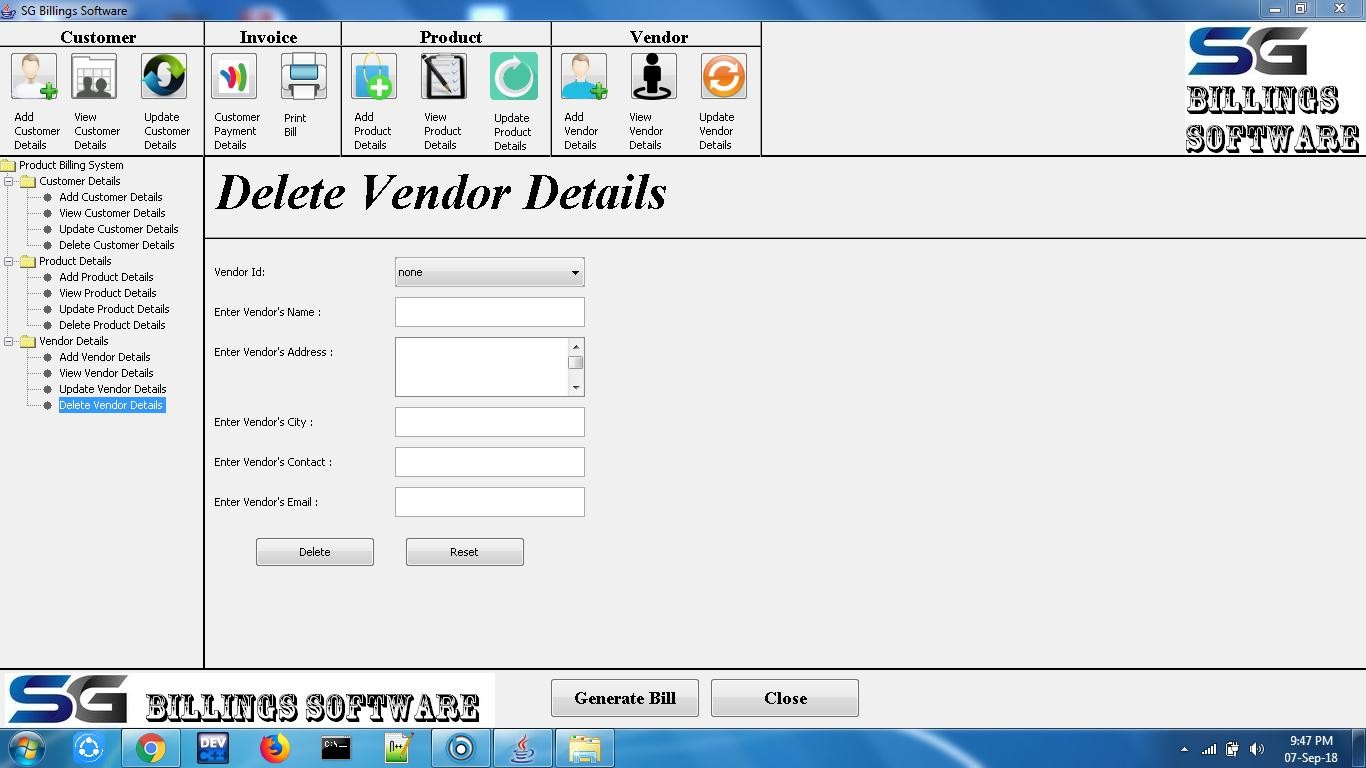


Figure 7.19 Delete Vendor Frame

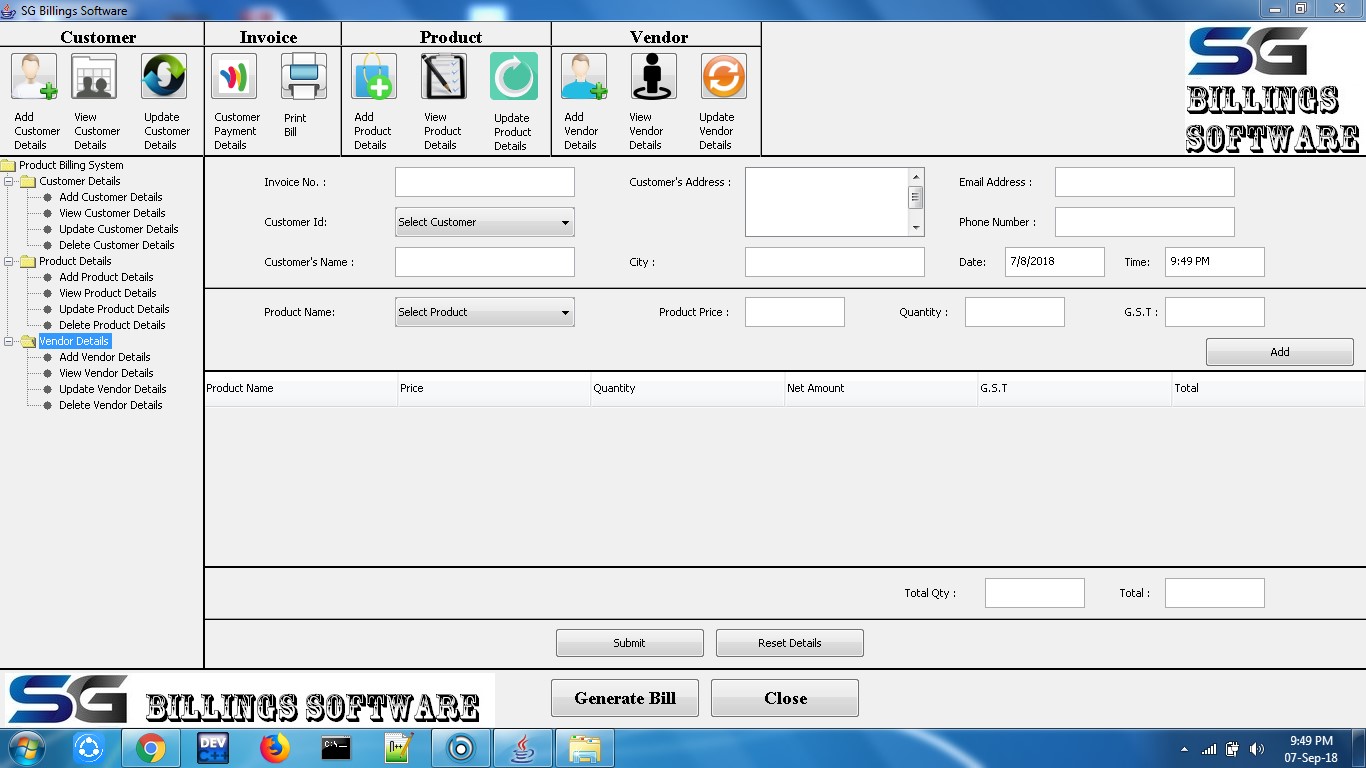


Figure 7.20 Generate Bill Frame

# CHAPTER-8 CONCLUSIONS

**8.1 Conclusion:**

The project titled as “Product billing system” is a web based application. This software provides facility for create, update and delete accountants details after login . it can search branch wise accountant. And also search all candidates studying in the various branches and can update and delete them. The software is developed with modular approach. All modules in the system have been tested with valid data and invalid data and everything work successfully. Thus the system has fulfilled all the objectives identified and is able to replace the existing system.

The project has been completed successfully with the maximum satisfaction of the organization. The constraints are met and overcome successfully. The system is designed as like it was decided in the design phase. The project gives good idea on developing a full-fledged application satisfying the user requirements.

The system is very flexible and versatile. This software has a user-friendly screen that enables the user to use without any inconvenience. Validation checks induced have greatly reduced errors. Provisions have been made to upgrade the software.

**CHAPTER-9**

# FURTHER ENHANCEMENTS/RECOMMENDATIONS

In future we can use photo reorganization instead of using heterogeneous database more over High speed, accuracy and non-redundant data are the main advantages of the proposed system. In the proposed system the user is provided with a choice of data screen, which are similar in formats to the source documents. Data entry errors can be minimized through validity checks. After the verification only the data are placed the permanent database.

The software can be developed further to include a lot of modules because the proposed system is developed on the view of future, for example we should develop the system as a database independent using JDBC so we can connect it to any other database, Now the proposed system is based on PC and intranet but in the future if we need to convert it into internet then we need to change the front end only because we are developing this on the basis of OOP technology and most of the business logic’s are bounded in the class files and module like reusable components.

In future we can use biometrics to add the product details it faster the the processing of billing and we can also introduce stock management or inventory management and we also want to introduce the igst and cgst and at least we also want to introduce monthly customer and for monthly customers there is a special discount.

# CHAPTER-10

**REFERENCES/BIBLIOGRAPHY**

1. https://www.geeksforgeeks.org/jdbc-drivers/

, https://www.geeksforgeeks.org/jvm-works-jvm-architecture/

1. http://[www.javatpoint.com](http://www.javatpoint.com/)
2. <https://www.youtube.com/watch?v=gWpg3yMiL0M>
3. <https://stackoverflow.com/questions/tagged/java>
4. https://www.tutorialspoint.com/java/java\_interfaces.htm