

ITE 1942 – ICT PROJECT

PROJECT REPORT

Level 01

Restaurant Point of Sale (POS) System

Submitted by:

SENARATHNA K.P.G.R.W.

E2046073

Bachelor of Information Technology (External Degree)

Faculty of Information Technology

University of Moratuwa

Table of Contents

1.INTRODUCTION.....	3
2.RELATED WORK	5
3.SYSTEMS ANALYSIS.....	9
4. SYSTEM DESIGN.....	12
5. SYSTEM IMPLEMENTATION	12
6. APPENDIX.....	12
7. REFERENCES	12

1.Introduction

1.1. Background & Motivation

The Pizza place is a restaurant located in Hingurakgoda. The Owners are trying to increase sales and promote their business with a loyalty program. This task has become difficult with the current manual book-keeping system.

The owners have requested to design and develop a computerized Point of Sale (POS) system for their business. This POS system can process orders quickly, handle billing operations automatically, view real-time sales reports, track inventories easily with real-time indicators, and run customized loyalty programs.

1.2 Problem in brief

They currently manage their sales fully in a manual process and use books to keep records of all financial transactions and inventories. All the calculations are done by the staff themselves with a calculator. The receipts are handwritten.

These things have caused increased customer waiting, inaccurate business reports, difficulties of tracking inventory, use of more human labor, and increased cost. This manual process has made it hard to run a loyalty program.

1.3. Aims & Objective

Design and develop a user-friendly Point of Sale (POS) system for Pizza place restaurant that can manage their sales and financial records effectively and efficiently to increase sales, promote their business, and build a loyal customer base.

1.4 Summary

The Pizza place owners are trying to move from their current manual book-keeping system to a computerized Point of Sale (POS) system with a loyalty program. From this implementation they expect to increase their business productivity and growth. In the next chapter of this report discusses about related works to this project.

2.Related Works

2.1. Manual restaurant bookkeeping system

Manual Bookkeeping is a system for managing and recording business activity transactions. All the transactions are written and maintained by hand. Calculations are processed using calculators without any specialized software. The staff handles orders, sales, inventories, and billing themselves manually.



Figure 1- Manual bookkeeping. [1]

➤ Features

- Manage sales, purchases, and orders.
- Calculate sales, cost, and profit/loss.
- Inventory management.
- Billing.
- Invoicing.
- Reporting.

➤ Missing features

- Tracking of real-time inventory, sales, purchases, and orders
- Customer management.
- Discount and customer loyalty management.
- Low stock indications.

2.2. IncoPOS

IncoPOS is All-in-One touch POS and Inventory Management system for Restaurants. Flexible and full functional software for Food & Beverage retail that adapts to the way you run your business. No matter small independent restaurant, or a big chain, whether it is table service, quick service, or a fast casual concept, IncoPOS provides the functionality your business desires. The interface is intuitive and easy to use, which guarantees quick and seamless operations in big stores with heavy client flow and thousands of transactions.[2]

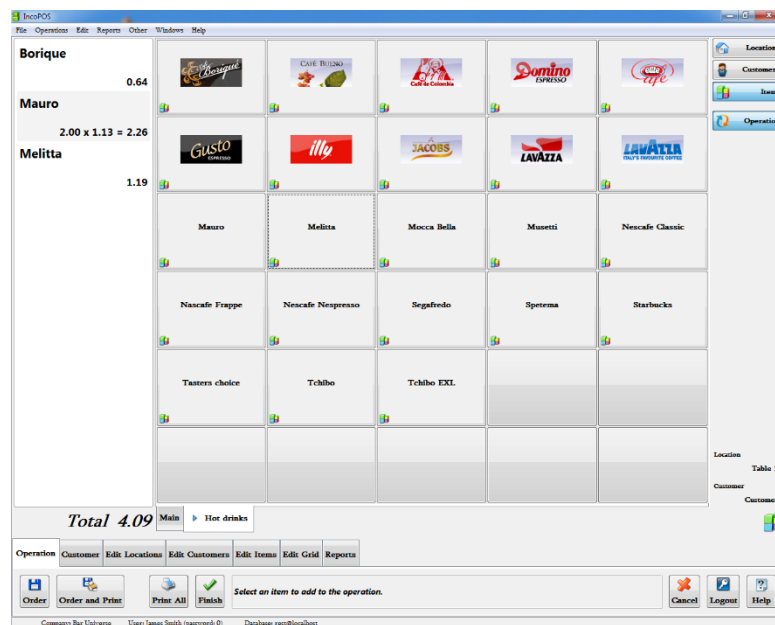


Figure 2 - IncoPOS GUI [3]

- Features
 - Track inventory, sales, purchases, purchase orders.
 - Real time inventory management.
 - Print and customize documents and cash receipts.
 - Create custom reports and visualize the results graphically.
 - Customer Management.
- Missing features
 - Low stock indications.
 - Calculate cost and profit /loss.

2.3. GoFrugal

GoFrugal is a comprehensive point of sale (POS) system which provides billing software for retail business, POS for restaurants, and distribution management solutions. The POS module has a suite of great features which include touch POS, inventory/recipe management, smart reports, integrations, CRM & loyalty, and multistore management. [4]



Figure 3 - GoFrugal GUI [5]

➤ Features

- Track inventory, sales, purchases, purchase orders.
- Multistore management.
- Fast billing.
- Smart Reports.
- Invoicing directly from POS.
- Table Management.
- Manage recipes and inventory.

➤ Missing features

- Low stock indications.
- Calculate cost and profit /loss.

2.4. Summary table

System Name	Features
1. Manual restaurant bookkeeping system.	<ul style="list-style-type: none"> • Uses stationary. • Increased risk of error. • Errors can be harder to track. • Uses more physical storage space. • Time consuming. • No backup records in case of loss or damage. • Require specialized knowledge to maintain. • High data redundancy. • Uses more human labor. • Cost is high.
2. IncoPOS.	<ul style="list-style-type: none"> • Uses hardware such as Computer, receipt printer, Touch screen monitor, and barcode scanner. • Least time consuming than other systems. • Has cloud-based backup records in case of loss or damage. • Minimal data redundancy. • Do not require specialized knowledge to maintain. • Most user-friendly interface among other these systems. • One person can manage the system. • Cost is higher than other two systems.
3. GoFrugal.	<ul style="list-style-type: none"> • Uses hardware such as Computer, receipt printer, monitor, and barcode scanner. • Less time consuming. • Has cloud-based backup records in case of loss or damage. • Low data redundancy. • Do not require specialized knowledge to maintain. • User friendly interface. • One person can manage the system. • Cost is average comparing to other two systems.

Table 1 - Summary table.

3. Systems Analysis

3.1. System requirements

The proposed computerized Point of Sale (POS) system should be able to handle and manage customers, All the business functions and data.

3.2. Functional requirements

3.1.1. Manage customers

- Register new customers.
- Issue customer Id.
- Update customer details.
- Delete customers.

3.1.2. Manage inventory

- Register new products.
- Update product details.
- Delete products.
- Low stock indication.
- Update unsold products.

3.1.3. Manage sales

- Place orders, add loyalty discounts, billing, and process payments.
- View sales records.
- View unsold product records.
- View customer details and their loyalty point count.
- View products according to popularity.
- Calculate sales, cost, and profit/loss.

3.2. Non-functional requirements

3.2.1. Usability

- The POS system should be easy to learn and operate for the user.

3.2.2. Efficiency of use

- An experienced user should be able to accomplish tasks from the system quickly.

3.2.3. Security

- The POS system should be protected from unauthorized access to the system and its stored data.

3.2.4. Intuitiveness

- The POS system should be simple to understand (interface, buttons, headings, etc.)

3.2.5. Reliability

- The POS system and its functions should operate consistently without failure.

3.2.6. Integrity

- Data maintained by the POS system should be accurate, authentic, and without corruption.

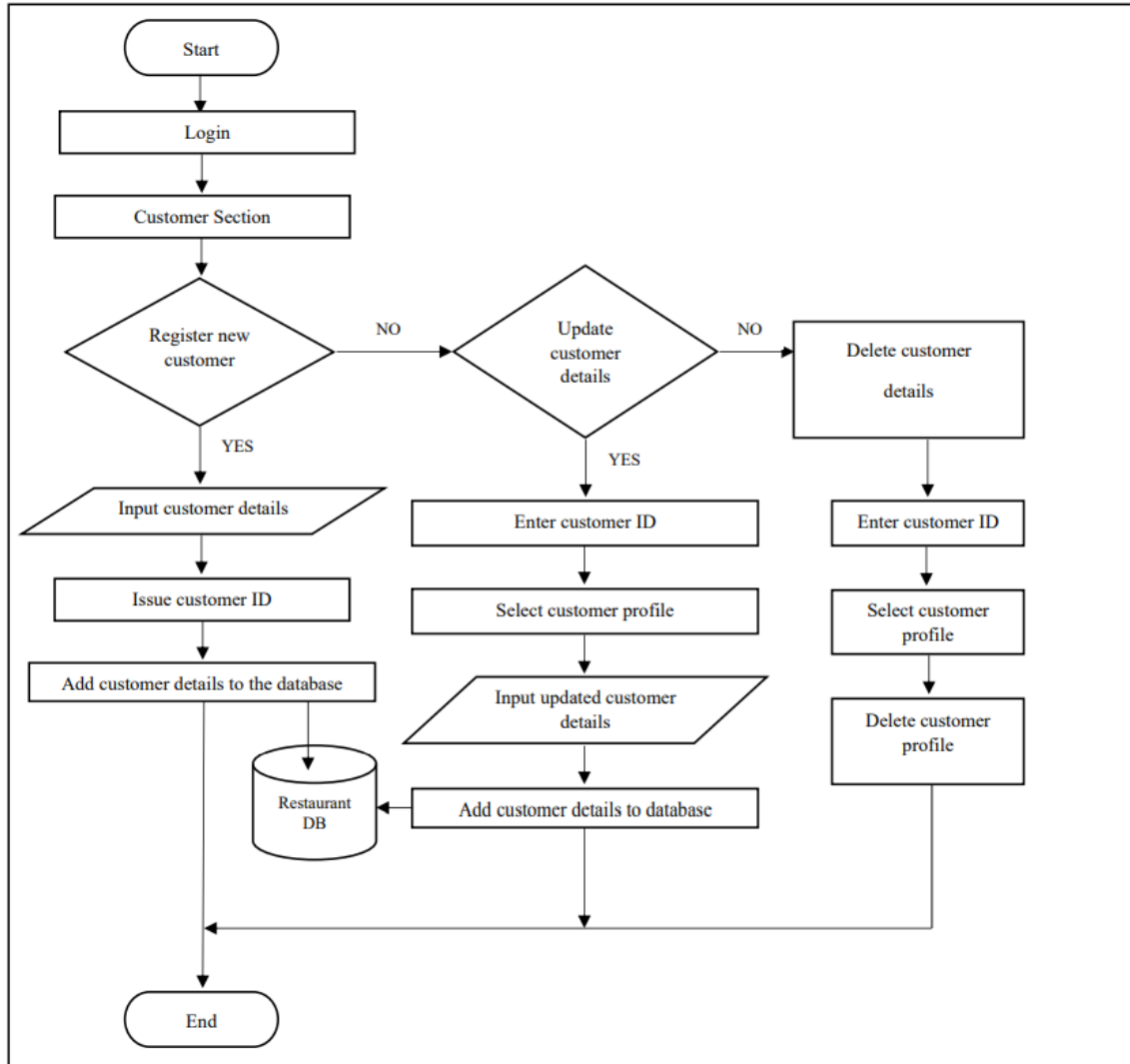
3.4 Summary

The above-mentioned functional requirements must implement to enable stakeholders to accomplish their required tasks. The non-functional requirements also necessary to improve system attributes such as security, reliability, performance, maintainability, intuitiveness, and usability. In the next chapter of this report discusses about design of proposed computerized Point of Sale (POS) system.

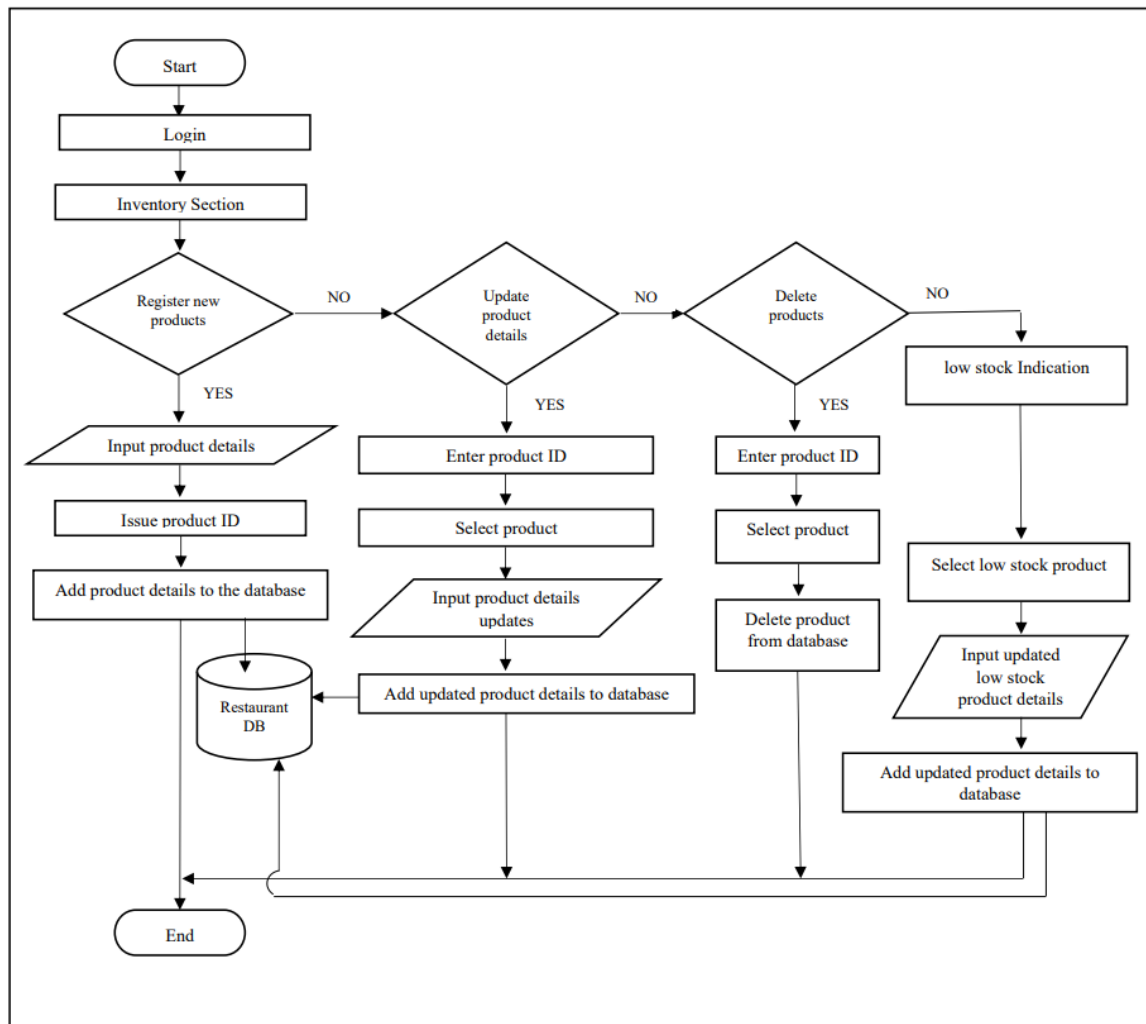
4. System Design

4.1 Flowcharts

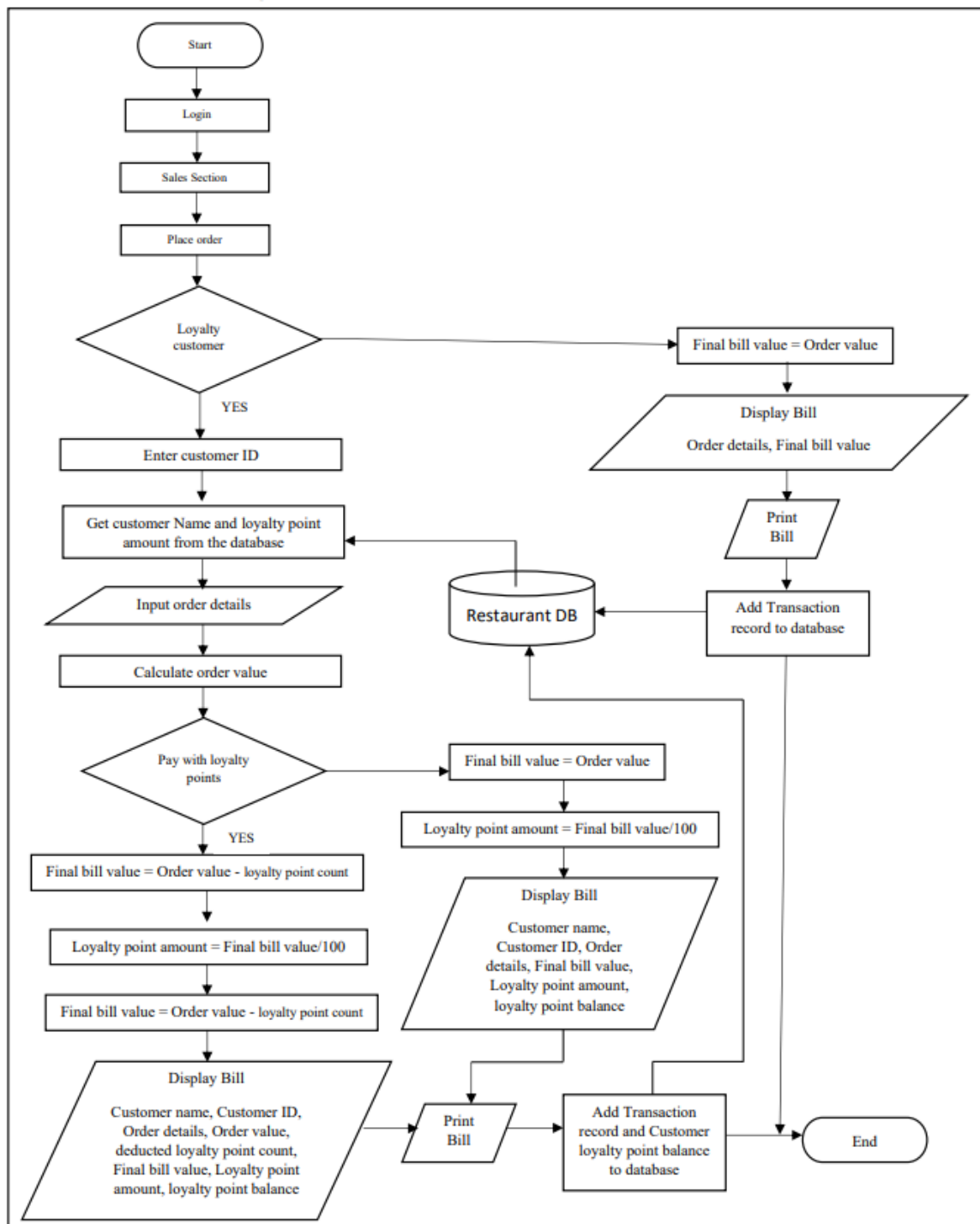
4.4.1. Manage customers



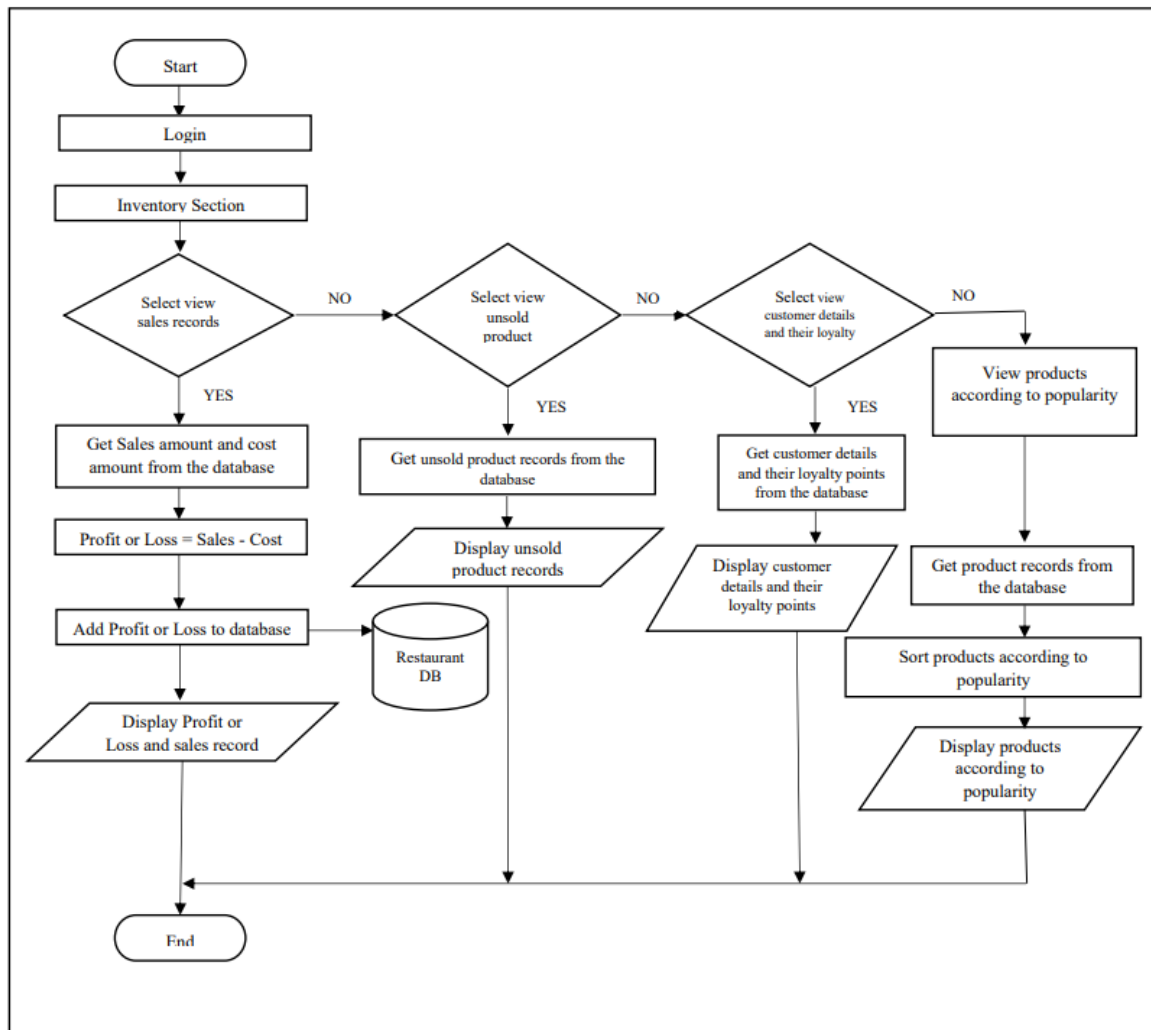
3.3.2. Manage inventory



3.3.3. Manage sales



3.3.4. Manage sales records



3.4. Pseudocode

3.4.1. Manage customers

Begin

Login to the system

Go to customer section

If Register new customer selected **then**

 Input customer details

 Issue customer ID

 Add customer details to the database

Else if Update customer details selected **then**

Enter customer ID
Select customer profile
Input updated customer details
Add customer details to database

Else Delete customer details selected then

Enter customer ID
Select customer profile
Delete customer profile

End if

End if

End

3.4.2. Manage inventory

Begin

Login to the system

Go to Inventory section

If Register new products selected then

Input product details
Issue product ID
Add product details to the database

Else if Update product details selected then

Enter product ID
Select product
Input product detail updates
Add updated product details to database

Else if Delete products selected then

Enter product ID
Select product
Delete product from database

Else low stock Indication selected then

Select low stock product

Input updated low stock product details

Add updated product details to database

End if

End if

End if

End

3.4.3. Manage sales

Begin

Login to the system

Go to Sales section

Select Place order

If Customer is a Loyalty member then

Enter customer ID

Get customer Name and loyalty point amount from the database

Input order details

Calculate order value

If Customer Pay with loyalty points then

Final bill value = Order value - loyalty point count

Loyalty point amount = Final bill value/100

Final bill value = Order value - loyalty point count

Display Bill Including Customer name, Customer ID, Order details, Order value, deducted loyalty point count, Final bill value, Loyalty point amount, loyalty point balance

Print Bill

Add Transaction record and Customer loyalty point balance to database

Else Customer Pay without loyalty points then

Final bill value = Order value

Loyalty point amount = Final bill value/100

Delete product from database

Display Bill including Customer name, Customer ID, Order details, Final bill value, Loyalty point amount, loyalty point balance

Print Bill

Add Transaction record and Customer loyalty point balance to database

Else Customer is not a Loyalty member **then**

Final bill value = Order value

Display Bill including Order details, Final bill value

Print Bill

Add Transaction record to database

End if

End if

End

3.4.4. Manage sales records

Begin

Login to the system

Go to Inventory section

If view sales records selected **then**

Get Sales amount and cost amount from the database

Profit or Loss = Sales – Cost

Display Profit or Loss and sales record

Else if view unsold product selected **then**

Get unsold product records from the database

Display unsold product records

Else if view customer details and their loyalty selected **then**

Get customer details and their loyalty points from the database

Display customer details and their loyalty points

Else View products according to popularity selected **then**

Get product records from the database

Sort products according to popularity

Display products according to popularity

End if

End if

End if

End


4.3 Summary

The flow chart and pseudocode can be implemented to the system as above demonstrated. In the next chapter of this report discusses about system implementation of proposed computerized Point of Sale (POS) system.

5. System Implementation

5.1. Login Interface

Login



Welcome to
Pizza Place


Where taste meets the myth

Username

admin

Password


Login



This is the login interface of the system. You can login to the system by entering username and the password. The default username is “admin” and password is “1234”.

5.2. Dashboard


Dashboard




Welcome to
Pizza Place

Where taste meets the myth


LOGOUT




Customer
Registration



POS



Inventory




Reports

This is the Dashboard interface of the system. You can navigate to Customer Registration, POS, Inventory and Reports sections of the system through the dashboard.

5.3. Customer Registration

Customer Registration



Pizza Place
Where taste meets the myth

Dashboard

Customer Name:


Wajira

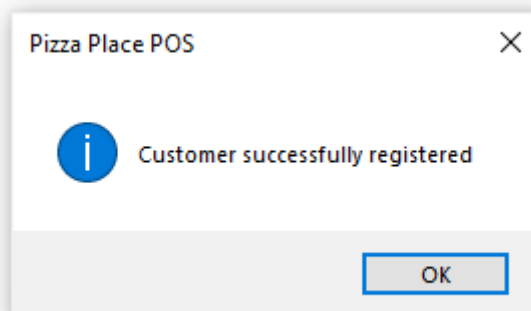
Customer ID:

Contact No:

0774510001

Register






This is the customer registration interface of the system. you can register new customer to the system through this section. customer ID will be automatically generated from the system.

5.4. Product registration (Inventory)

Inventory



Pizza Place
Where taste meets the myth


Dashboard

Name: Italian Pizza


Price(Rs.): 1250

Product Quantity: 16

Add



Pizza Place POS


 Product successfully registered

OK

This is the Product registration (Inventory) interface of the system. you can register new products to the system through this section.

5.5. Cashier Form (Sales Management)

Cashier



Welcome to
Pizza Place

05-24-2022 09:35 PM

where taste meets the myth

Item	Unit Price	Quantity	Total Price
Peperoni Pizza	2500	2	5000

Dashboard

Customer Type: Loyalty

Customer ID: 1

Customer Name: Sachin

Loyalty Points: 108

Select Product: Peperoni Pizza

Product Price(Rs.): 2500

Available Product Quantity: 12

Product Quantity: 2

Add to cart

Current Bill Amount(Rs.): 5000

Loyalty Discount(Rs.): 108

Sale Profit(Rs.): 392


Final Bill Amount(Rs.): 4892

Cash(Rs.): 5000

New

Pay Bill

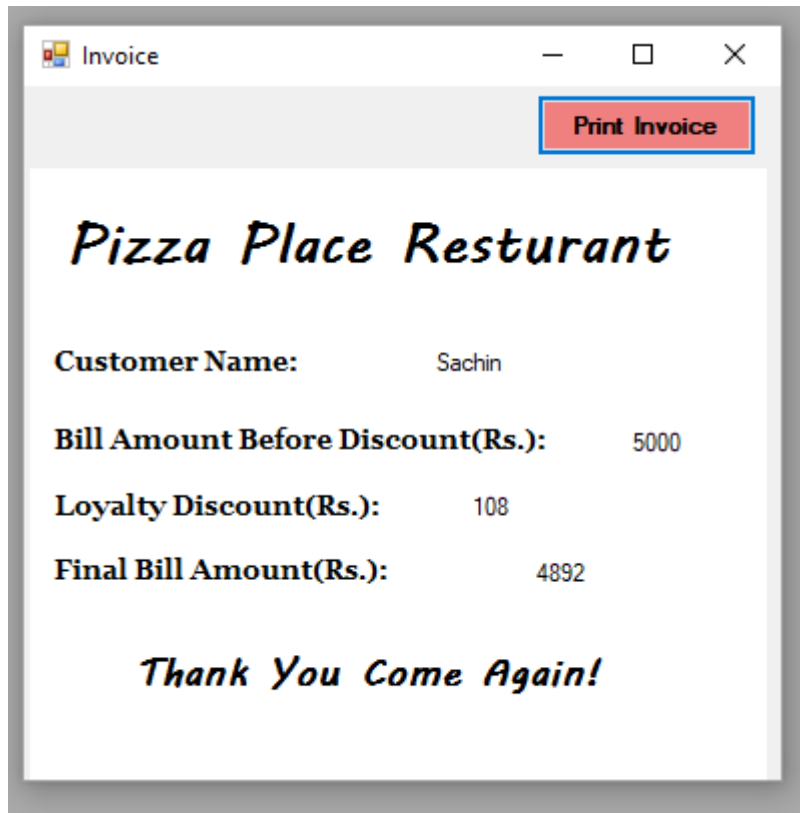
Pizza Place POS

 Transaction successfull, Change for Customer is Rs. 108

OK

This is the POS(Cashier) interface of the system. You can add products to the cart, complete bill transaction from this section.

5.6. Bill Printing



The screenshot shows a window titled 'Invoice' with standard Windows window controls (minimize, maximize, close). A red 'Print Invoice' button is located in the top right corner. The main content area displays the restaurant name 'Pizza Place Restaurant' in a large, bold, italicized font. Below this, the customer name 'Sachin' is listed. The bill details are as follows:


Customer Name:	Sachin
Bill Amount Before Discount(Rs.):	5000
Loyalty Discount(Rs.):	108
Final Bill Amount(Rs.):	4892

At the bottom, a message 'Thank You Come Again!' is displayed in a bold, italicized font.

This is the bill Printing interface of the system. You can print the invoice of the transaction from this section.

5.7. Customer Management

Reports

*Where taste meets the myth*

Welcome to
Pizza Place

Dashboard

Customers

Products

Sales Records

Customer ID:

1

Customer Name:

Sachin

Contact No:

0778856212

Loyalty Points:

0

Update


Delete

	CustomerID	CustomerName	ContactNo	LoyaltyPoints
▶	1	Sachin	0778856212	0
	3	Ravindu	0774510001	0
	4	Shan	0771234567	112
	5	Vishwa	0781234567	0
	6	Wishwa	0771237894	0
	7	Wajira	0774510001	0

This is the Customer Management interface of the system. You can Manage customer details (Update, Delete) in the system through this section.

5.8. Products Inventory Management

Reports

*Where taste meets the myth*

Welcome to Pizza Place

Dashboard

CustomersProductsSales Records

Select Product:

Price(Rs.):

Product Quantity:


UpdateDeleteLow Stock Check

	Name	Price	ProductQuantity
▶	Peperoni Pizza	2500	10
	Mushroom Pizza	4000	12
	Sicilian Pizza	3500	9
	Veggie Pizza	3500	22
	Italian Pizza	1250	16
	Tomato Pizza	750	16

This is the Product Inventory Management interface of the system. You can Manage Product details (Update, Delete) in the system through this section.

5.9. Sales Records

Reports



Where taste meets the myth

Welcome to
Pizza Place

Dashboard

Customers

Products

Sales Records

	Time	BillAmount	Discount	FinalBillAmount	SaleProfit
▶	04-18-2022 04:46 PM	15500	0	15500	1550
	04-18-2022 04:57 PM	5000	0	5000	500
	04-18-2022 05:06 PM	5000	102	5000	500
	05-18-2022 09:07 PM	8000	120	7880	788
	05-18-2022 09:45 PM	2250	112	2138	213
	05-24-2022 09:38 PM	5000	108	4892	489.2
	05-22-2022 09:09 PM	2250	105	2145	214
	05-24-2022 02:07 PM	9250	0	9250	925
	05-24-2022 07:41 PM	10500	0	10500	1050
	05-24-2022 08:08 PM	8000	0	8000	800

This is the Sales record report interface of the system. You can view all sales records in the system through this section.

6. Appendix

6.1. Login Interface coding

```
18 }
19 int loginAtmp = 0;
20 1reference
21 private void btnLogin_Click(object sender, EventArgs e)
22 {
23     while (loginAtmp <= 5)
24     {
25         if (txtUsername.Text == "admin" && txtPassword.Text == "1234")
26         {
27             MessageBox.Show("Login Successful", "", MessageBoxButtons.OK, MessageBoxIcon.Information);
28             this.Hide();
29             Dashboard f1 = new Dashboard();
30             f1.Show();
31             break;
32         }
33         else
34         {
35             MessageBox.Show("Incorrect username or password! \n" + "You have " + (4 - loginAtmp) + " login attempts left", "", MessageBoxButtons.OK, MessageBoxIcon.Warning);
36             loginAtmp++;
37             txtUsername.Text = "";
38             txtPassword.Text = "";
39             if (loginAtmp == 5)
40             {
41                 MessageBox.Show("Maximum login attempts exceeded.\n" + "The application will close immediately!!!", "", MessageBoxButtons.OK, MessageBoxIcon.Error);
42                 this.Close();
43             }
44             break;
45         }
46     }
47 }
48 }
```

6.2. Dashboard coding

```
7 using System.Text;
8 using System.Threading.Tasks;
9 using System.Windows.Forms;
10
11 namespace Pizza_place_POS____E2046073_
12 {
13     public partial class Dashboard : Form
14     {
15         public Dashboard()
16         {
17             InitializeComponent();
18         }
19
20         private void btnCreg_Click(object sender, EventArgs e)
21         {
22             CustomerReg cregfrm = new CustomerReg();
23             cregfrm.Show();
24             this.Hide();
25         }
26
27         private void btnInventory_Click(object sender, EventArgs e)
28         {
29             Inventory inventfrm = new Inventory();
30             inventfrm.Show();
31             this.Hide();
32         }
33
34         private void btnPOS_Click(object sender, EventArgs e)
35         {
36             Cashier cshfrm = new Cashier();
37             cshfrm.Show();
38             this.Hide();
39         }
40
41         private void btnReports_Click(object sender, EventArgs e)
42         {
43             Reports rpts = new Reports();
44             rpts.Show();
45             this.Hide();
46         }
47
48         private void btnLogOut_Click(object sender, EventArgs e)
49         {
50             Application.Exit();
51         }
52     }
53 }
```

6.3. Customer Registration coding

```

18 public CustomerReg()
19 {
20     InitializeComponent();
21 }
22
23 1 reference
24 private void btnCreg_Click(object sender, EventArgs e)
25 {
26     try
27     {
28         con.Open();
29         string query = "Insert into LoyalCustomerTB(CustomerName,ContactNo,LoyaltyPoints) values('" + txtCName.Text + "', '" + txtCNO.Text + "', '" + 0 + "')";
30         cmd = new SqlCommand(query, con);
31         cmd.ExecuteNonQuery();
32     }
33     catch (Exception ex)
34     {
35         MessageBox.Show(ex.Message);
36     }
37     finally
38     {
39         con.Close();
40     }
41     txtCName.Clear();
42     txtCNO.Clear();
43     txtCID.Clear();
44     MessageBox.Show("Customer successfully registered ", "Pizza Place POS", MessageBoxButtons.OK, MessageBoxIcon.Information);
45 }
46
47 1 reference
48 private void btnDashboard_Click(object sender, EventArgs e)
49 {
50     Dashboard dshbdFrm = new Dashboard();
51     dshbdFrm.Show();
52     this.Hide();
53 }
54
55 }
56

```

6.4. Product registration (Inventory) coding

```

14 public partial class Inventory : Form
15 {
16     SqlConnection con = new SqlConnection("Data Source=LAPTOP-N69JLGVQ;Initial Catalog=PPPOSDB;Integrated Security=True");
17     SqlCommand cmd;
18     1 reference
19     public Inventory()
20     {
21         InitializeComponent();
22     }
23
24
25 1 reference
26 private void btnAdd_Click(object sender, EventArgs e)
27 {
28     try
29     {
30         con.Open();
31         string query = "Insert into ProductTB values('" + txtPName.Text + "', '" + int.Parse(txtPrice.Text) + "', '" + int.Parse(txtPQTY.Text) + "')";
32         cmd = new SqlCommand(query, con);
33         cmd.ExecuteNonQuery();
34     }
35     catch (Exception ex)
36     {
37         MessageBox.Show(ex.Message);
38     }
39     finally
40     {
41         con.Close();
42     }
43     txtPName.Clear();
44     txtPQTY.Clear();
45     txtPrice.Clear();
46     MessageBox.Show("Product successfully registered ", "Pizza Place POS", MessageBoxButtons.OK, MessageBoxIcon.Information);
47 }
48
49
50 1 reference
51 private void btnDashboard_Click(object sender, EventArgs e)
52 {
53     Dashboard dshbdFrm = new Dashboard();
54     dshbdFrm.Show();
55     this.Hide();
56 }
57
58 }
59

```

6.5. Cashier Form (Sales Management) coding

```

8      using System.Threading.Tasks;
9      using System.Windows.Forms;
10     using System.Data.SqlClient;
11
12     namespace Pizza_place_POS____E2046073_
13     {
14         public partial class Cashier : Form
15         {
16             public Cashier()
17             {
18                 InitializeComponent();
19             }
20             SqlConnection con = new SqlConnection("Data Source=LAPTOP-N69JLGVQ;Initial Catalog=PPPOSD;Integrated Security=True");
21             SqlCommand cmd;
22             SqlDataReader dr;
23             private void Cashier_Load(object sender, EventArgs e)
24             {
25                 timer1.Start();
26                 txtLCName.ReadOnly = true;
27                 txtLPoints.ReadOnly = true;
28                 con.Open();
29                 BindData();
30                 con.Close();
31             }
32             public void BindData()
33             {
34                 cmd = new SqlCommand("select name from ProductTB", con);
35                 dr = cmd.ExecuteReader();
36                 while (dr.Read())
37                 {
38                     comboBox1.Items.Add(dr[0].ToString());
39                 }
40                 dr.Close();
41             }

```

```

42
43
44
45     private void btnAddtoCart_Click(object sender, EventArgs e)
46     {
47
48         try
49         {
50             dgvCart.Rows.Add(comboBox1.Text, int.Parse(txtCPrice.Text), int.Parse(txtQty.Text), int.Parse(txtCPrice.Text) * int.Parse(txtQty.Text));
51             int sum = 0;
52             for (int i = 0; i < dgvCart.Rows.Count; ++i)
53             {
54                 sum += Convert.ToInt32(dgvCart.Rows[i].Cells[3].Value);
55             }
56             txtCBill.Text = sum.ToString();
57             int lp = int.Parse(txtLPoints.Text);
58             int sleProfit = (sum * 10 / 100) - lp;
59
60             if (lp > 100)
61             {
62                 txtLD.Text = txtLPoints.Text;
63                 int finalBill = (sum - lp);
64                 txtFBill.Text = finalBill.ToString();
65                 sleProfit = (sum * 10 / 100) - lp;
66                 txtSleProfit.Text = sleProfit.ToString();
67             }
68
69             }
70         else
71         {
72             txtLD.Text = "0";
73             txtFBill.Text = sum.ToString();
74             sleProfit = (sum * 10 / 100);
75             txtSleProfit.Text = sleProfit.ToString();
76         }
77
78         con.Open();
79         SqlCommand cmd = new SqlCommand("update ProductTB set ProductQuantity='"+ (int.Parse(txtAPQty.Text) - int.Parse(txtQty.Text)) + "' where Name='"+ comboBox1.Text + "'", con);
80         cmd.ExecuteNonQuery();
81         con.Close();
82     }
83     catch (Exception ex)
84     {
85         MessageBox.Show(ex.Message);
86     }
87
88
89
90

```

```

93     1 reference
94     private void txtLCID_TextChanged(object sender, EventArgs e)
95     {
96         con.Open();
97         SqlCommand cmd = new SqlCommand("select CustomerName,LoyaltyPoints from LoyalCustomerTB where CustomerID='" + txtLCID.Text + "'", con);
98         SqlDataReader srd = cmd.ExecuteReader();
99         while (srd.Read())
100         {
101             txtLCName.Text = srd.GetValue(0).ToString();
102             txtLPPoints.Text = srd.GetValue(1).ToString();
103         }
104         con.Close();
105     }
106
107     1 reference
108     private void comboBox1_SelectedValueChanged(object sender, EventArgs e)
109     {
110         con.Open();
111         SqlCommand cmd = new SqlCommand("select Price,ProductQuantity from ProductTB where Name='" + comboBox1.Text + "'", con);
112         SqlDataReader srd = cmd.ExecuteReader();
113         while (srd.Read())
114         {
115             txtCIPrice.Text = srd.GetValue(0).ToString();
116             txtAPQty.Text = srd.GetValue(1).ToString();
117         }
118         con.Close();
119     }
120
121     1 reference
122     private void comboBox2_SelectedValueChanged(object sender, EventArgs e)
123     {
124         if (comboBox2.Text=="Normal")
125         {
126             txtLCID.ReadOnly = true;
127         }
128         else
129         {
130             txtLCID.ReadOnly = false;
131         }
132     }

```

```

130     }
131
132     1 reference
133     private void btnPay_Click(object sender, EventArgs e)
134     {
135         DateTime datetime = DateTime.Now;
136         int change = int.Parse(txtCash.Text) - int.Parse(txtBill.Text);
137         try
138         {
139             con.Open();
140             string query = "Insert into SalesRecordsTB values('" + datetime.ToString("MM/dd/yyyy hh:mm tt") + "', '" + txtCBill.Text + "','" + txtCID.Text + "','" + txtfBill.Text + "','" + txtLPPoints.Text + "')";
141             cmd = new SqlCommand(query, con);
142             cmd.ExecuteNonQuery();
143         }
144         catch (Exception ex)
145         {
146             MessageBox.Show(ex.Message);
147         }
148         finally
149         {
150             con.Close();
151         }
152         int lp = int.Parse(txtLPPoints.Text);
153         if (lp > 100)
154         {
155             con.Open();
156             SqlCommand cmd2 = new SqlCommand("update LoyalCustomerTB set LoyaltyPoints='" + 0 + "' where CustomerID='" + txtLCID.Text + "'", con);
157             cmd2.ExecuteNonQuery();
158             con.Close();
159         }
160         else
161         {
162             con.Open();
163             SqlCommand cmd2 = new SqlCommand("update LoyalCustomerTB set LoyaltyPoints='" + int.Parse(txtLPPoints.Text) + int.Parse(txtfBill.Text) + "' where CustomerID='" + txtLCID.Text + "'", con);
164             cmd2.ExecuteNonQuery();
165         }
166         int chnge = int.Parse(txtCash.Text) - int.Parse(txtfBill.Text);
167
168         MessageBox.Show(string.Format("Transaction successfull, Change for Customer is Rs. {0}", chnge), "Pizza Place POS", MessageBoxButtons.OK, MessageBoxIcon.Information);
169
170         Invoice invce = new Invoice();
171         invce.CName = txtLCName.Text;
172         invce.Bill = txtCBill.Text;
173         invce.Ldis = txtLPPoints.Text;
174         invce.Pbill = txtfBill.Text;
175         invce.Show();
176     }

```

```

181
182
183
184 1 reference
185 private void timer1_Tick(object sender, EventArgs e)
186 {
187     DateTime datetime = DateTime.Now;
188     lblTime.Text = datetime.ToString("MM/dd/yyyy hh:mm tt");
189 }
190
191 1 reference
192 private void btnDashboard_Click(object sender, EventArgs e)
193 {
194     Dashboard dshbdFrm = new Dashboard();
195     dshbdFrm.Show();
196     this.Hide();
197 }
198
199 1 reference
200 private void btnNew_Click(object sender, EventArgs e)
201 {
202     txtLCName.Clear();
203     txtLpoints.Text = "0";
204     txtCprice.Text = "0";
205     txtAPQty.Clear();
206     txtCBill.Text = "0";
207     txtLD.Text = "0";
208     txtSleProfit.Text = "0";
209     txtFbill.Text = "0";
210     txtCash.Clear();
211 }

```

6.6. Bill Printing coding

```
11 namespace Pizza_place_POS____E2046073_
12 {
13     4 references
14     public partial class Invoice : Form
15     {
16         public string Cname, bill, Ldis, Fbill;
17         Bitmap memoryImage;
18         1 reference
19         private void Invoice_Load(object sender, EventArgs e)
20         {
21             lbl1.Text = Cname;
22             lbl2.Text = bill;
23             lbl3.Text = Ldis;
24             lbl4.Text = Fbill;
25         }
26
27         1 reference
28         public Invoice()
29         {
30             InitializeComponent();
31         }
32
33         1 reference
34         private void btnPrint_Click(object sender, EventArgs e)
35         {
36             Graphics myGraphics = this.CreateGraphics();
37             Size s = this.Size;
38             memoryImage = new Bitmap(s.Width, s.Height, myGraphics);
39             Graphics memoryGraphics = Graphics.FromImage(memoryImage);
40             memoryGraphics.CopyFromScreen(this.Location.X, this.Location.Y, 0, 0, s);
41
42             printDocument1.Print();
43         }
44
45         0 references
46         private void PrintDocument1_PrintPage(System.Object sender,
47             System.Drawing.Printing.PrintPageEventArgs e) =>
48             e.Graphics.DrawImage(memoryImage, 0, 0);
49     }
50 }
```


6.7. Customer Management coding

```

9  using System.Windows.Forms;
10 using System.Data.SqlClient;
11
12 namespace Pizza_place_POS__F2046073_
13 {
14     4 references
15     public partial class Reports : Form
16     {
17         1 reference
18         public Reports()
19         {
20             InitializeComponent();
21         }
22         SqlConnection con = new SqlConnection("Data Source=LAPTOP-N69JLGVQ;Initial Catalog=PPPOSDB;Integrated Security=True");
23         SqlCommand cmd;
24         SqlDataReader dr;
25         1 reference
26         private void Reports_Load(object sender, EventArgs e)
27         {
28             this.salesRecordsTBTableAdapter2.Fill(this.pPOSDBDataSet5.SalesRecordsTB);
29             this.productTBTableAdapter1.Fill(this.pPOSDBDataSet4.ProductTB);
30             this.loyalCustomerTBTableAdapter.Fill(this.pPOSDBDataSet.LoyalCustomerTB);
31             con.Open();
32             BindData();
33             con.Close();
34         }
35         1 reference
36         public void BindData()
37         {
38             cmd = new SqlCommand("select name from ProductTB", con);
39             dr = cmd.ExecuteReader();
40             while (dr.Read())
41             {
42                 comboBox1.Items.Add(dr[0].ToString());
43             }
44             dr.Close();
45         }
46         1 reference
47         private void txtCID_TextChanged(object sender, EventArgs e)
48         {
49             con.Open();
50             SqlCommand cmd = new SqlCommand("select CustomerName,ContactNo,LoyaltyPoints from LoyalCustomerTB where CustomerID='" + txtCID.Text + "'", con);
51             SqlDataReader srd = cmd.ExecuteReader();
52             while (srd.Read())
53             {
54                 txtCName.Text = srd.GetValue(0).ToString();
55                 txtCNO.Text = srd.GetValue(1).ToString();
56                 txtLP.Text = srd.GetValue(2).ToString();
57             }
58             con.Close();
59         }
60
61         1 reference
62         private void btnUpdate_Click(object sender, EventArgs e)
63         {
64             con.Open();
65             SqlCommand cmd = new SqlCommand("update LoyalCustomerTB set CustomerName='" + txtCName.Text + "',ContactNo='" + txtCNO.Text + "',LoyaltyPoints='" + txtLP.Text + "' where CustomerID='" + txtCID.Text + "'", con);
66             cmd.ExecuteNonQuery();
67             con.Close();
68             this.loyalCustomerTBTableAdapter.Fill(this.pPOSDBDataSet.LoyalCustomerTB);
69             txtCName.Clear();
70             txtCNO.Clear();
71             txtCID.Clear();
72             MessageBox.Show("Customer Details successfully Updated! ", "Pizza Place POS", MessageBoxButtons.OK, MessageBoxIcon.Information);
73         }
74
75         1 reference
76         private void btnDel_Click(object sender, EventArgs e)
77         {
78             con.Open();
79             SqlCommand cmd = new SqlCommand("delete from LoyalCustomerTB where CustomerID='" + txtCID.Text + "'", con);
80             cmd.ExecuteNonQuery();
81             con.Close();
82             this.loyalCustomerTBTableAdapter.Fill(this.pPOSDBDataSet.LoyalCustomerTB);
83             txtCName.Clear();
84             txtCNO.Clear();
85             txtCID.Clear();
86             MessageBox.Show("Customer successfully deleted! ", "Pizza Place POS", MessageBoxButtons.OK, MessageBoxIcon.Information);
87         }
88     }
89 }


```

6.8. Products Inventory Management coding

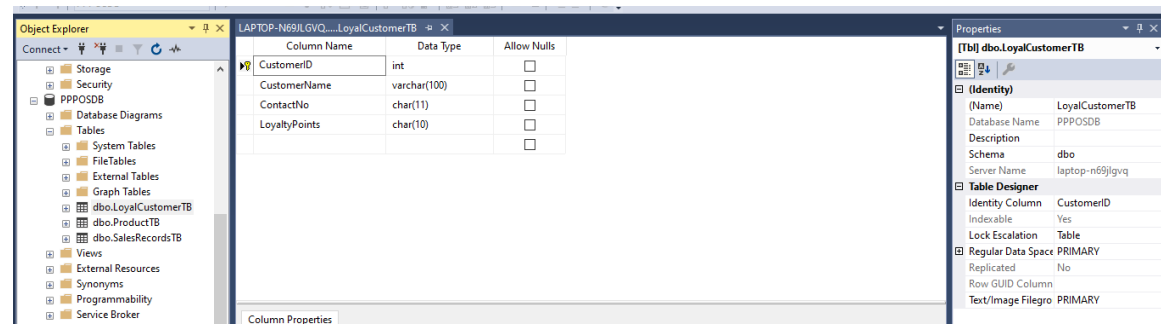
```
83 private void comboBox1_SelectedIndexChanged(object sender, EventArgs e)
84 {
85     con.Open();
86     SqlCommand cmd = new SqlCommand("select Price,Profit,ProductQuantity from ProductTB where Name='" + comboBox1.Text + "'", con);
87     SqlDataReader srd = cmd.ExecuteReader();
88     while (srd.Read())
89     {
90         txtPrice.Text = srd.GetValue(0).ToString();
91
92         txtPQTY.Text = srd.GetValue(2).ToString();
93     }
94     con.Close();
95     this.productTBTableAdapter.Fill(this.pPPOSDBDataSet1.ProductTB);
96 }
97
98 private void btnUpdate2_Click(object sender, EventArgs e)
99 {
100     con.Open();
101     SqlCommand cmd = new SqlCommand("update ProductTB set Price='" + txtPrice.Text + "',ProductQuantity='" + txtPQTY.Text + "' where Name='" + comboBox1.Text + "'", con);
102     cmd.ExecuteNonQuery();
103     con.Close();
104     this.productTBTableAdapter1.Fill(this.pPPOSDBDataSet4.ProductTB);
105     ;
106     txtPQTY.Clear();
107     txtPrice.Clear();
108     MessageBox.Show("Product successfully Updated ", "Pizza Place POS", MessageBoxButtons.OK, MessageBoxIcon.Information);
109 }
110
111 private void btnDelete2_Click(object sender, EventArgs e)
112 {
113     con.Open();
114     SqlCommand cmd = new SqlCommand("delete from ProductTB where Name='" + comboBox1.Text + "'", con);
115     cmd.ExecuteNonQuery();
116     con.Close();
117     this.productTBTableAdapter1.Fill(this.pPPOSDBDataSet4.ProductTB);
118     txtPQTY.Clear();
119     txtPrice.Clear();
120     MessageBox.Show("Product successfully Deleted ", "Pizza Place POS", MessageBoxButtons.OK, MessageBoxIcon.Information);
121 }
122
123 private void btnLowSTK_Click(object sender, EventArgs e)
124 {
125
126     dataGridView2.Sort(dataGridView2.Columns[3], ListSortDirection.Ascending);
127
128 }
```

6.9. Sales Records coding

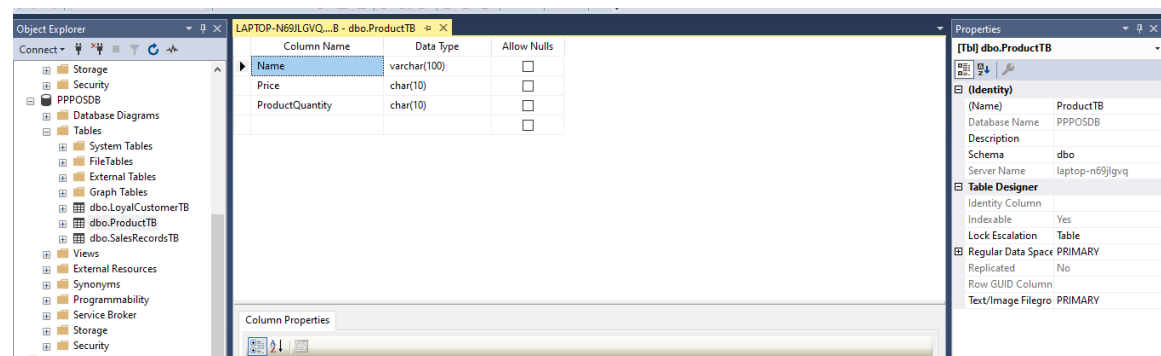
```
14 public partial class Reports : Form
15 {
16     1 reference
17     public Reports()
18     {
19         InitializeComponent();
20
21         SqlConnection con = new SqlConnection("Data Source=LAPTOP-N69JLGWQ;Initial Catalog=PPPOSDB;Integrated Security=True");
22         SqlCommand cmd;
23         SqlDataReader dr;
24     private void Reports_Load(object sender, EventArgs e)
25     {
26         this.salesRecordsTBTableAdapter2.Fill(this.pPPOSDBDataSet5.SalesRecordsTB);
27         this.productTBTableAdapter1.Fill(this.pPPOSDBDataSet4.ProductTB);
28         this.loyalCustomerTBTableAdapter.Fill(this.pPPOSDBDataSet.LoyalCustomerTB);
29         con.Open();
30         BindData();
31         con.Close();
32     }
}
```



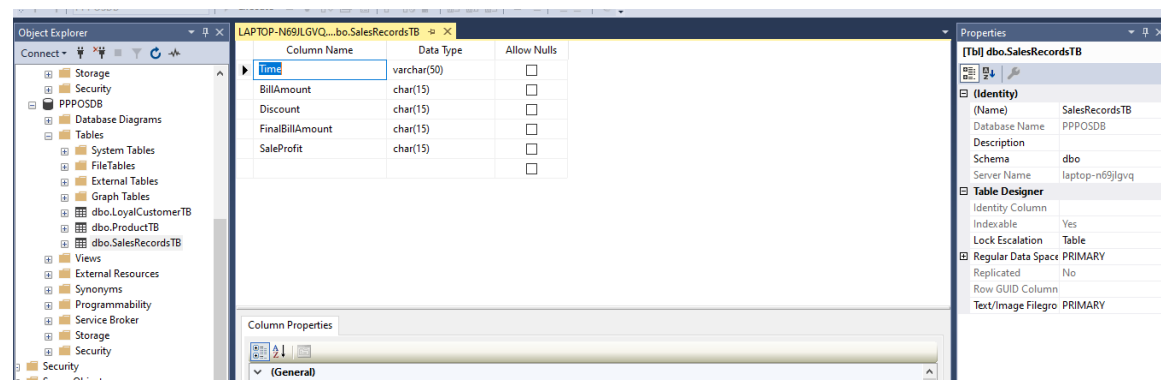
6.10. Customer table of database (PPPOSDB)



6.11. Product table of database (PPPOSDB)



6.12. Sales records table of database (PPPOSDB)



7. References

- [1] <https://i.pinimg.com/564x/b4/fd/4c/b4fd4c202f78eb52211809d6047eecb4.jpg>
- [2] <https://www.pos-hub.com/pos-and-inventory-managament-software-for-restaurant>
- [3] https://images.g2crowd.com/uploads/attachment/file/25359/uploads_2F8502277a-be03-4c5c-b483-6ed80bbe3434_2F2.png
- [4] <https://reviews.financesonline.com/p/gofrugal-pos/#user-review>
- [5] <https://www.gofrugal.com/images/vertical-images/Restaurant-bar/table-select.jpg>