POS system

Daniel Lozovskiy, Qing Liang, Zack Lewis

Outline

- ❑ **Abstract**
- ❑ **Objectives**
- ❑ **Process and Methodology**
- ❑ **Specifications / Design**
- ❑ **Demo!**
- ❑ **Timeline**
- ❑ **Future improvements**
- ❑ **Conclusion**

Abstract

A Point of Sale system (POS) is a device used to handle the transaction between a business and the customer. Our project is to create a POS system intended to be used by a restaurants. Our POS system will be able to handle orders, checkout, inventory/menu management, and some analysis.

Objectives

- General objective
  - Increase speed of services, sales volumes, and customer satisfaction
- Ordering system specifications
  - Add/view/delete menu item
  - Place orders/add specification
  - Display orders (open, paid, served, complete)
  - Calculate total amount due/ Receive payment
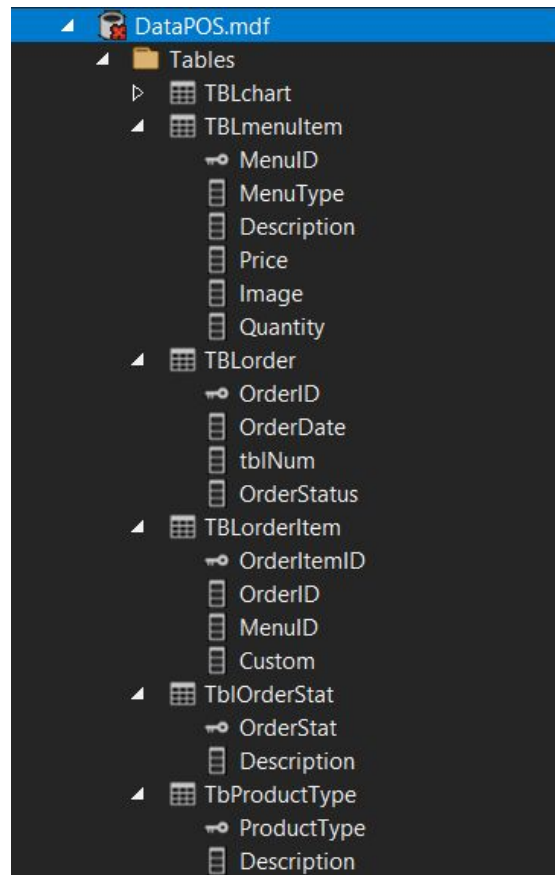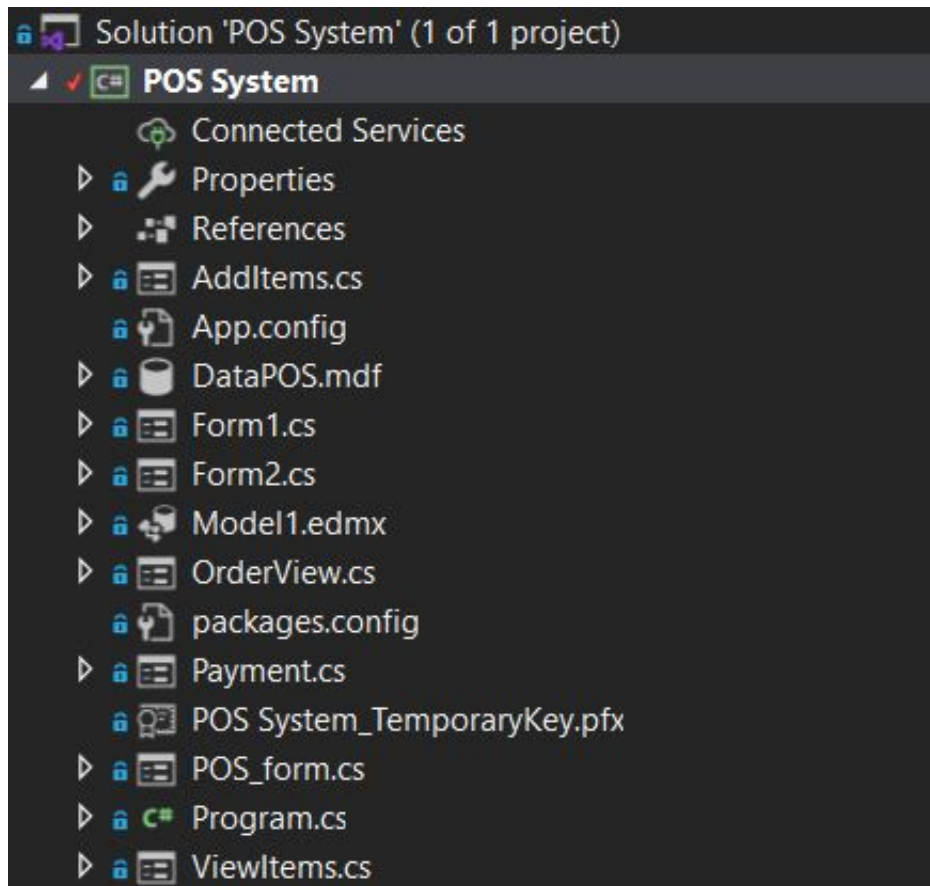 - Print recept
  - Sales Graph

Process and Methodology
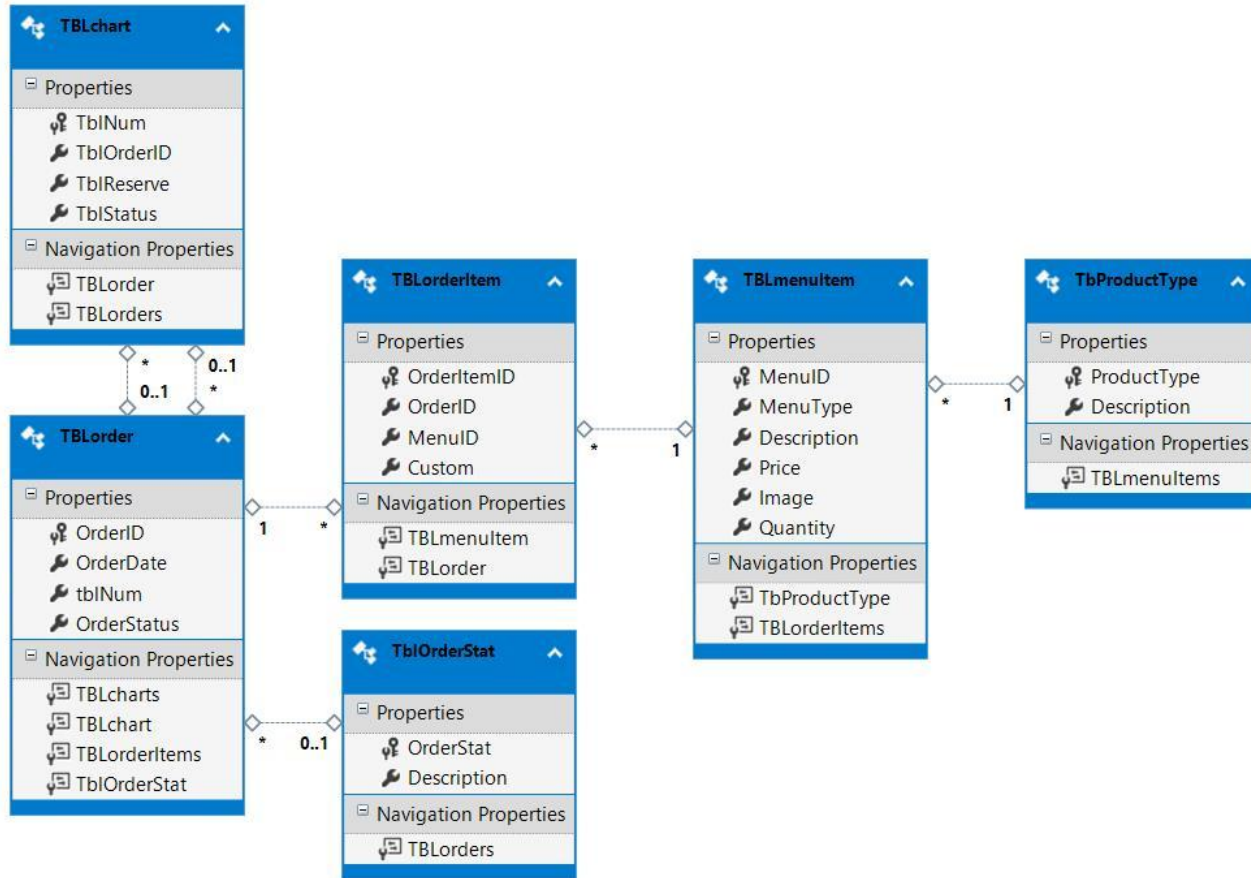
Programming Language: C#

.NET Framework Data Provider for SQL Server

Visual Studio 2019

Solution 'POS System' (1 of 1 project)
- ✓ **POS System**
  - Connected Services
  - Properties
  - References
  - AddItems.cs
  - App.config
  - DataPOS.mdf
  - Form1.cs
  - Form2.cs
  - Model1.edmx
  - OrderView.cs
  - packages.config
  - Payment.cs
  - POS System_TemporaryKey.pfx
  - POS_form.cs
  - Program.cs
  - ViewItems.cs

DataPOS.mdf
- Tables
  - TBLchart
  - TBLmenuItem
    - MenuID
    - MenuType
    - Description
    - Price
    - Image
    - Quantity
  - TBLorder
    - OrderID
    - OrderDate
    - tblNum
    - OrderStatus
  - TBLorderItem
    - OrderItemID
    - OrderID
    - MenuID
    - Custom
  - TblOrderStat
    - OrderStat
    - Description
  - TbProductType
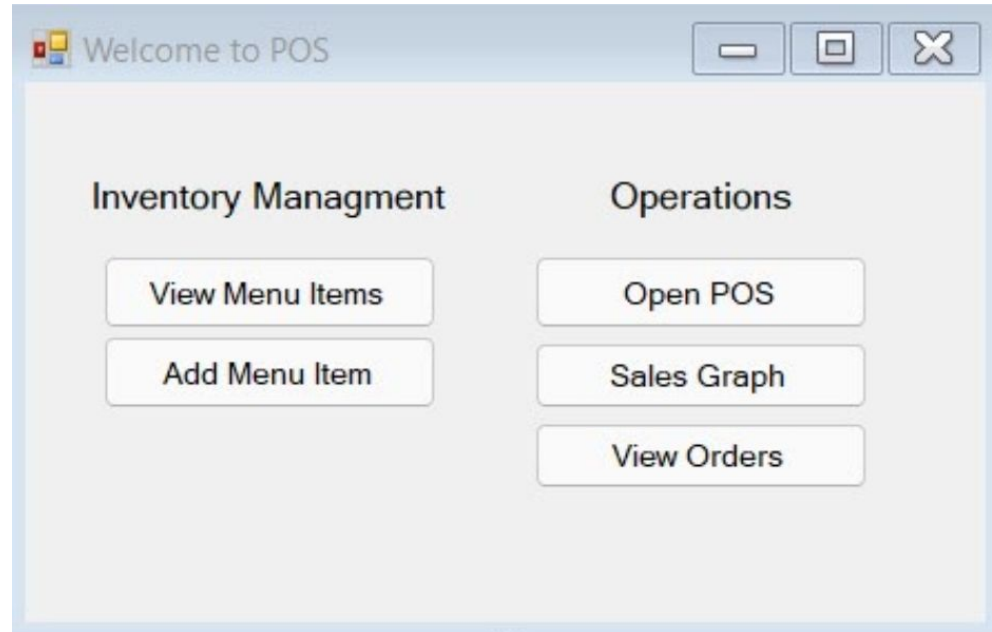   - ProductType
    - Description

Database

Ordering System Specifications

- Inventory Management
  - Add Menu Items
  - View Menu Item
- Operation section
  - Open POS
  - Sales Graph
  - View orders page

```csharp
public Form1()
{
    InitializeComponent();
}

1 reference | dlozovskiy, 16 days ago | 1 author, 1 change
private void Button1_Click(object sender, EventArgs e)
{
    AddItems addItems = new AddItems();
    addItems.Show();
}

1 reference | jane liang, 16 days ago | 1 author, 1 change
private void button2_Click(object sender, EventArgs e)
{
    ViewItems viewItems = new ViewItems();
    viewItems.Show();
}

1 reference | z-lew, 9 days ago | 1 author, 1 change
private void btnOpenPOS_Click(object sender, EventArgs e)
{
    POS_form posForm = new POS_form();

    posForm.Show();
}

1 reference | dlozovskiy, 3 days ago | 1 author, 1 change
private void btnSales_Click(object sender, EventArgs e)
{
    DisplaySales displaySales = new DisplaySales();
    displaySales.Show();
}
```
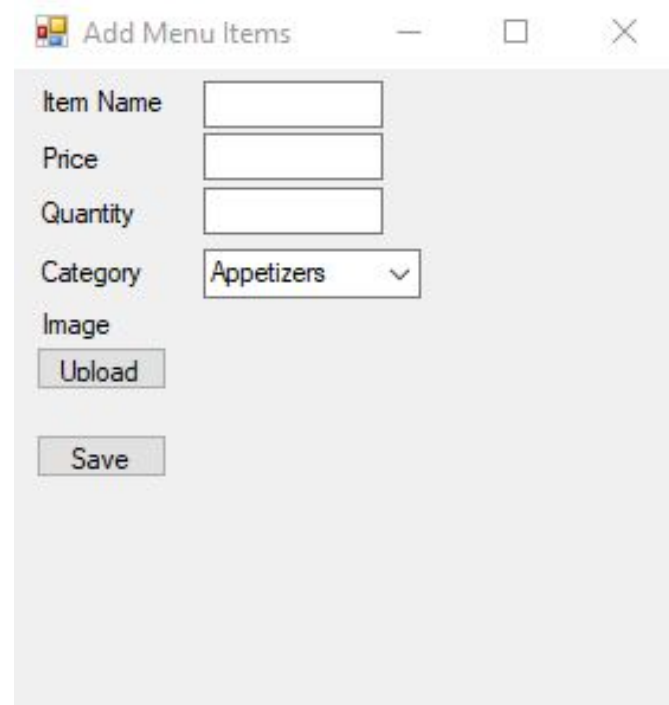
Code Snippet for Main Menu

Inventory Management

- Add Menu Items
  - Item name
  - Price
  - Quantity
  - Category
  - Image

```csharp
private void btUpload_Click(object sender, EventArgs e)
{
    DialogResult result = openFileDialog1.ShowDialog();

    if (result == DialogResult.OK)
    {
        FileStream fsFile = new FileStream(openFileDialog1.FileName, FileMode.Open, FileAccess.Read);

        byteData = new byte[fsFile.Length];

        fsFile.Read(byteData, 0, byteData.Length);

        fsFile.Close();

        MemoryStream stmData = new MemoryStream(byteData);

        pbImage.Image = Image.FromStream(stmData);
    }
}

1 reference | dlozovskiy, 2 days ago | 1 author, 3 changes
private void btSave_Click(object sender, EventArgs e)
{
    TBLmenuItem item = new TBLmenuItem();
    try
    {
        item.Description = txtName.Text;
        item.Price = decimal.Parse(txtPrice.Text);
        item.Quantity = int.Parse(txtQuantity.Text);
        item.Image = byteData;
        item.MenuType = (int)cbCategory.SelectedValue;

        pos.TBLmenuItems.Add(item);
        pos.SaveChanges();
        MessageBox.Show("New Item Saved");
        txtName.Clear();
        txtPrice.Clear();
        txtQuantity.Clear();
    }
    catch
    {
        MessageBox.Show("Please Check Fields");
    }
}
```

Code for the addition of new menu items

Inventory Management

- View Menu Items
  - MenuID
  - Description
  - Price
  - Image
  - Quantity
  - Delete

```csharp
private DataPOSEntities pos = new DataPOSEntities();
1 reference | dlozovskiy, 6 hours ago | 2 authors, 3 changes
public ViewItems()
{
    InitializeComponent();

    Filter.DisplayMember = "Description";
    Filter.ValueMember = "ProductType";
    Filter.DataSource = pos.TbProductTypes.ToList();

    dataGridView1.DataSource = returnFilter().ToList();


    dataGridView1.Columns["MenuType"].Visible = false;
    dataGridView1.Columns["TbProductType"].Visible = false;
    dataGridView1.Columns["TBLorderItems"].Visible = false;


    //Delete Button

    var deleteButton = new DataGridViewButtonColumn();
    deleteButton.Name = "dataGridViewDeleteButton";
    deleteButton.HeaderText = "Delete";
    deleteButton.Text = "Delete";
    deleteButton.UseColumnTextForButtonValue = true;
    this.dataGridView1.Columns.Add(deleteButton);



}
```

```csharp
private void dataGridView1_CellContentClick(object sender, DataGridViewCellEventArgs e)
{
    //if click is on new row or header row
    if (e.RowIndex == dataGridView1.NewRowIndex || e.RowIndex < 0)
        return;

    //Check if click is on specific column
    if (e.ColumnIndex == dataGridView1.Columns["dataGridViewDeleteButton"].Index)
    {
        //Put some logic here, for example to remove row from your binding list.

        var toDelete = (TBLmenuItem)dataGridView1.Rows[e.RowIndex].DataBoundItem;


        pos.TBLmenuItems.Remove(toDelete);
        pos.SaveChanges();
        dataGridView1.DataSource = returnFilter().ToList();
        // Or
        // var data = (Product)dataGridView1.Rows[e.RowIndex].DataBoundItem;
        // do something
    }

}
```

Code Snippet for the listing of menu items

Operation Section

- Open POS
  - Take out/dine in
  - Pay now/Place order
  - Customization
  - Delete order

```
44    private void AddProductToPanel()
45    {
46        int i = 1;
47
48        foreach(TabPage tp in MenuTabs.TabPages)
49        {
50            //ObjectQuery<TBLmenuItem> filteredItem = new ObjectQuery<TBLmenuItem>("SELECT VALUE P FOM TBLmenuItems AS P", pos);
51            var ToShow = from b in pos.TBLmenuItems
52                         where b.MenuType == i
53                         select b;
54
55            FlowLayoutPanel flp = new FlowLayoutPanel();
56            flp.Dock = DockStyle.Fill;
57
58            foreach(TBLmenuItem tb in ToShow)
59            {
60                Button b = new Button();
61                b.Size = new Size(100, 100);
62                b.Text = tb.Description;
63
64                b.Tag = tb;
65                b.Click += UpdateItemList;
66
67                flp.Controls.Add(b);
68            }
69
70            tp.Controls.Add(flp);
71            i++;
72        }
73    }
74
```

```
private void UpdateCustomerInformationPanel(TBLmenuItem item)
{
    string currentDescription = item.Description;
    string currentPrice = String.Format("{0:c}", item.Price);
    string currentDescriptionPadded = currentDescription.PadRight(30);

    txtInfoPanel.Text = currentDescriptionPadded + currentPrice;
}
```

```
private void CreateTabbedPanel()
{
    foreach(TbProductType pt in pos.TbProductTypes)
    {
        MenuTabs.TabPages.Add(pt.ProductType.ToString(), pt.Description);
    }
}
```

```
private void FormatListItem(object sender, ListControlConvertEventArgs e)
{
    string currentDescription = ((TBLmenuItem)e.ListItem).Description;
    string currentPrice = String.Format("{0:c}",((TBLmenuItem)e.ListItem).Price);
    string currentDescriptionPadded = currentDescription.PadRight(30);

    e.Value = currentDescriptionPadded + currentPrice;
}
```
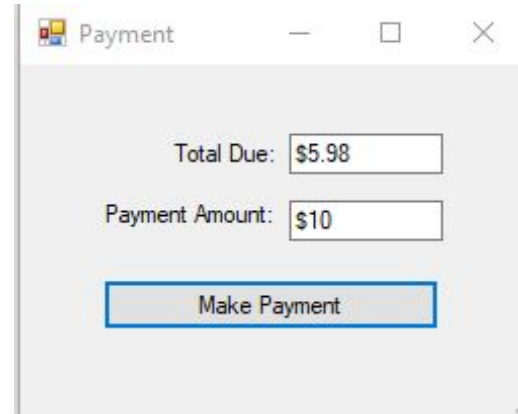
```
private void DeleteItem(object sender, EventArgs e)
{
    TBLmenuItem selectedItem = (TBLmenuItem)lstItemsChosen.SelectedItem;
    menuItems.Remove(selectedItem);
    OrderTotal = OrderTotal - (decimal)selectedItem.Price;
}
```
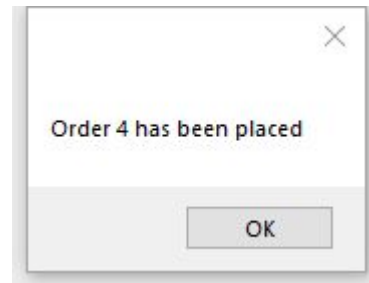
Code snippet for functionality of POS view

Operation Section

- Pay now
  - Payment page
- Place order
  - Order success page

```csharp
    private void PaymentMade(object sender, EventArgs e)
    {
        decimal Total = 0;

        try
        {
            Total = decimal.Parse(txtTotal.Text.TrimStart('$')) - decimal.Parse(txtPay.Text.TrimStart('$'));
        }
        catch
        {
            MessageBox.Show("Please enter a valid amount");
            return;
        }
        if (Total > 0)
        {
            txtTotal.Text = Total.ToString();
        }
        else
        {
            MessageBox.Show("Change: " + String.Format("{0:c}", -Total));
            GivenPayment(this, new PaymentEventMadeArg() { PaySuccess1 = true });
        }

    }

    1 reference | jane liang, 8 days ago | 1 author, 1 change
    private void Payment_Load(object sender, EventArgs e)
    {

    }
}
4 references | dlozovskiy, 8 days ago | 1 author, 1 change
public class PaymentEventMadeArg: EventArgs{
    private bool PaySuccess;

    3 references | dlozovskiy, 8 days ago | 1 author, 1 change
    public bool PaySuccess1
    {
        get { return PaySuccess; }
        set { PaySuccess = value; }
    }
}
```
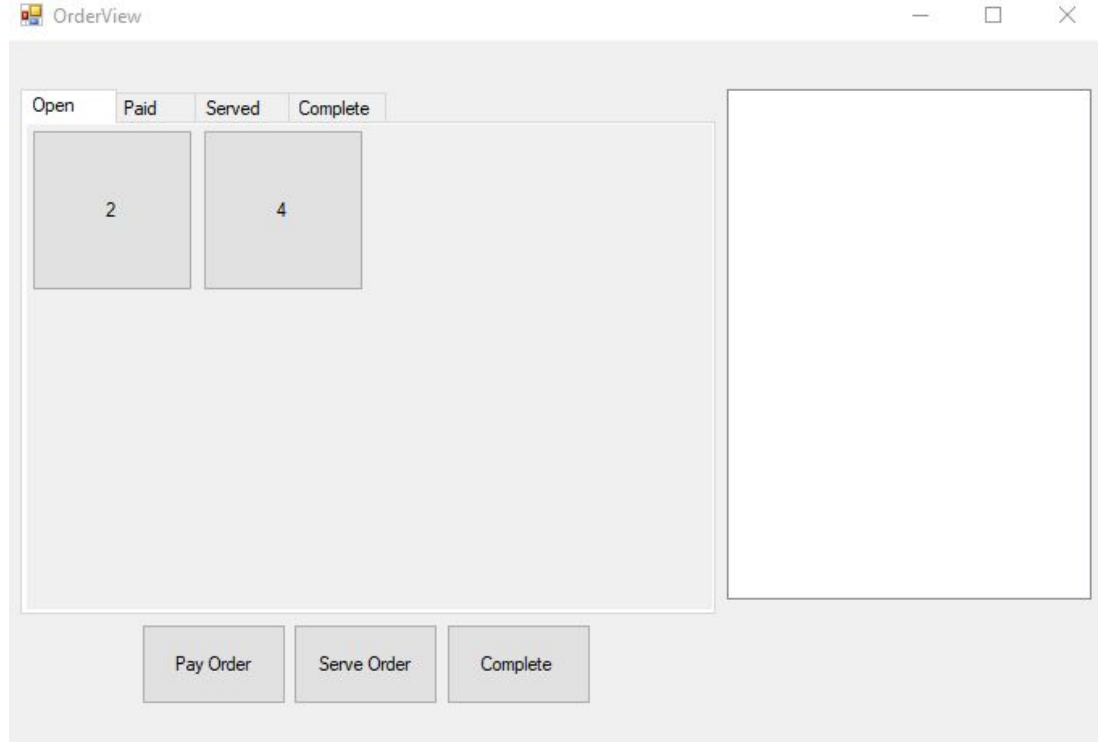
Code snippet for the payment functionality

Operation Section

- View orders
  - Open
  - Paid
  - Served
  - Complete

```csharp
public OrderView()
{
    InitializeComponent();
    CreatePanel();
    selectOrder();
    orderDetail.DataSource = orderMenu;
    orderDetail.DisplayMember = "Description";
}

1 reference | dlozovskiy, 7 hours ago | 1 author, 1 change
private void CreatePanel()
{
    foreach(TblOrderStat ot in pos.TblOrderStats)
    {
        orderTabs.TabPages.Add(ot.OrderStat.ToString(), ot.Description);
    }
}

4 references | dlozovskiy, 6 hours ago | 1 author, 2 changes
private void selectOrder()
{

    int orderStatIndex = 0;
    foreach (TabPage ot in orderTabs.TabPages)
    {
        ot.Controls.Clear();
        var ToShow = from b in pos.TBLorders
                     where b.OrderStatus == orderStatIndex
                     select b;

        FlowLayoutPanel flp = new FlowLayoutPanel();
        flp.Dock = DockStyle.Fill;

        foreach (TBLorder tb in ToShow)
        {
            Button b = new Button();
            b.Size = new Size(100, 100);
            b.Text = tb.OrderID.ToString();
            b.Tag = tb;
            b.Click += new EventHandler(OrderDetailList);
            flp.Controls.Add(b);
        }
        orderStatIndex += 1;
        ot.Controls.Add(flp);
    }
```

Code Snippet for managing orders

```csharp
private void FormatOrderList(object sender, ListControlConvertEventArgs e)
{
    string currentDescription = ((TBLmenuItem)e.ListItem).Description;
    //string currentPrice = String.Format("{0,c}",((TBLmenuItem)e.ListItem).Price);
    string currentDescriptionPadding = currentDescription.PadRight(30);

    e.Value = currentDescriptionPadding;
}

1 reference | dlozovskiy, 6 hours ago | 1 author, 1 change
private void btnServed_Click(object sender, EventArgs e)
{
    if(lastSelected != null)
    {

        lastSelected.OrderStatus = 2;
        pos.SaveChanges();
        selectOrder();
    }

}

1 reference | dlozovskiy, 6 hours ago | 1 author, 1 change
private void btnComplete_Click(object sender, EventArgs e)
{
    if(lastSelected != null)
    {
        lastSelected.OrderStatus = 3;
        pos.SaveChanges();
        selectOrder();
    }

}

1 reference | dlozovskiy, 6 hours ago | 1 author, 1 change
private void btnPaid_Click(object sender, EventArgs e)
{
    if (lastSelected != null)
    {
        lastSelected.OrderStatus = 1;
        pos.SaveChanges();
        selectOrder();
    }
}
```
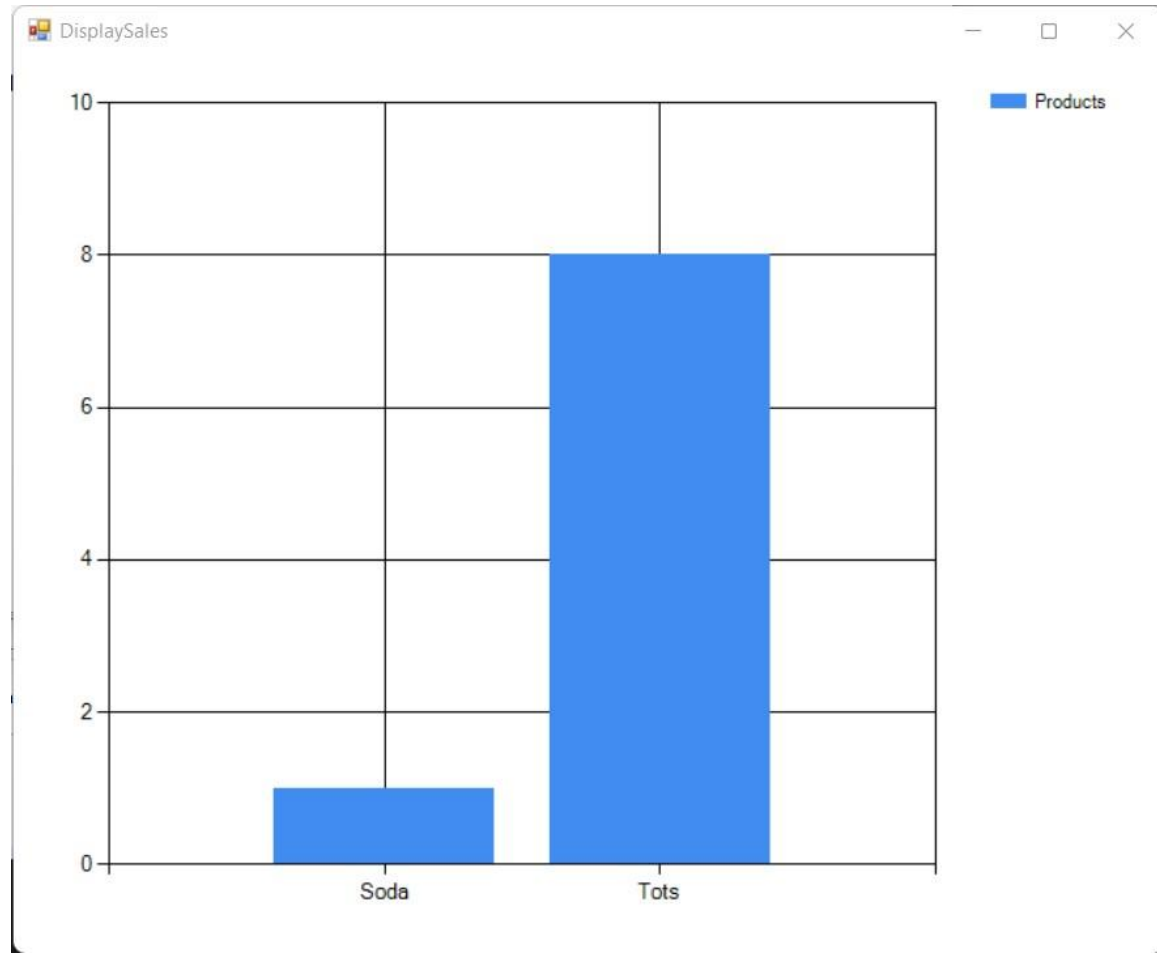
Operation Section

- Sales Graph

```csharp
private void makeGraph()
{
    using(DataPOSEntities pos = new DataPOSEntities())
    {
        var query = from item in pos.TBLorderItems
                    group item by item.TBLmenuItem.Description into g
                    select new { ItemID = g.Key, TotalUnitsSold = g.Count() };

        chart1.DataSource = query.ToList();

        chart1.Series[0].XValueMember = "ItemID";
        chart1.Series[0].YValueMembers = "TotalUnitsSold";
        chart1.Series[0].Name = "Products";

        chart1.DataBind();
        chart1.Show();
    }
}
```
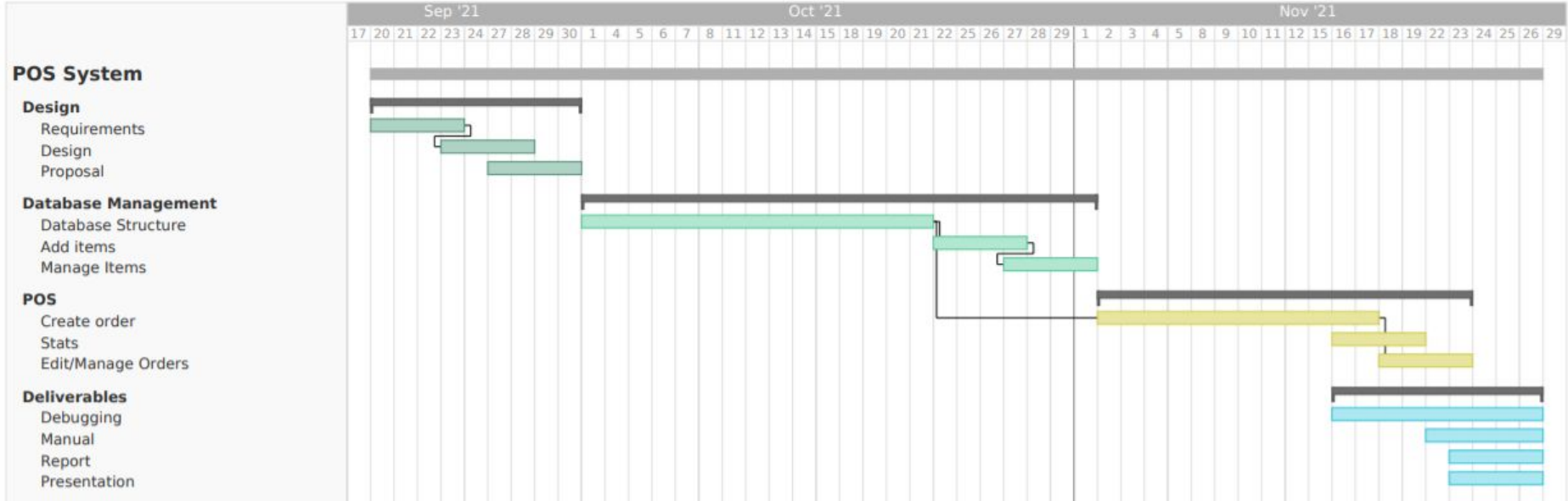
Code Snippet from Sales Bar Graph

Demo!

Timeline of Project

Future Improvements

- Add functionality for placed online
- Add a reservation feature
- Add more charts to show different statistics
- Ability to use coupon codes or receive loyalty reward points

Conclusion

Handwriting customer's orders is outdated and has many limitations; It takes a longer amount of time to place an order, increases chances of human error, and can be a huge burden on restaurants. POS systems in restaurants are convenient, effective, and easy to use, reducing the amount of time it takes to place and order and improving the overall performance of the restaurant.

This project help to improve:

- Group work skills
- Time management
- Accountability
- Coding skill set

Q&A!

References

[1]D. J. Oamen, "How to create point of sale management system in visual basic.net - full tutorial," YouTube, 17-Feb-2017. [Online]. Available: https://www.youtube.com/watch?v=DCCMLBNEds8. [Accessed: 26-Nov-2021].

[2]D. Liew, "Visual Basic 2019 Handbook," Visual Basic 2019 Handbook A Concise Guide to VB2019 Programming, 2019. [Online]. Available: https://www.vbtutor.net/vb2019/vb2019hndbk_preview.pdf. [Accessed: 2021].

[3] "IEEE code of Ethics," IEEE. [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html. [Accessed: 01-Oct-2021].

[4] "Restaurant point-of-sale terminal market worth $25.95 billion by 2028," Market Research Reports & Consulting. [Online]. Available: https://www.grandviewresearch.com/press-release/global-restaurant-point-of-sale-pos-terminal-market.

 [5]"Survey examines shift towards cashless society – Americans use less cash today than 10 years ago," MasterCard Social Newsroom. [Online]. Available: https://newsroom.mastercard.com/press-releases/survey-examines-shift-towards-cashless-society-americans-use-less-cash-today-than-10-years-ago/. [Accessed: 26-Nov-2021].