

CS 107.3 – Object Oriented Programming with C#

Group Assignment

Contents

.....	1
About the project	3
About the application.....	3
Function List of the POS System	3
Main Features.....	3
Admin Dashboard Page includes	3
Gantt Chart	4
ER Diagram	5
Class Diagram.....	5
.....	5
Wireframe of POS System	6
System Design	11
.....	11
Bibliography	20

Reg. No	Name
19582	JAMAB Jayakody
19987	EGPK Madushanka
19287	GMKA Thilakarathna
19182	NM Jayaweera
19204	K.G.S.A De silva
19131	SRCN Sabaraganuwa

Workload

EGPK Madushanka: -creating Dashboard/Database and design Wireframe

GMKA Thilakarathna: -Create the Employee Form and User Form

JAMAB Jayakody: - Implement Database connection and Design Login Form

Sabaraganuwa:-Creating Categories/Dealers Forms

NM Jayaweera: -Creating Customers/ Inventory Forms

K.G.S.A De silva: -Creating Transactions/Sales Forms

GitHub : - https://github.com/Malaka98/POS_System

About the project

POS-Sales and Inventory Management (POS System)

This is a new system. It is Modern, simple and easy-to-use management system which is developed by C# using WPF and SQL Server. This system allows you to manage Sales and Inventory of your Company. POS-Sales and Inventory Management (POS System)–has an easy Admin and User(Cashier) Dashboard pages that gives you Full Control to Manage Employees, Users, Products, Categories, Dealers(Suppliers), Customers, Inventory, Transactions, Purchases and Sales. So it helps to develop your business and it makes easy to get decision about the company.

About the application

When design this application mainly use the Windows Presentation Foundation (WPF). WPF is a UI framework. When design interface WPF better than Windows Form Application. WPF can make most pretty and attractive interface. When use WPF to design this application, [Material Design In XAML Toolkit](#) library use to develop UI. This library help to create most beautiful, attractive UI and It makes easy to develop very fast UI. In addition, [LiveCharts](#) use to display the charts about summarizing, sales and purchases etc. this library helps to create any type of chart (pie chart, bar chart, line chart etc.) with colorful and attractively.

Function List of the POS System

Main Features

- Manage Employees, Users, Dealers and Customers
- Manage Products, Categories, Inventory, Transactions and Records
- Reports -Employees, Users, Products, Dealers & Customers, Inventor & Transaction
- User Access Role

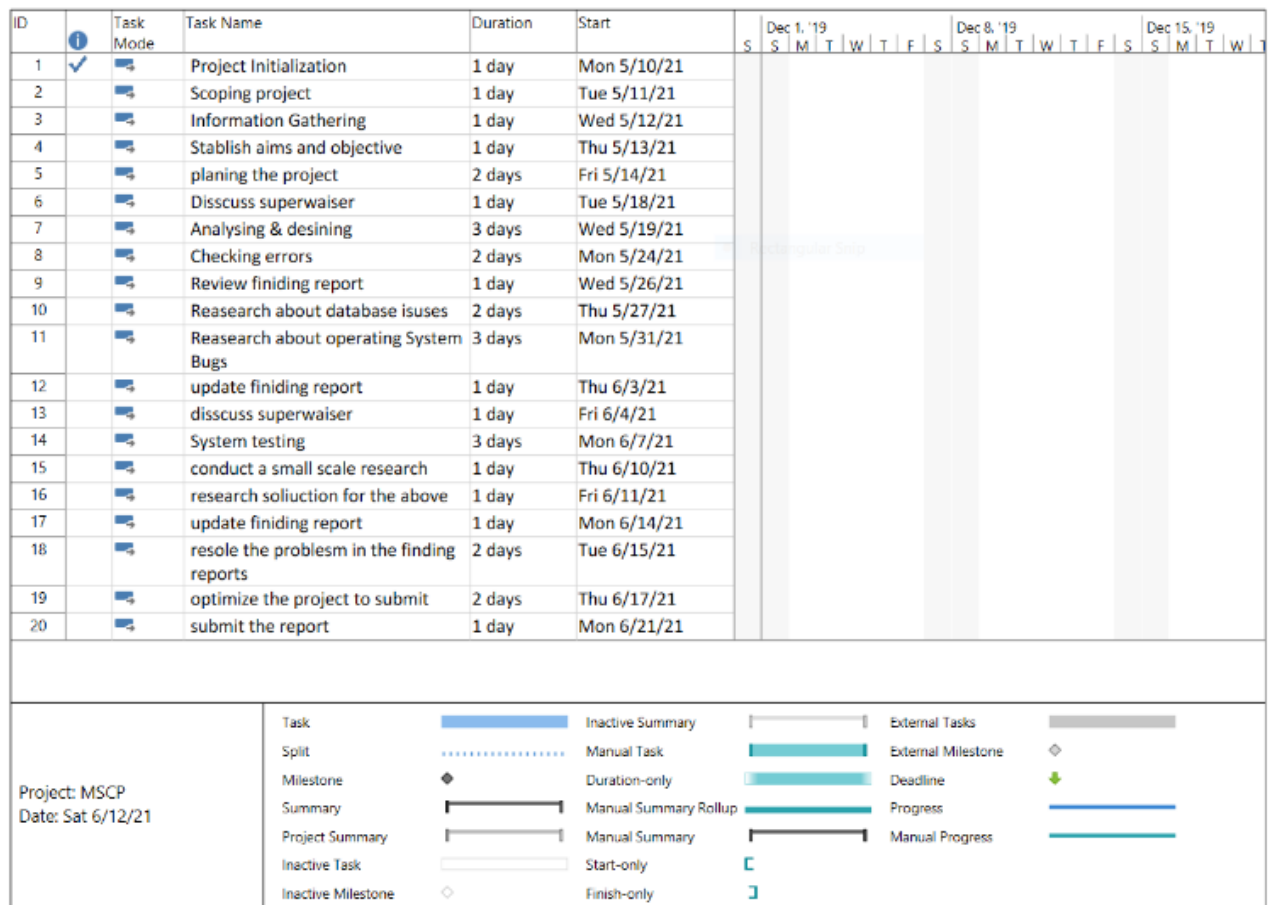
Admin Dashboard Page includes

POS System(Make a Purchase or Make a Sale)

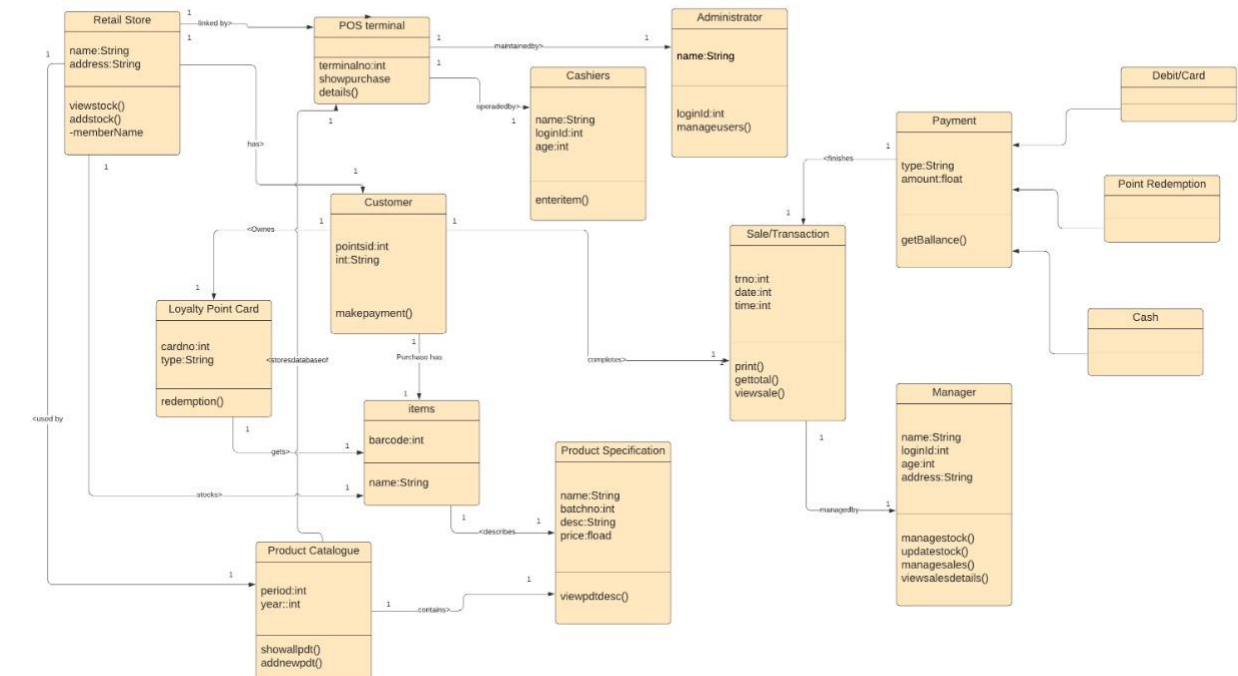
- Manage Employees
- Manage Users
- Manage Products

- Manage Categories
- Manage Dealers
- Manage Customers
- Manage Inventory
- Manage Transactions
- Manage Records

Gantt Chart



Class Diagram



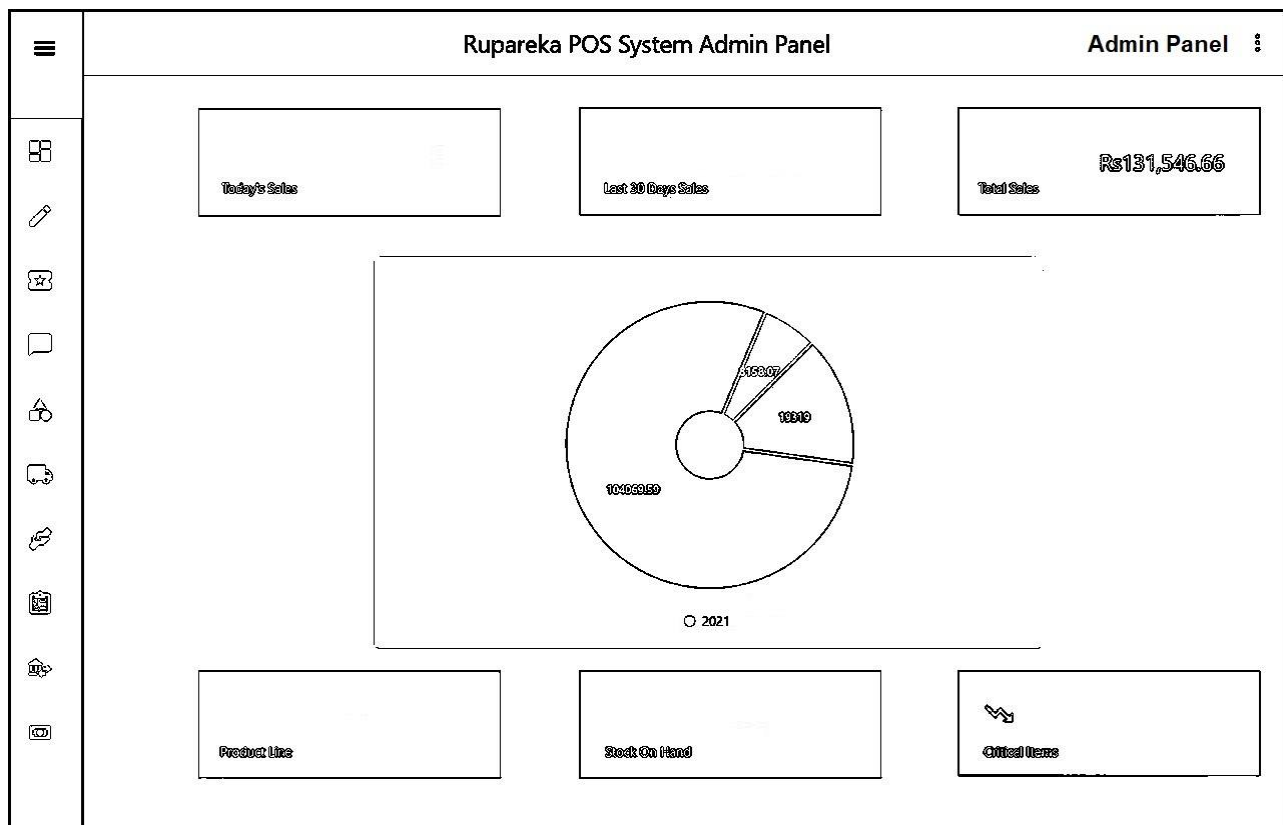
Wireframe of POS System

Login

User Name

Password

Login



Rupareka POS System Admin Panel

Admin Panel

User

Print

Name :

Surname :

User Name :

Password :

User Type :

Sex :

Date of birth :

Picture :

Select Image

UserID

Name

Surname

Username

Password

User Type

Gender

Birth Date

Img

isActive

Added Time

Insert

Update

Delete

Rupareka POS System Admin Panel

Admin Panel

Employees

Print

Name :

Surname :

ID Number :

Date of Birth :

Pick Date

Address :

Date Recived :

Pick Date

Sex :

Mobile :

Sallary :

Age :

Search :

EmplID

Name

Surname

ID

Gender

BirthDate

Age

Address

Mobile

Reciving Date

Sallary

Adder

Insert

Update

Delete

Rupareka POS System Admin Panel

Admin Panel

Categories

Category ID :

Title :

Description :

Search :

CatID

Title

Description

Added Time

Added By

Insert

Update

Delete

Rupareka POS System Admin Panel

Admin Panel

Inventory

Filter By Category :

Show All

Print Report

PID

PCode

Brand

Model

Full Name

Price

Category

D

Rupareka POS System Admin Panel

Admin Panel

Inventory

Filter By Transactions Type :

Show All

Print Report

Filter By Date(From - To) :

Pick Date

Pick Date

Load

Print Filtered

ID	Type	DealCustID	Grand Total	Transaction Time	TAX	Discount

Rupareka POS System Admin Panel

Admin Panel

User

Transaction

Customer Details

Search

Name

Surname

Email

Mobile

Address

Product Details

Search

Inventory

Quantity

Name

Price

Add

Transaction Time

3:00:34 PM

Wednesday, June 30, 2021

Boring Text

Price Details

Sub Total

Discount(%)

VAT

Grand Total

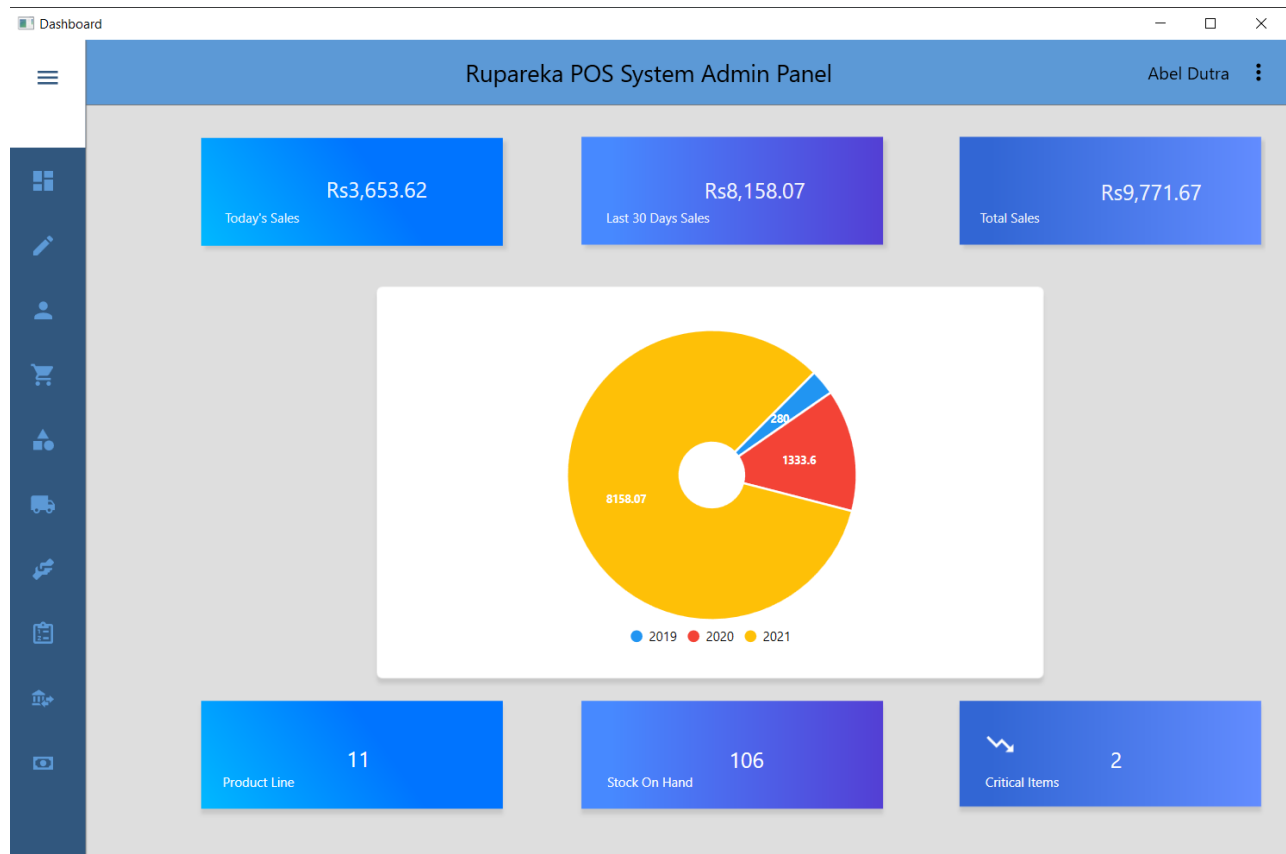
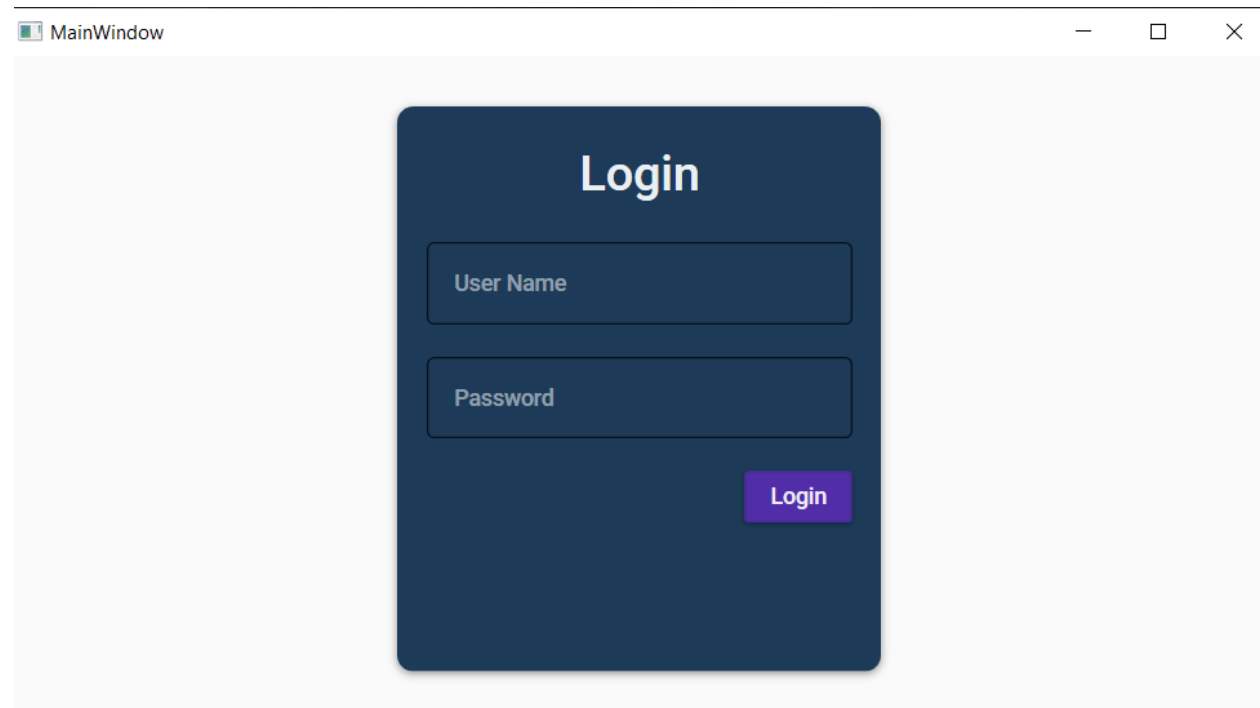
Paid Amount

Return Amount

Discard

Checkout

System Design



Dashboard

Rupareka POS System Admin Panel

Abel Dutra

Employees

Print

Name :

Password

Surname :

Password

ID Number :

Name

Date of Birth :

Pick Date

Address :

Password

Date Recived :

Pick Date

Sex :

Some item

Mobile :

Password

Sallary :

Password

Age :

Password

Search :

Password

EmpID	Name	Surname	ID	Gender	BirthDate	Age	Address	Mobile	Reciving Date	Sallary	Added Time
2011	Malaka	Bandara	4	0	6/25/2021	23	Kuliyapitiya	772145079	6/25/2021	100000	6/25/2021 3:38:27 PM
2013	Malaka	Bandara	8	0	6/11/2021	23	kuliyapitiya	772145079	6/18/2021	50000	6/26/2021 3:55:03 PM

Insert

Update

Delete

Dashboard

←

Rupareka POS System Admin Panel

Abel Dutra

User

Print

Name :

Name

Surname :

Surname

User Name :

User Name

Password :

Password

User Type :

Some item

Sex :

Some item

Date of birth :

Date of birth

Picture :

Select Image

UserID	Name	Surname	Username	Password	User Type	Gender	Birth Date	Img	isActive	Added Time
65	Malaka	jayakody	root	toor	Admin	Male	1998/07/26	POS_USER238.jpg	True	Jun 29 2021 7:25
66	Pramila	Krishan	admin	admin	Admin	Male	1998/6/17	POS_USERS543.JPG	True	Jun 30 2021 7:55

Insert

Update

Delete

Dashboard

Home

Employee

Users

Product

Categories

MDealers

Customers

Inventory

Transactions

Sale

Rupareka POS System Admin Panel

Abel Dutra

Products

Manufacturer :

Manufacturer

Full Name :

Full Name

Price :

Price

Code :

Code

Release Year :

Release Year

Barcode :

Barcode

Discription :

Discription

Model :

Model

Category :

Some item

Dealer :

Some item

Warranty :

Warranty

Quantity :

Quantity

Recorder Level :

Recorder Level

Image :

Select Image

Search Product

Search Product

PID	PCode	Brand	Model	Full Name	Price	Category	Description
8	P101	MSI	RTX 2080 Ti	Nvidia GeForce RTX 2080 Ti	999.00	Graphics Cards	GPU Cores: ↑
9	P102	AMD	Radeon RX Vega 56 8GB	AMD Radeon RX Vega 56 8GB	850.00	Graphics Cards	GPU Cores: ↑
10	P103	Intel	Core i7-9700K	Intel Core i7-9700K (9th Gen)	350.00	CPU Processors	Architecture
12	P104	CORSAIR	VENGEANCE LED 16GB (2x8GB)	CORSAIR VENGEANCE LED 16GB (2x8GB) DDR4	250.00	RAM	CORSAIR VE
13	P105	AMD	Ryzen 7 2700X	AMD Ryzen 7 2700X Processor	280.00	CPU Processors	8 Cores/16 T Frequency: 4 ↓

Insert

Update

Delete

Dashboard

Home

Employee

Users

Product

Categories

MDealers

Customers

Inventory

Transactions

Sale

Rupareka POS System Admin Panel

Abel Dutra

Categories

Catogry ID :

Category ID

Title :

Title

Description :

Description

Search :

Search

CatID	Title	Description	Added Time	Added By
1	Personal Computers	Computers	6/9/2020 6:41:31 PM	3
2	Laptops	Laptops and Notebooks	5/27/2020 12:38:16 AM	1
3	RAM	Random Access Memory	7/22/2020 1:18:07 AM	6
4	Graphics Cards	Graphics Cards	7/15/2020 2:45:21 PM	6
5	CPU Processors	Processors	7/15/2020 1:25:14 PM	48
6	Laptops	Laptops and Notebooks	6/26/2020 1:28:53 AM	6
7	Power Supliers	PSU	5/27/2020 7:36:36 PM	1
9	Keuhnarde	Keuhnarde	5/27/2020 7:36:43 PM	1

Insert

Update

Delete

Dashboard

←

Home

Employee

Users

Product

Categories

MDealers

Customers

Inventory

Transactions

Sale

Rupareka POS System Admin Panel

Abel Dutra

Dealers

ID :

ID Number

Company :

Company

Person :

Person

Mobile :

Mobile

Email :

Email

Address :

Address

Search :

Search

Print

DealID	Company Name	Distributor	Email	Mobile	Address	Added Time	Added By
1	Company 1	Giorgi Peikra	giorga@gmail.com	577 10 12 13	Tbilisi	7/20/2020 12:00:00 AM	6
2	Company 2	Guga Okita	gokit@gmail.com	599 20 11 12	Tbilisi	7/22/2020 12:00:00 AM	6
4	DHL	Erekle Omia	eoe@gmail.com	574 10 10 01	Tbilisi	7/22/2020 12:00:00 AM	6

Insert

Update

Delete

Dashboard

←

Home

Employee

Users

Product

Categories

MDealers

Customers

Inventory

Transactions

Sale

Rupareka POS System Admin Panel

Abel Dutra

Customers

ID :

ID

Name :

Name

SureName :

Password

Mobile :

Mobile

Email :

Email

Address :

This is a text area

Search :

Search

Print

CID	Name	Surname	Email	Mobile	Address	Added Time	Added By
1	Guest	Customer	guest@gmail.com	557771122	Tbilisi	7/22/2020 9:12:28 PM	6
2	Giorgi	Peikra	gp@gmail.com	557771122	თბილისი	7/28/2020 12:35:44 AM	6
8	Guga	Okitashvili	Comp2@yahoo.com	559112333	თბილისი	6/13/2021 11:13:33 PM	6
10	hashd	asdsa	root@gmail.com	0772145079	asdafdgdfghgfhf	6/30/2021 5:29:06 PM	0

Insert

Update

Delete

Dashboard

←

Home

Employee

Users

Product

Categories

MDealers

Customers

Inventory

Transactions

Sale

Rupareka POS System Admin Panel

Abel Dutra

Inventory

Filter By Category :

Some item

Show All

Print Report

PID	PCode	Brand	Model	Full Name	Price	Category	D
8	P101	MSI	RTX 2080 Ti	Nvidia GeForce RTX 2080 Ti	999.00	Graphics Cards	G
9	P102	AMD	Radeon RX Vega 56 8GB	AMD Radeon RX Vega 56 8GB	850.00	Graphics Cards	G
10	P103	Intel	Core i7-9700K	Intel Core i7-9700K (9th Gen)	350.00	CPU Processors	A
12	P104	CORSAIR	VENGEANCE LED 16GB (2x8GB)	CORSAIR VENGEANCE LED 16GB (2x8GB) DDR4	250.00	RAM	C
13	P105	AMD	Ryzen 7 2700X	AMD Ryzen 7 2700X Processor	280.00	CPU Processors	8
15	P106	Intel	Core i9-9900K	Intel Core i9-9900K Desktop Processor 8 Cores	1350.00	Personal Computers	8
16	P107	Intel	Core i9-9700	Intel Core i9-9700	800.00	CPU Processors	Ir
17	P108	Intel	Core i9-8900	Intel Core i9-8900	999.00	CPU Processors	Ir
1019	P109	Intel	Core i5-9400F	Intel Core i5-9400F	480.00	CPU Processors	Ir

Dashboard

←

Home

Employee

Users

Product

Categories

MDealers

Customers

Inventory

Transactions

Sale

Rupareka POS System Admin Panel

Abel Dutra

Inventory

Filter By Transactions Type :

Some item

Show All

Print Report

Filter By Date(From - To) :

Pick Date

Pick Date

Load

Print Filtered

ID	Type	DealCustID	Grand Total	Transaction Time	TAX	Discount
17	Sale	2	280.00	7/9/2019 12:00:00 AM	0.00	0.00
18	Purchase	1	7990.00	7/9/2019 12:00:00 AM	0.00	0.00
4117	Sale	5	480.00	7/28/2020 11:24:27 PM	0.00	0.00
4118	Sale	5	853.60	7/30/2020 4:51:51 PM	0.00	3.00
4119	Sale	1	1281.55	6/13/2021 10:46:15 PM	0.00	5.00
4120	Purchase	0	815.58	6/15/2021 1:27:04 PM	1.00	5.00
4121	Sale	0	2097.90	6/21/2021 11:29:11 AM	5.00	0.00
4122	Sale	1	1125.00	6/21/2021 11:30:03 AM	0.00	10.00
5121	Sale	0	3653.62	6/30/2021 12:17:20 PM	4.00	5.00
5122	Sale	0	0.00	6/30/2021 12:18:29 PM	0.00	0.00

←

Dashboard

Home

Employee

Users

Product

Categories

MDealers

Customers

Inventory

Transactions

Sale

Rupareka POS System Admin Panel

Abel Dutra

User

Transaction 0000000000

Customer Details

Search

Name

Surname

Email

Mobile

Address

Product Details

Search

Inventory

Quantity

Name

Price

Add

Transaction Time

10:56:53 PM

Wednesday, June 30, 2021

Boring Text

Price Details

Sub Total

Discount(%)

VAT

Grand Total

Paid Amount

Return Amount

Sub Total

Discount

VAT

Grand Total

Paid Amount

Return Amount

1

2

3

4

5

6

7

8

9

.

0

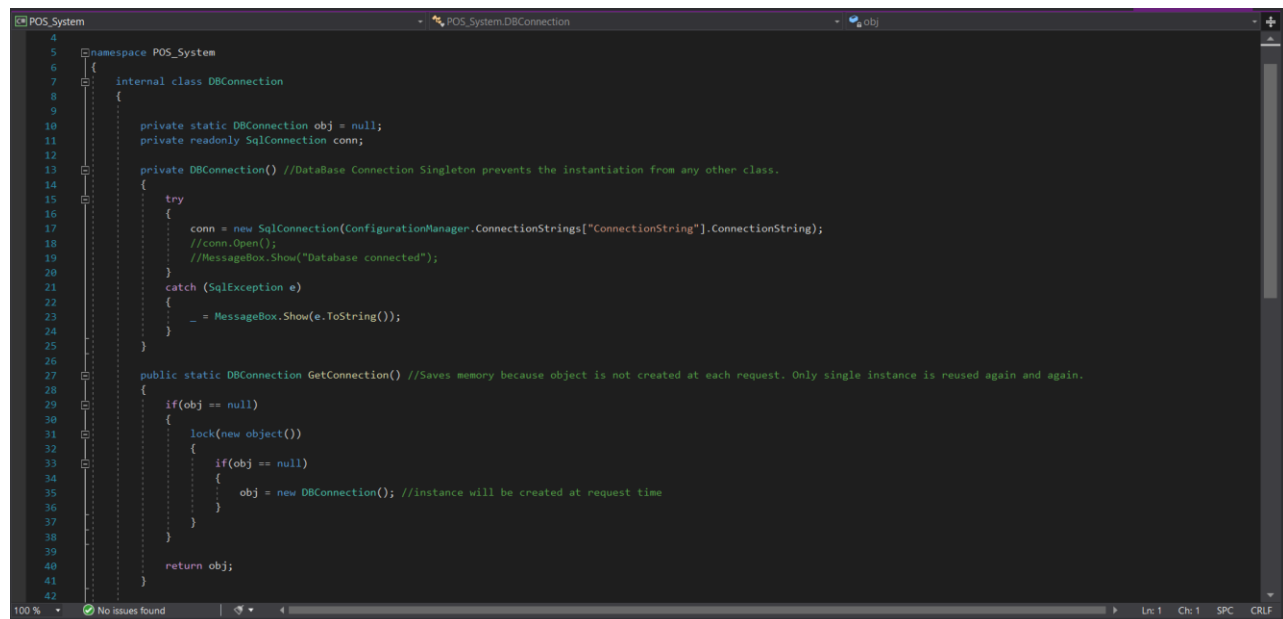
X

Discard

Checkout

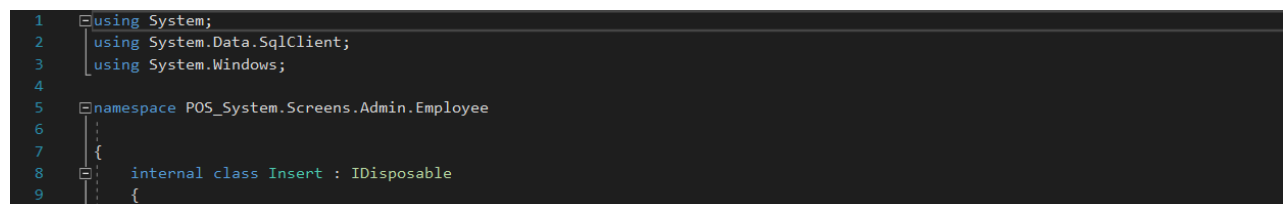
Programming techniques of designing the application

- 1) Singleton pattern is that has only one instance and provides a global point of access to it. The advantage of singleton pattern is saving memory. Because there is not crated request to each request. There is Only single instance is reused again and again. singleton pattern mostly use in multi-threaded and database applications. we used singleton pattern in this application to connect the database. When creating singleton pattern, we use the best practice method of “double checked locking”.



```
4  namespace POS_System
5  {
6      internal class DBConnection
7      {
8      {
9
10         private static DBConnection obj = null;
11         private readonly SqlConnection conn;
12
13         private DBConnection() //DataBase Connection Singleton prevents the instantiation from any other class.
14         {
15             try
16             {
17                 conn = new SqlConnection(ConfigurationManager.ConnectionStrings["ConnectionString"].ConnectionString);
18                 //conn.Open();
19                 //MessageBox.Show("Database connected");
20             }
21             catch (SqlException e)
22             {
23                 _ = MessageBox.Show(e.ToString());
24             }
25         }
26
27         public static DBConnection GetConnection() //Saves memory because object is not created at each request. Only single instance is reused again and again.
28         {
29             if(obj == null)
30             {
31                 lock(new object())
32                 {
33                     if(obj == null)
34                     {
35                         obj = new DBConnection(); //instance will be created at request time
36                     }
37                 }
38             }
39             return obj;
40         }
41     }
42 }
```

- 2) According to the memory optimization of this application, we use IDisposable Interface to release unmerged resources in this application. If this is done by the garbage collector, before optimizing the memory by the garbage collector, we manually optimize the memory when we create a new object and after the process of that object.



```
1  using System;
2  using System.Data.SqlClient;
3  using System.Windows;
4
5  namespace POS_System.Screens.Admin.Employee
6  {
7      {
8          internal class Insert : IDisposable
9          {
```

```

74     protected virtual void Dispose(bool disposing)
75     {
76         if (!disposedValue)
77         {
78             if (disposing)
79             {
80                 // TODO: dispose managed state (managed objects)
81             }
82
83             // TODO: free unmanaged resources (unmanaged objects) and override finalizer
84             // TODO: set large fields to null
85             disposedValue = true;
86         }
87     }
88
89     // // TODO: override finalizer only if 'Dispose(bool disposing)' has code to free unmanaged resources
90     // ~Insert()
91     // {
92     //     // Do not change this code. Put cleanup code in 'Dispose(bool disposing)' method
93     //     Dispose(disposing: false);
94     // }
95
96     public void Dispose()
97     {
98         // Do not change this code. Put cleanup code in 'Dispose(bool disposing)' method
99         Dispose(disposing: true);
100        GC.SuppressFinalize(this);
101    }
102 }
103
104

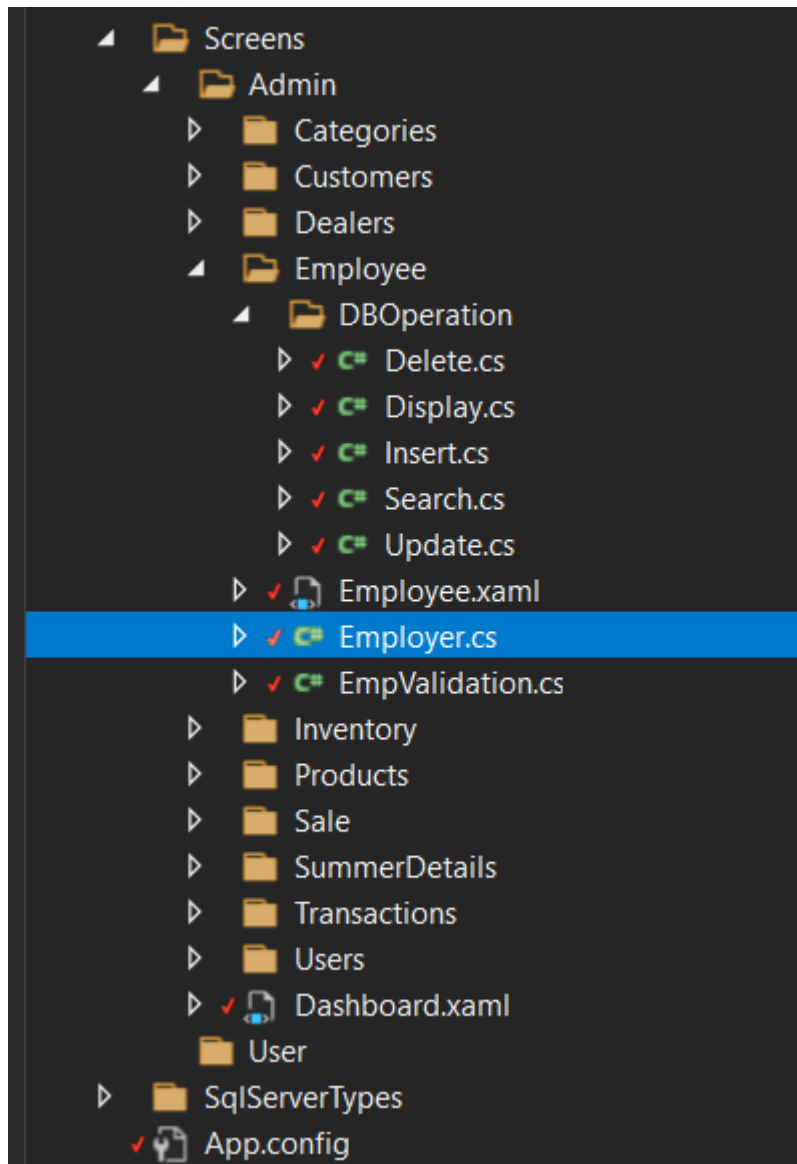
```

```

33
34     if (!EmpValidation.IsValidate(name.Text, idnum.Text, address.Text, mobile.Text, sname.Text, dob.Text, rdate.Text, sex.SelectedIndex.
35     {
36         Employer newEmp = new Employer(name.Text, int.Parse(idnum.Text), address.Text, int.Parse(mobile.Text), sname.Text, dob.Text, rda
37
38         Insert insert_query = new Insert(newEmp);
39         insert_query.Insert_Query();
40         insert_query.Dispose();
41         DisplayData();
42     }
43
44

```

- 3) When develop this application we try to follow under the SOLID Principle and do not violate the SOLID Principle. Mostly use Single-responsiblity Principle in this application. As an example, we use one model class for build a one employer and according to the database operation with that employer we build one by one classes. We have created one class for insert operation, one class for delete operation and one class for update operation.



Bibliography

Material Design In XAML Toolkit. 2021. Material Design In XAML. [ONLINE] Available at <http://materialdesigninxaml.net/> [Accessed 30 June 2021].

Live Charts. 2021. Live Charts. [ONLINE] Available at <https://lvcharts.net/> [Accessed 30 June 2021].

www.javatpoint.com. 2021. Singleton Design Patterns - Javatpoint. [ONLINE] Available at <https://www.javatpoint.com/singleton-design-pattern-in-java> [Accessed 30 June 2021].

YouTube. 2021. "C# GUI Tutorial using WPF | XAML | - Windows Presentation Foundation - YouTube. [ONLINE] Available at <https://www.youtube.com/watch?v=oSeYvMEH7jc&t=274s> [Accessed 30 June 2021].