CRM SYSTEM

Project Report

IS204.3 Enterprise System

Lecturer:Mrs.Chalani Oruthotaarachchi

Group M

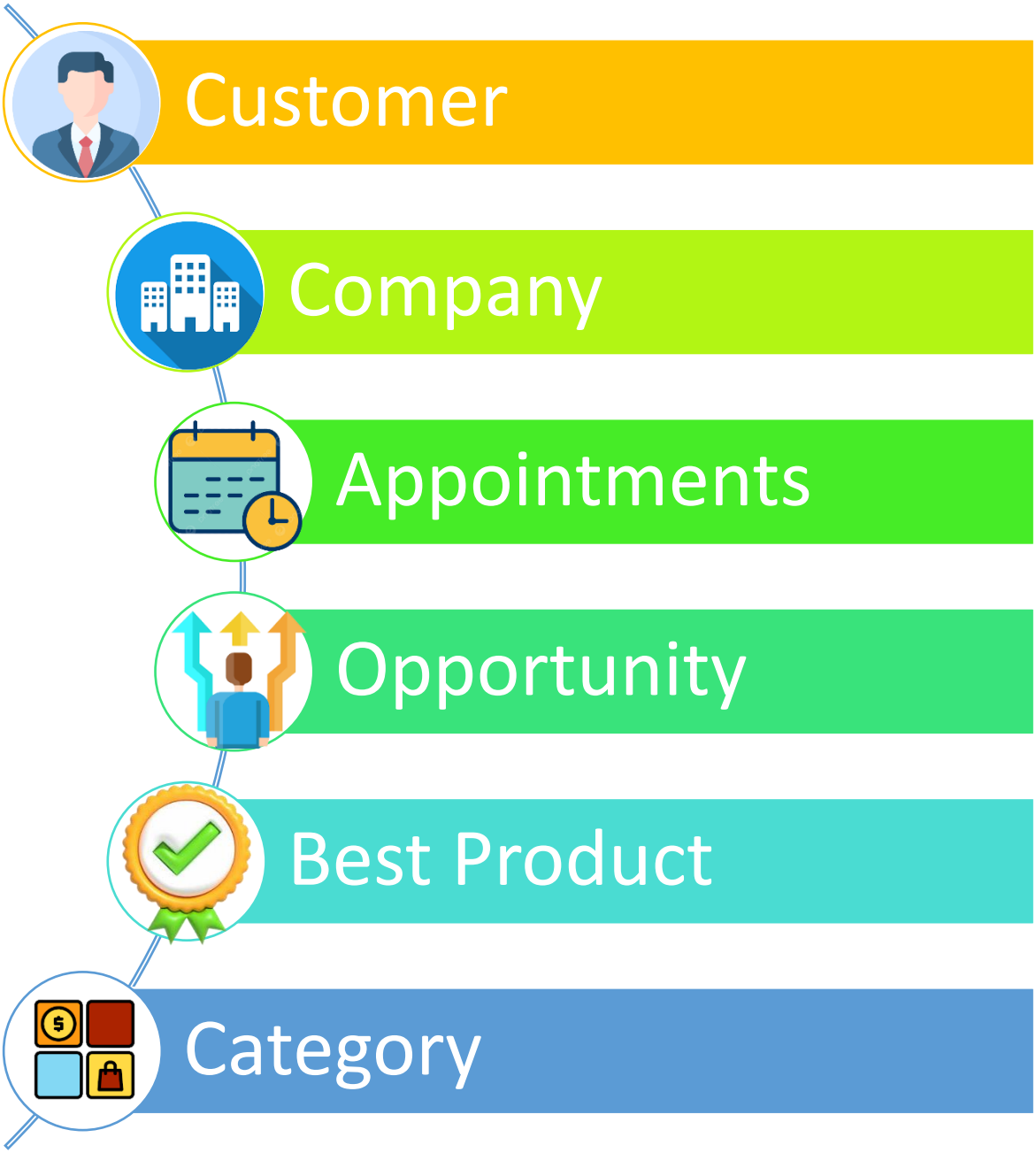
TABLE OF CONTENTS

01. System Introduction

CRM system, or Customer Relationship Management system, is a powerful tool that allows businesses to manage their interactions with customers and prospects. It provides a centralized database to store customer information and allows businesses to track and analyze customer interactions and behavior over time. By providing a comprehensive view of the customer, a CRM system can help businesses better understand and meet their customers' needs, improve customer satisfaction, and ultimately drive sales and revenue growth. A CRM system can also help businesses streamline their internal processes, automate tasks, and improve communication across teams. Overall, a CRM system is an essential tool for businesses looking to build and maintain strong relationships with their customers.

CRM system, or Customer Relationship Management system, is a powerful tool that allows businesses to manage their interactions with customers and prospects. It provides a centralized database to store customer information and allows businesses to track and analyze customer interactions and behavior over time. By providing a comprehensive view of the customer, a CRM system can help businesses better understand and meet their customers' needs, improve customer satisfaction, and ultimately drive sales and revenue growth. A CRM system can also help businesses streamline their internal processes, automate tasks, and improve communication across teams. Overall, a CRM system is an essential tool for businesses looking to build and maintain strong relationships with their customers.

02. Main type of function in The system



03. Identification of the system functions

- User – The person who used the CRM system. The user has done every task in this system. He enters every data to the system.
- customer- The owners of the companies. This system can be used by many companies, so that many customers have in this system. In this system the customer can get appointments from the user and the user gives appointments to the customer.
- Company – The organizations who buy goods or services from the users.
- Appointments – In this function, the company or customer fixes the date with users for an important thing.

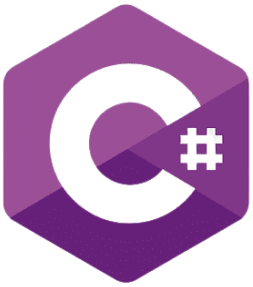
  Example-: special meeting
  New product discussions
  Discussion of the product quality
- Opportunity – Opportunity is created for users by companies or customers.

  Examples-: if there two companies order same products, one of those companies.
- Best product – It's included recommended ideas according to the company side.

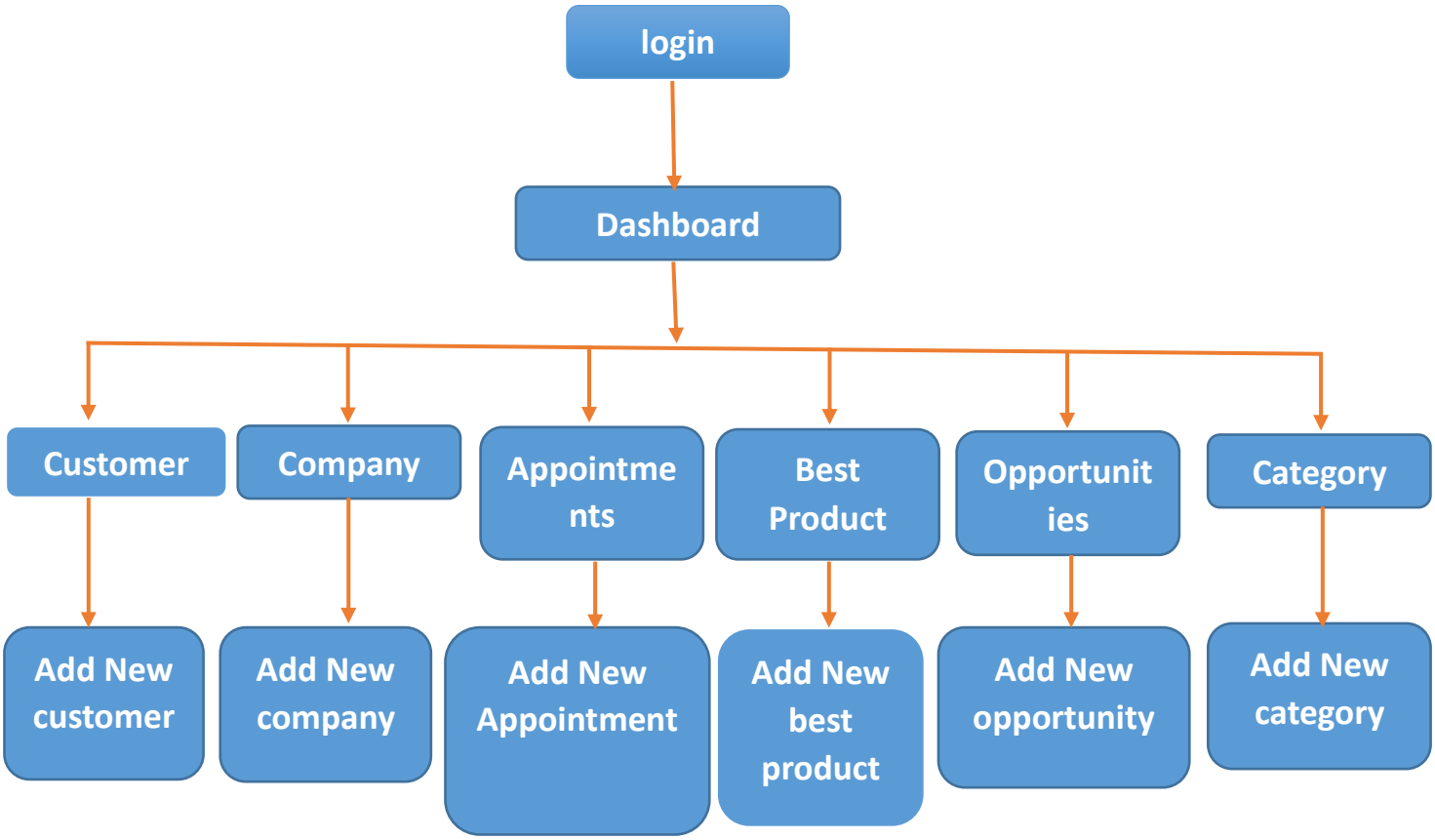
  Example-: product ranges
- Category – Thats can be used for any types of fields, such as transportation, health field, agricultural field etc.

04. How we build the system

- ✓ First, we discussed the system that we are going to build from ERP, SCM, CRM, KM, then we selected to build a CRM system.
- ✓ Next, we designed rough sketches of UI interfaces. Then we started to design UI and cording part of UI's, using **visual studio software**,
- ✓ Then we use **SQL server** to store our data, we store data as well as retrieve data through this server but it's difficult to deal with server, so we decided to make database tables by using **SQL server management studio**.
- ✓ Finally, we used software such as visual studio, SQL server, SQL server management studio to build this CRM system.

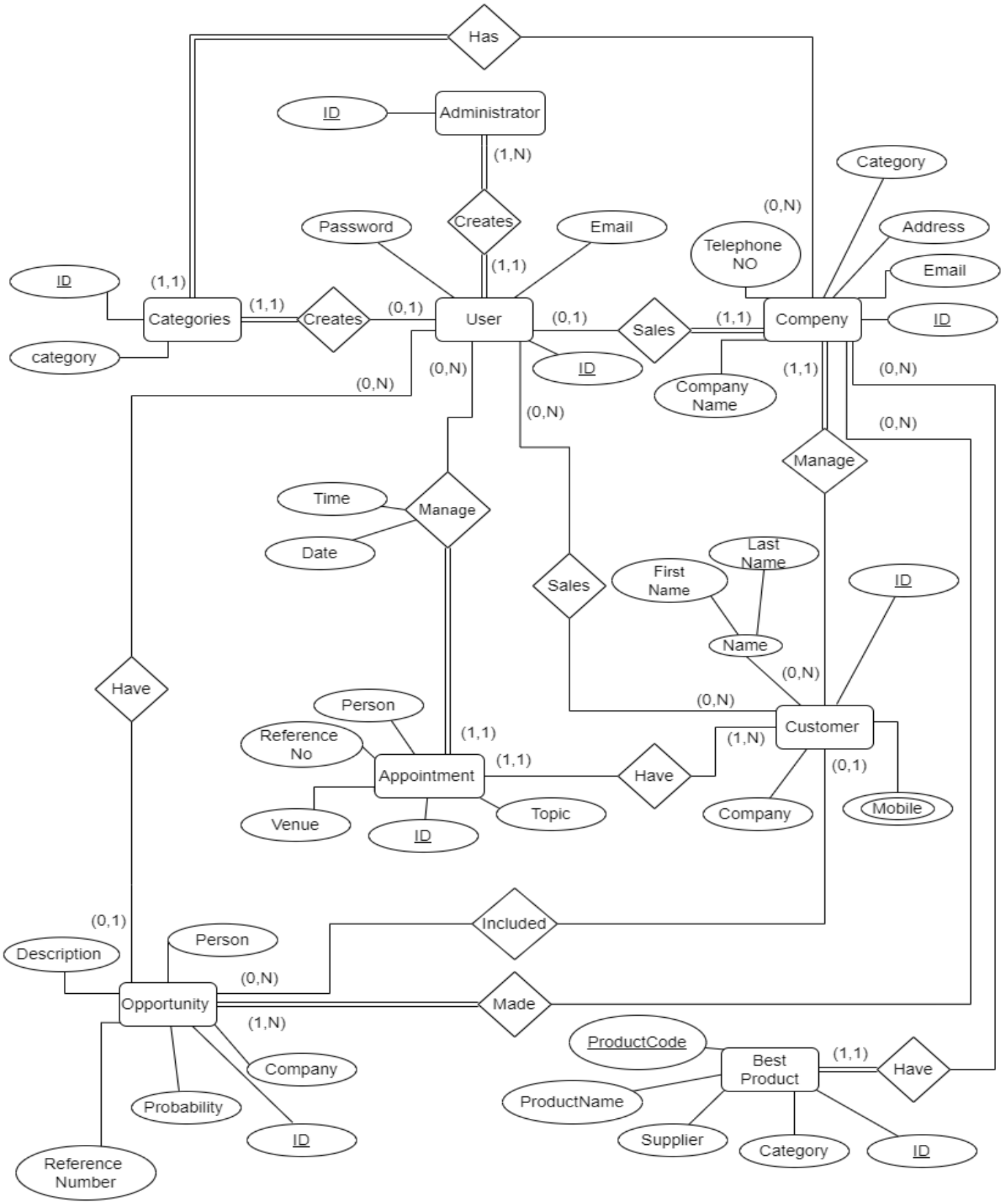


05. Flow of the system



06. Diagrams

I. ER- Diagram

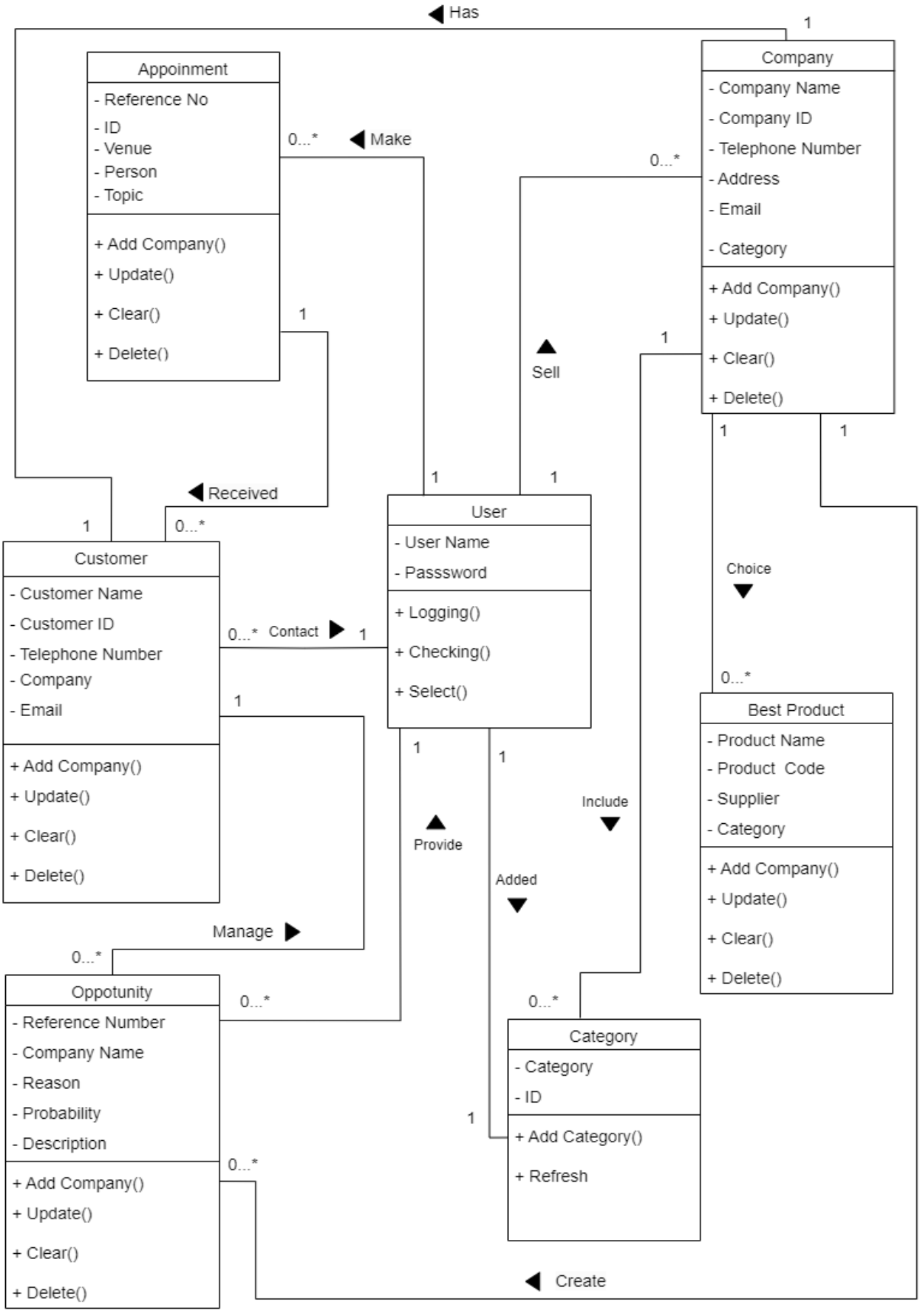


Assumptions

- If we sell a product, we act as a user in this CRM System.
- If the user doesn't have an account in this CRM System, the user can contact the administrator and create an account.
- In the category function, we create a new category, and each category uses in the company.
- In the company function, the company name includes use in the customer function.
- In the customer function, the first name that includes is used as a person in the appointment function.

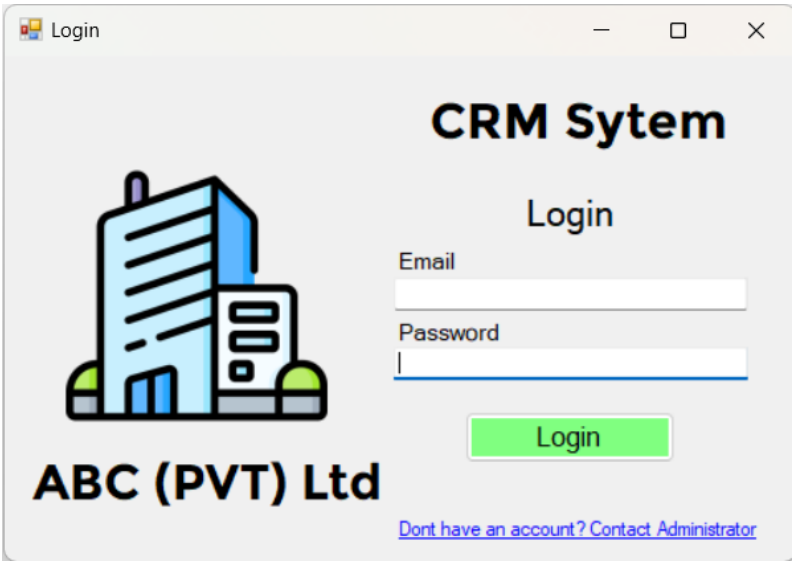
II. User Case Diagram

III. Class Diagram



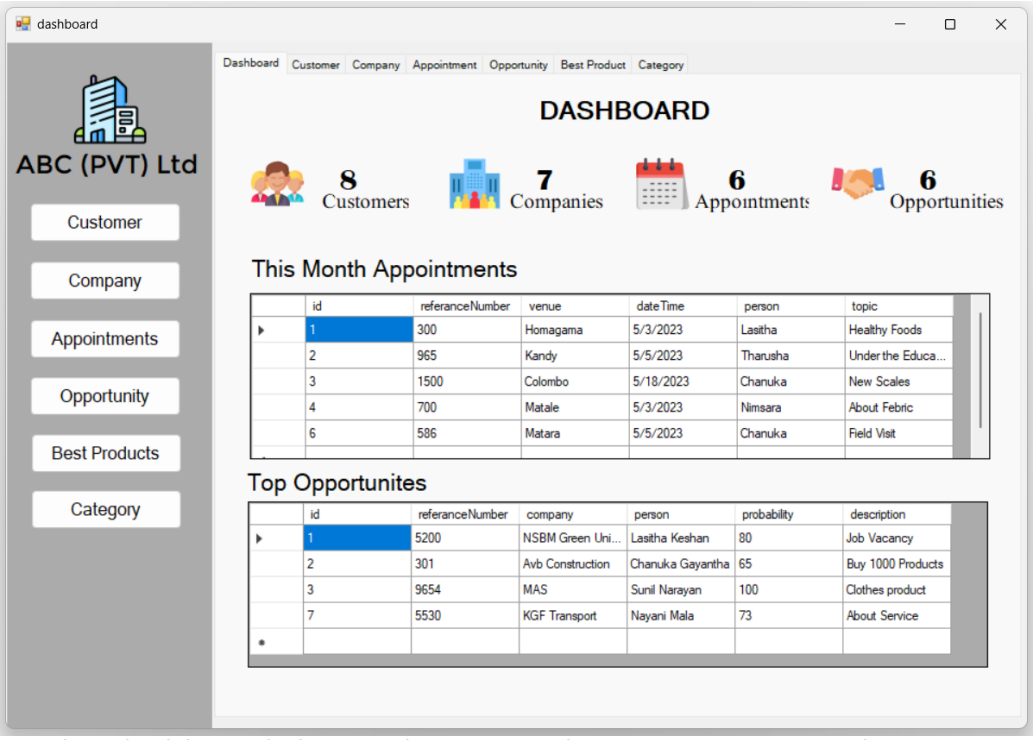
07. User Interfaces designs

Login page



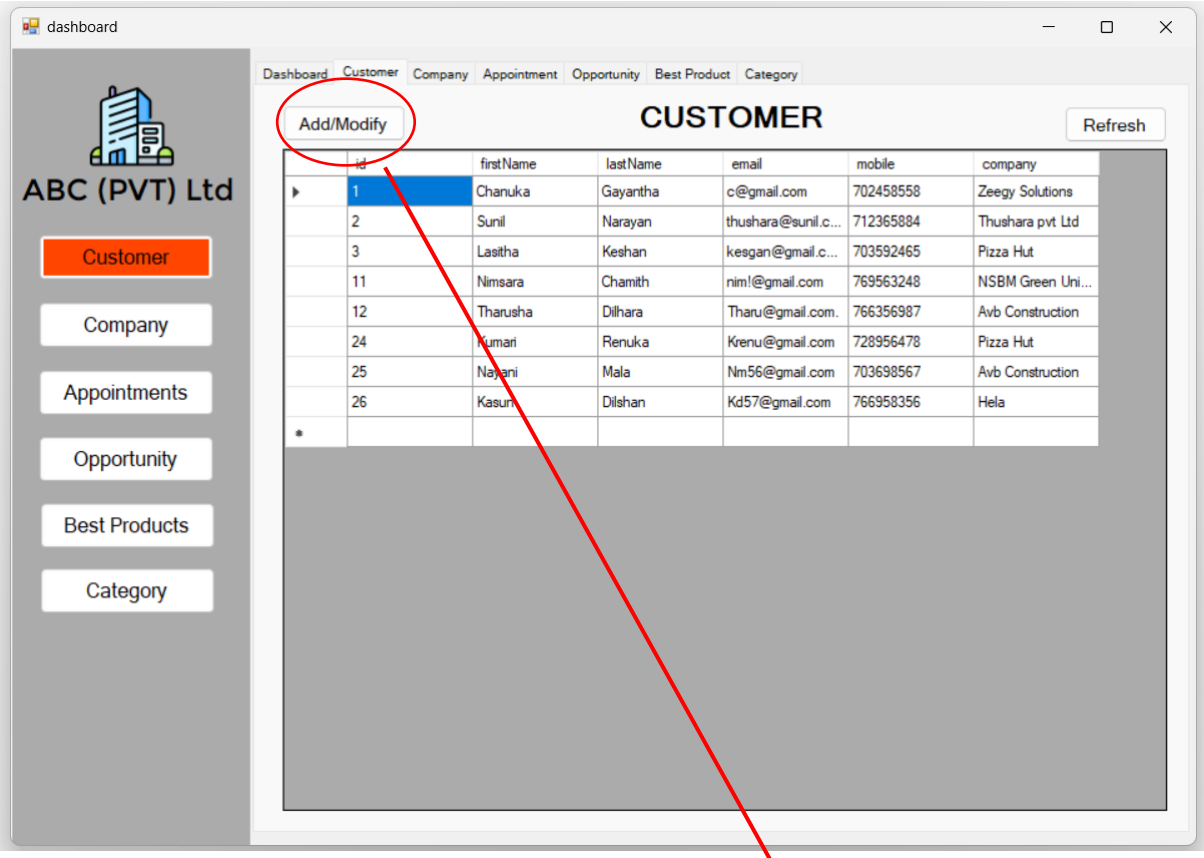
In this page user can login to the system using their password and mail. First of all, users want to create an account for this system by contacting the Administrator.

Dashboard



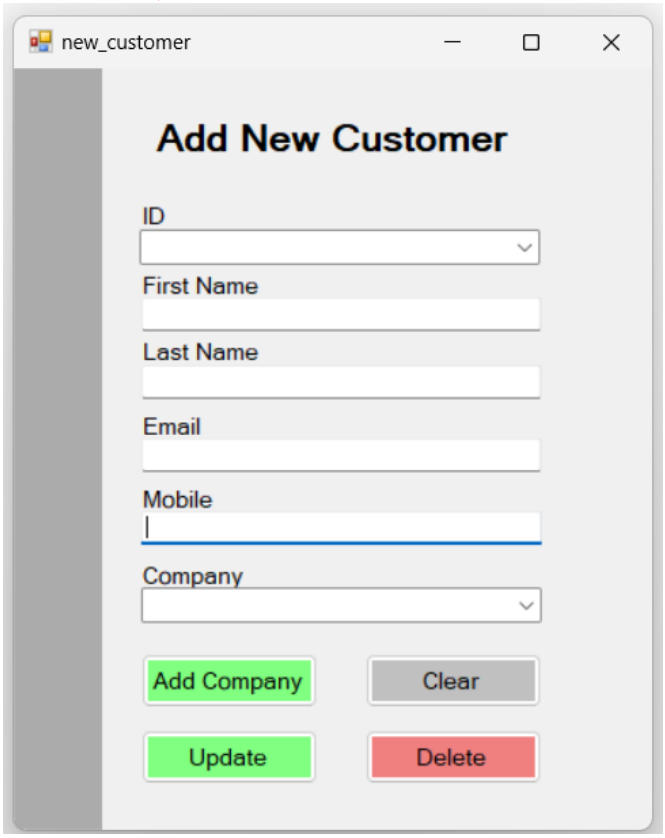
When the user logs in using email and password. Then display this dashboard. In this dashboard we can select any functions. It shows, how many customers, companies, appointments, and Opportunities in their system. Specially in this dashboard shows about month appointments and Top opportunities. It is very helpful for the users.

Customer page



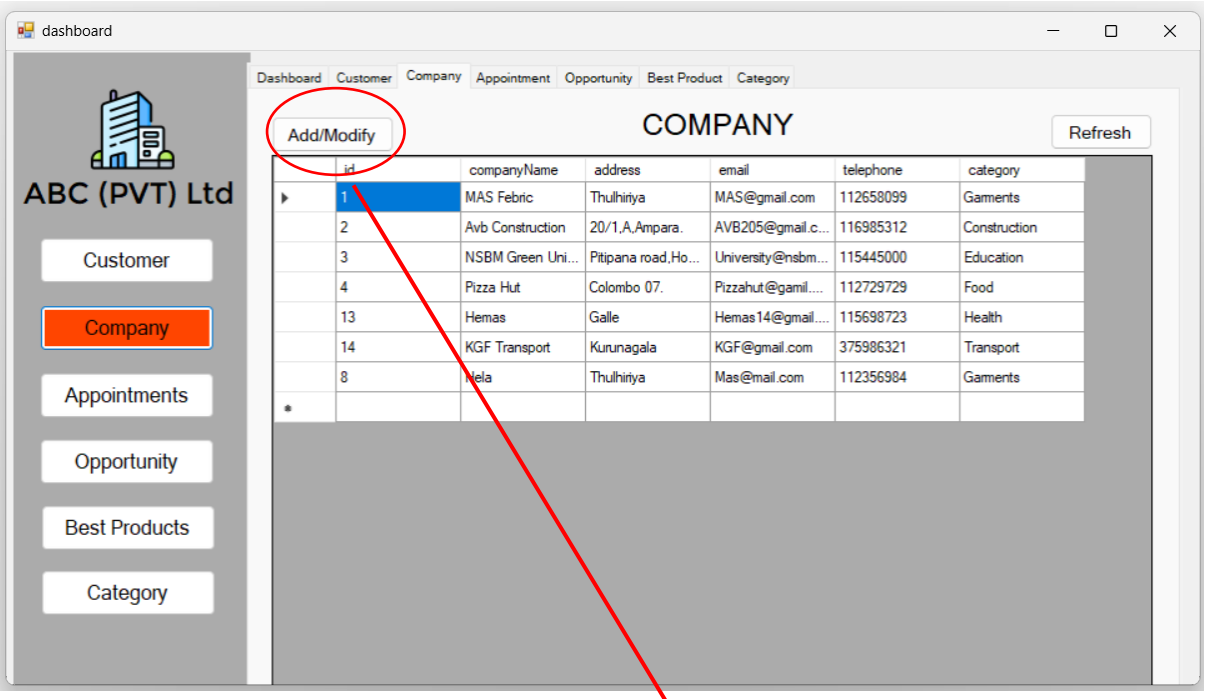
01

In this page every customer's detail shows. If we want to add a new customer to the system, then we need to go to the add or modify button. After that we can see the second user interface. In this user Interface we can add new customer, update, Delete and clear customer details.

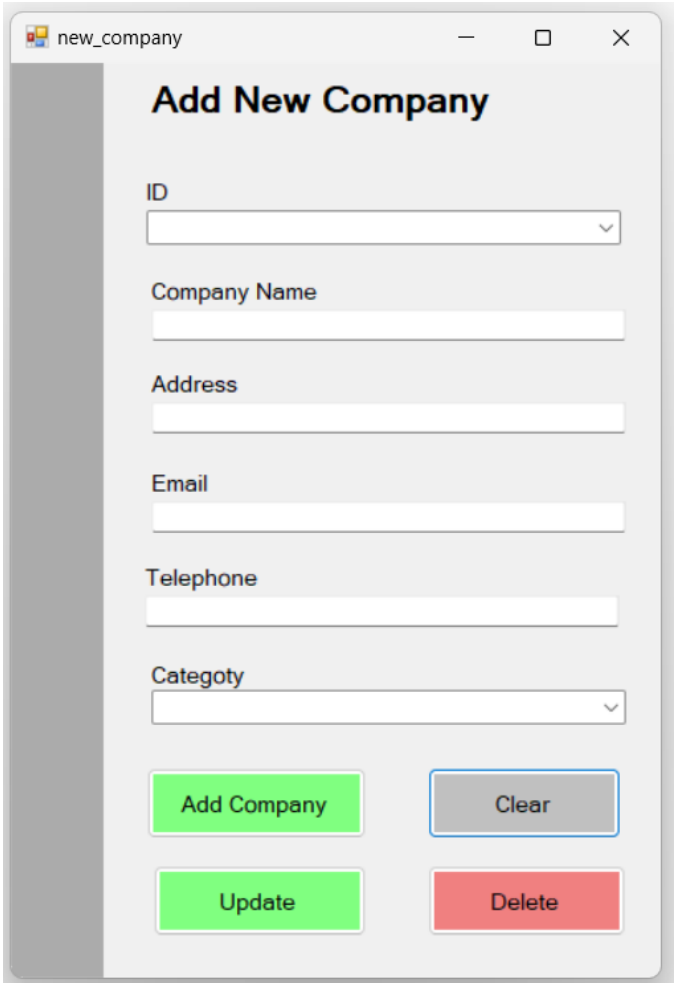


02

Company page



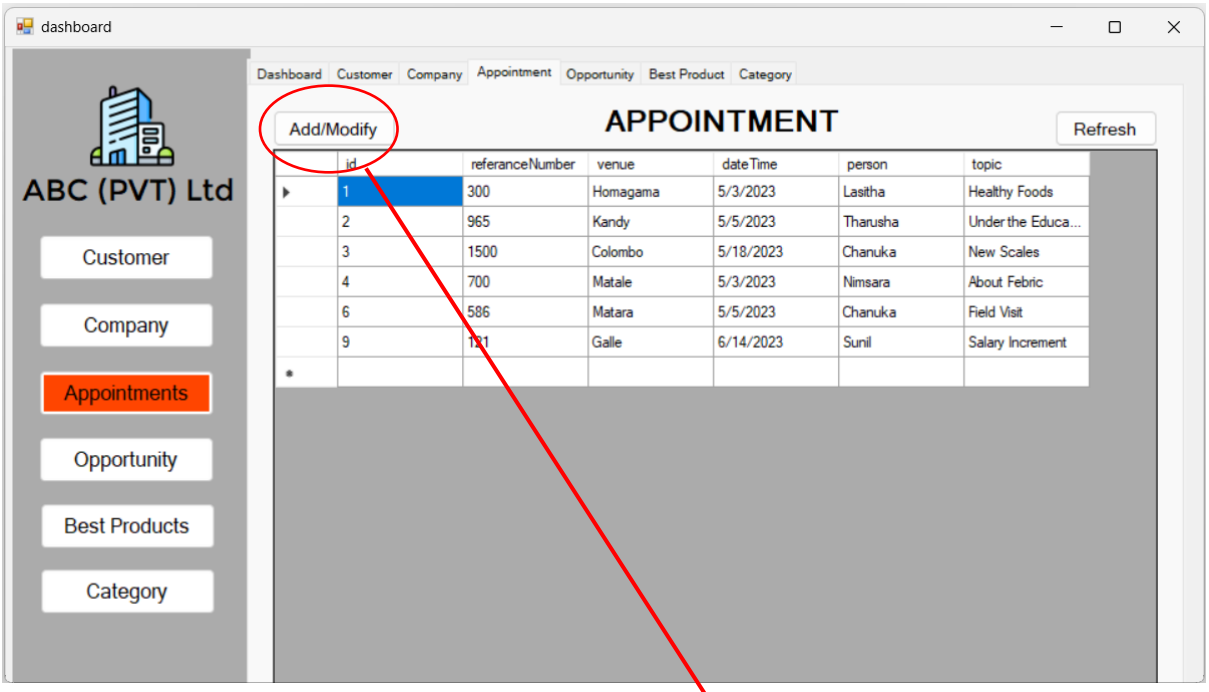
01



In this page every company's detail shows. If we want to add a new company to the system, then we need to go to the add or modify button. After that we can see the second user interface. In this user Interface we can add new company, update, Delete and clear company details.

02

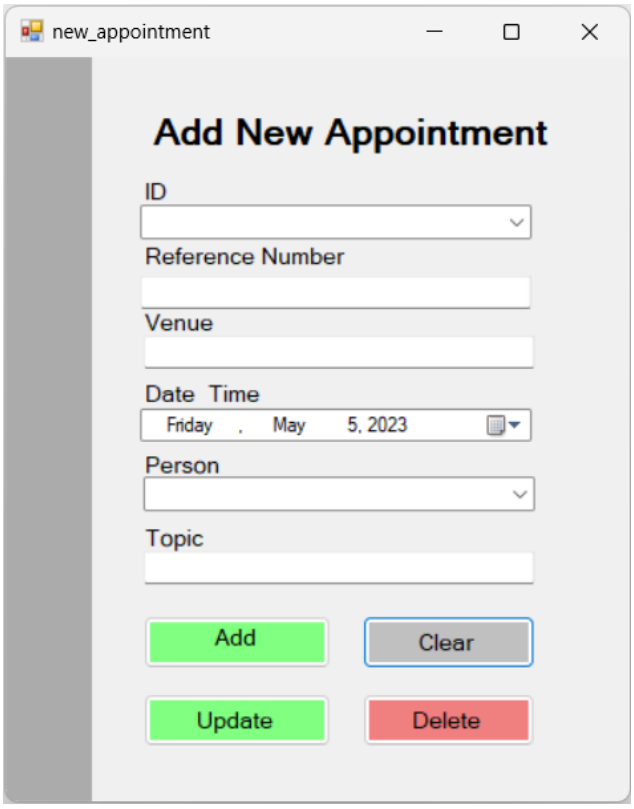
Appointment page



01

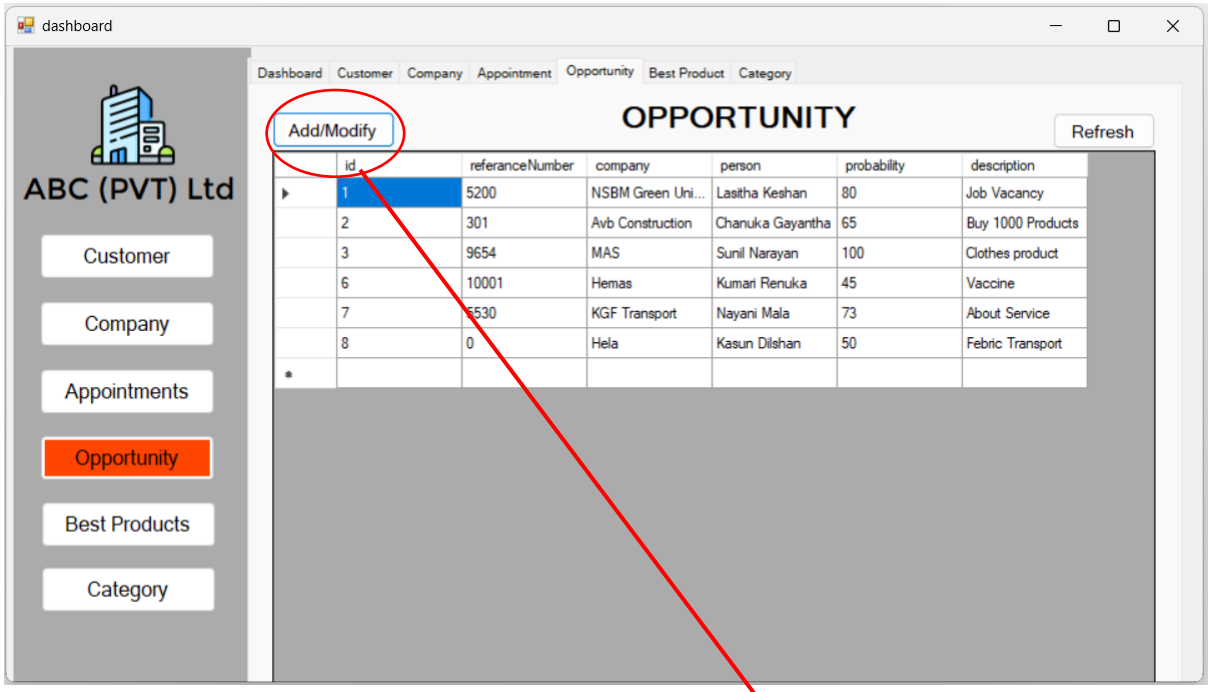
In this page every appointment's detail shows. If we want to add a new appointment to the system, then we need to go to the add or modify button. After that we can see the second user interface. In this user Interface we can add new appointment, update, Delete and clear customer details.

Every month appointments are shown on the dashboard. It is very useful to the system users.



02

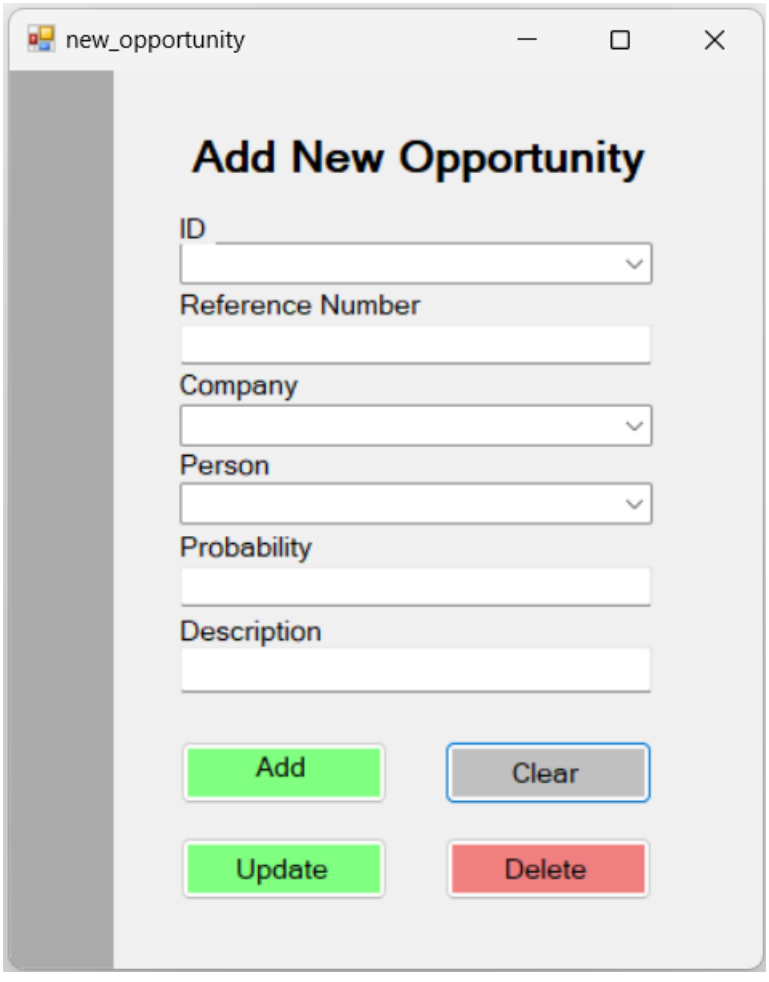
Opportunity page



01

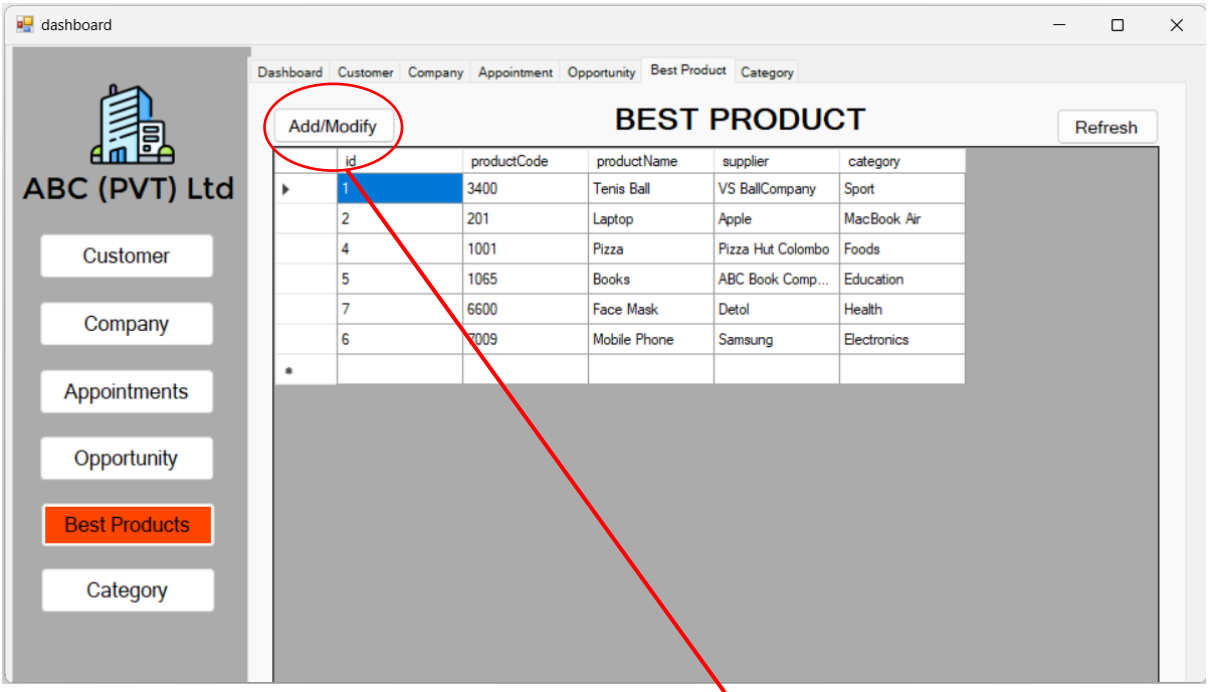
In this page every opportunity's detail shows. If we want to add a new opportunity to the system, then we need to go to the add or modify button. After that we can see the second user interface. In this user Interface we can add new opportunity, update, Delete and clear customer details.

Top opportunity is shown on the dashboard.



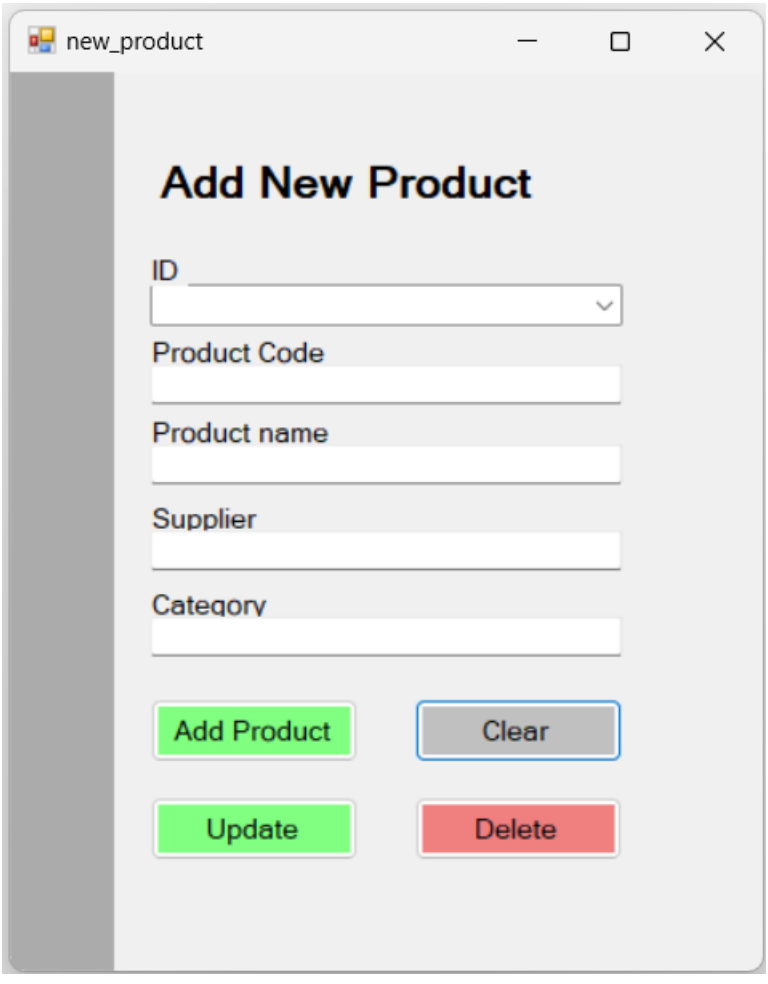
02

Best product



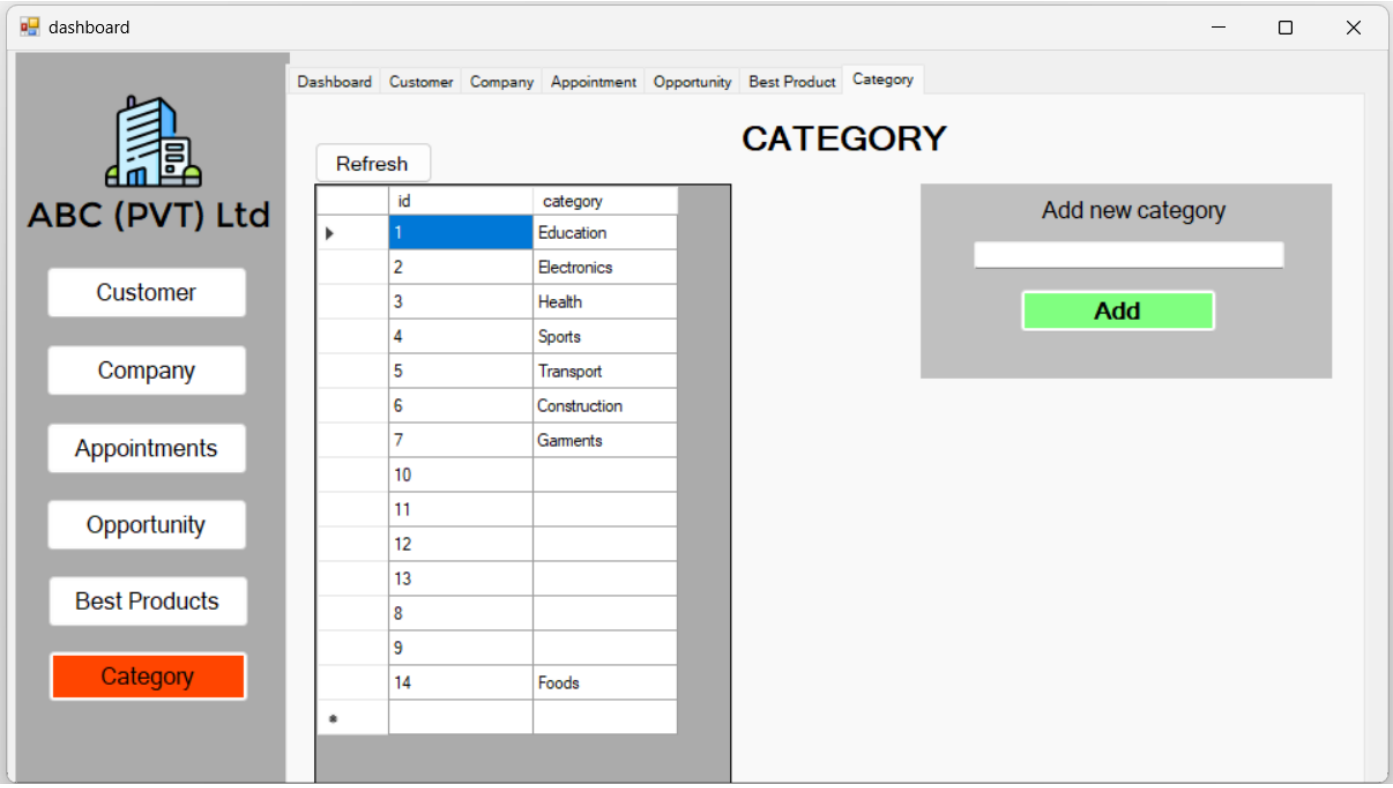
01

This page is related to the best products in their company or system. every product's detail shows in this page. If we want to add the best new product to the system, then we need to go to the add or modify button. After that we can see the second user interface. In this user Interface we can add best new product, update, Delete and clear product details.



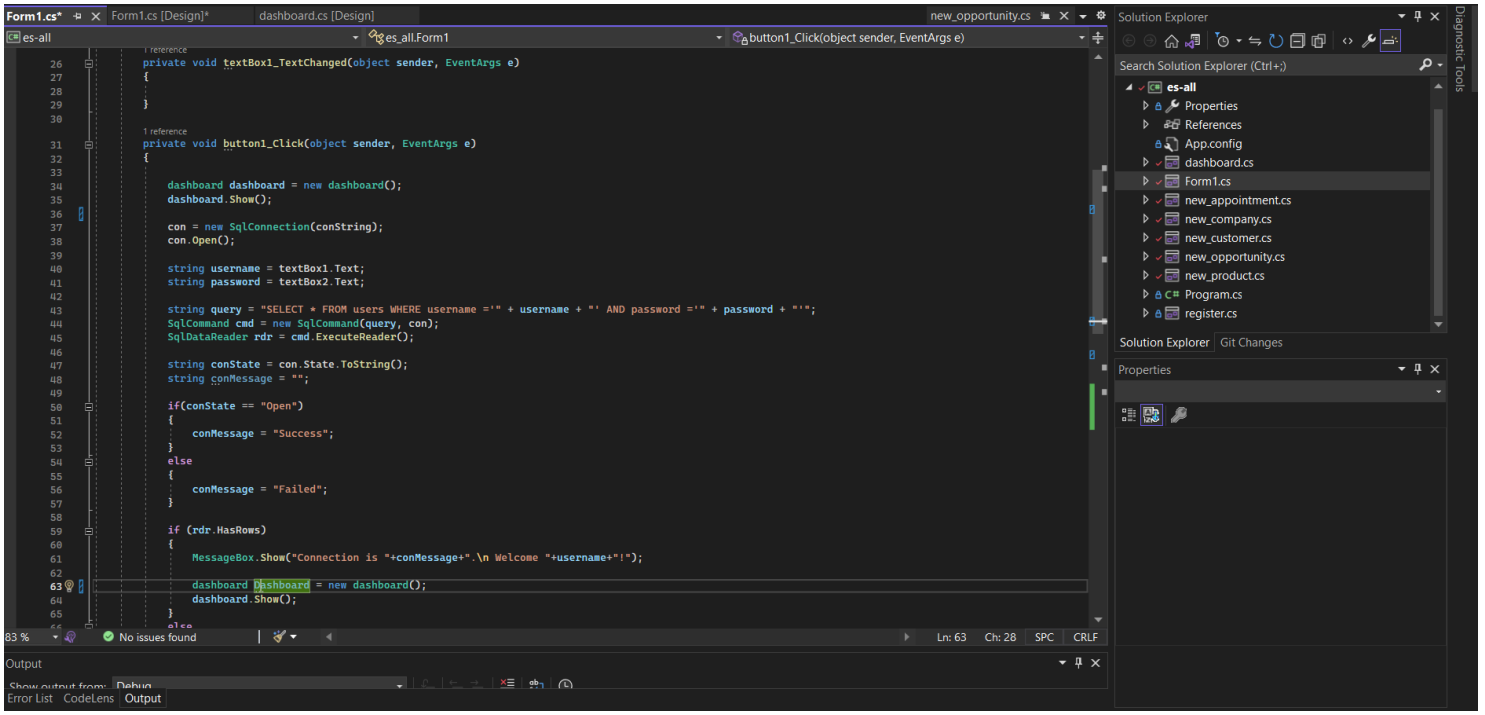
02

Category

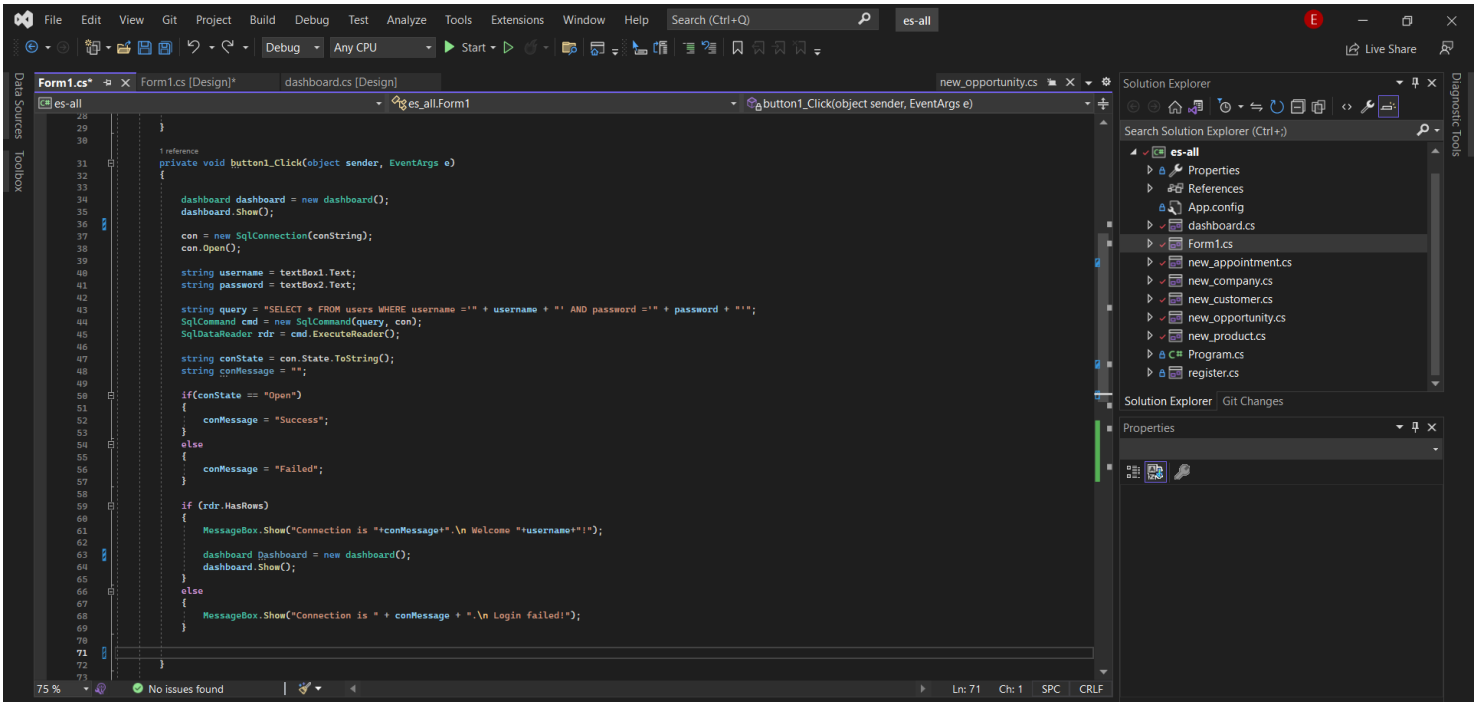


This CRM system we used in business. So, there are so many categories of products or services. This page is used for filling out all the details about categories that chosen product and services. When we add the company to the system, we can choose what categories are going on.

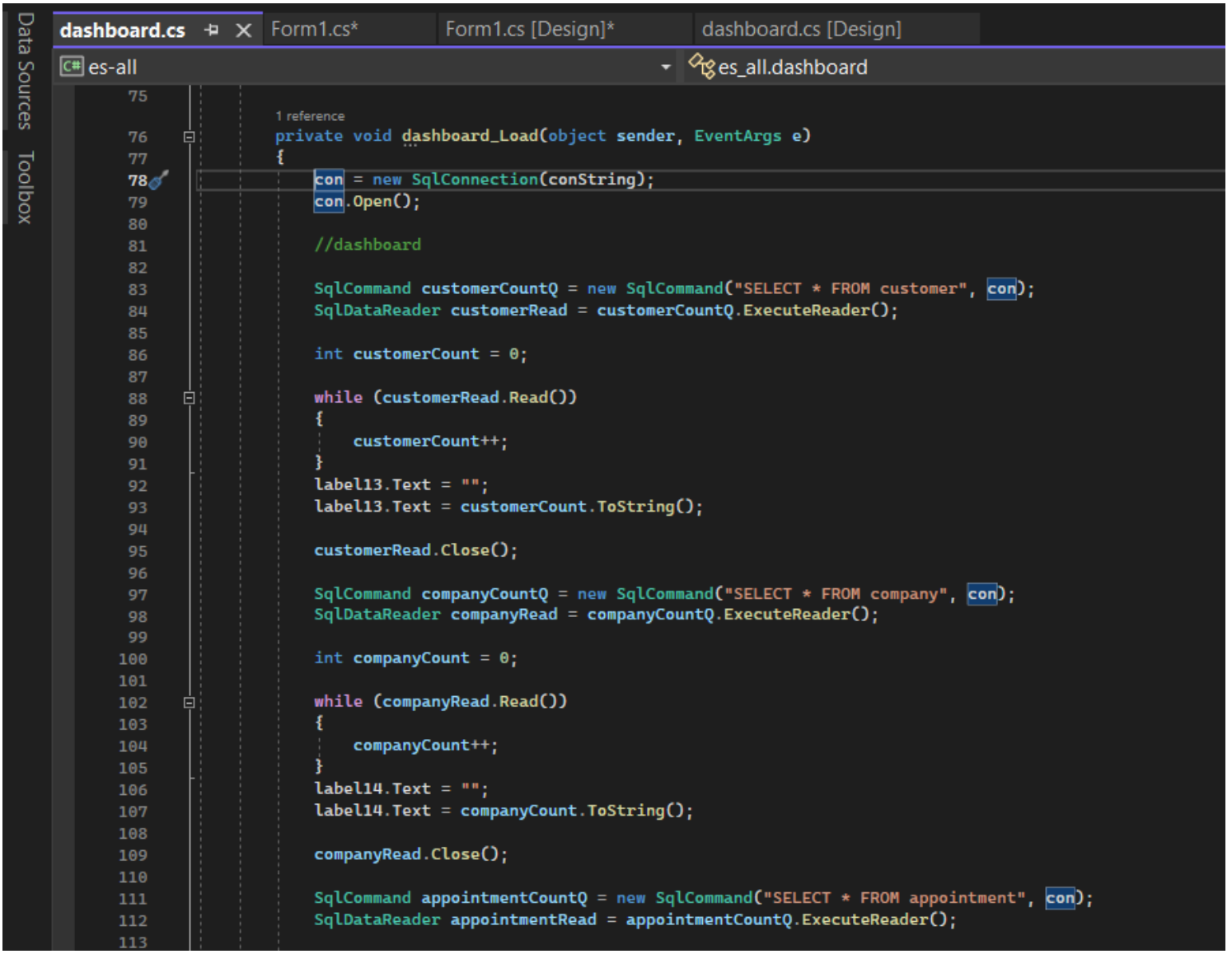
08. System back-end



```
private void textBox1_TextChanged(object sender, EventArgs e)
{

}

private void button1_Click(object sender, EventArgs e)
{

    dashboard dashboard = new dashboard();
    dashboard.Show();

    con = new SqlConnection(conString);
    con.Open();

    string username = textBox1.Text;
    string password = textBox2.Text;

    string query = "SELECT * FROM users WHERE username ='" + username + "' AND password ='" + password + "'";
    SqlCommand cmd = new SqlCommand(query, con);
    SqlDataReader rdr = cmd.ExecuteReader();

    string conState = con.State.ToString();
    string conMessage = "";

    if(conState == "Open")
    {
        conMessage = "Success";
    }
    else
    {
        conMessage = "Failed";
    }

    if (rdr.HasRows)
    {
        MessageBox.Show("Connection is "+conMessage+".\n Welcome "+username+"!");

        dashboard Dashboard = new dashboard();
        dashboard.Show();
    }
```



```
}

private void button1_Click(object sender, EventArgs e)
{

    dashboard dashboard = new dashboard();
    dashboard.Show();

    con = new SqlConnection(conString);
    con.Open();

    string username = textBox1.Text;
    string password = textBox2.Text;

    string query = "SELECT * FROM users WHERE username ='" + username + "' AND password ='" + password + "'";
    SqlCommand cmd = new SqlCommand(query, con);
    SqlDataReader rdr = cmd.ExecuteReader();

    string conState = con.State.ToString();
    string conMessage = "";

    if(conState == "Open")
    {
        conMessage = "Success";
    }
    else
    {
        conMessage = "Failed";
    }

    if (rdr.HasRows)
    {
        MessageBox.Show("Connection is "+conMessage+".\n Welcome "+username+"!");

        dashboard Dashboard = new dashboard();
        dashboard.Show();
    }
    else
    {
        MessageBox.Show("Connection is " + conMessage + ".\n Login failed!");
    }


}
```


dashboard.cs
Form1.cs*
Form1.cs [Design]*
dashboard.cs [Design]
es-all
es_all.dashboard
Data Sources
Toolbox
1 reference
private void dashboard_Load(object sender, EventArgs e)
{
con = new SqlConnection(conString);
con.Open();
//dashboard
SqlCommand customerCountQ = new SqlCommand("SELECT * FROM customer", con);
SqlDataReader customerRead = customerCountQ.ExecuteReader();
int customerCount = 0;
while (customerRead.Read())
{
customerCount++;
}
label13.Text = "";
label13.Text = customerCount.ToString();
customerRead.Close();
SqlCommand companyCountQ = new SqlCommand("SELECT * FROM company", con);
SqlDataReader companyRead = companyCountQ.ExecuteReader();
int companyCount = 0;
while (companyRead.Read())
{
companyCount++;
}
label14.Text = "";
label14.Text = companyCount.ToString();
companyRead.Close();
SqlCommand appointmentCountQ = new SqlCommand("SELECT * FROM appointment", con);
SqlDataReader appointmentRead = appointmentCountQ.ExecuteReader();

dashboard.cs | Form1.cs* | Form1.cs [Design]* | dashboard.cs [Design]

es-all | es_all.dashboard

```
113
114         int appointmentCount = 0;
115
116         while (appointmentRead.Read())
117         {
118             appointmentCount++;
119         }
120         label15.Text = "";
121         label15.Text = appointmentCount.ToString();
122
123         appointmentRead.Close();
124
125         SqlCommand opportunityCountQ = new SqlCommand("SELECT * FROM opportunity", con);
126         SqlDataReader opportunityRead = opportunityCountQ.ExecuteReader();
127
128         int opportunityCount = 0;
129
130         while (opportunityRead.Read())
131         {
132             opportunityCount++;
133         }
134         label16.Text = "";
135         label16.Text = opportunityCount.ToString();
136
137         opportunityRead.Close();
138
139         //This moth appointments
140
141         string currentMonth = DateTime.Now.ToString("MM",CultureInfo.InvariantCulture);
142
143         string thisAppoQueary = $"SELECT * FROM appointment WHERE MONTH(dateTime) ={currentMonth} ";
144         SqlCommand loadThisAppointment = new SqlCommand(thisAppoQueary, con);
145         SqlDataAdapter thisAppointmentAdapter = new SqlDataAdapter(loadThisAppointment);
146         DataSet thisAppointmentDataSet = new DataSet();
147         thisAppointmentAdapter.Fill(thisAppointmentDataSet);
148
149         dataGridView7.DataSource = thisAppointmentDataSet.Tables[0];
150         //
151
```

dashboard.cs | Form1.cs* | Form1.cs [Design]* | dashboard.cs [Design]

es-all | es_all.dashboard | dashbo

```
152         //Top oppt
153
154         string topOppotunities = $"SELECT * FROM opportunity WHERE probability > 50";
155         SqlCommand loadtopOppotunities = new SqlCommand(topOppotunities, con);
156         SqlDataAdapter topopportunityAdapter = new SqlDataAdapter(loadtopOppotunities);
157         DataSet topoppertunityDataset = new DataSet();
158         topopportunityAdapter.Fill(topoppertunityDataset);
159
160         dataGridView8.DataSource = topoppertunityDataset.Tables[0];
161
162         //End dashboad
163
164
165
166         SqlCommand loadCustomers = new SqlCommand(customerQuery, con);
167         SqlDataAdapter customerAdapter = new SqlDataAdapter(loadCustomers);
168         DataSet customerDataSet= new DataSet();
169         customerAdapter.Fill(customerDataSet);
170
171         dataGridView1.DataSource= customerDataSet.Tables[0];
172
173         SqlCommand loadCategories = new SqlCommand(catQuery, con);
174         SqlDataAdapter catAdapter = new SqlDataAdapter(loadCategories);
175         DataSet catDataSet= new DataSet();
176         catAdapter.Fill(catDataSet);
177
178         dataGridView2.DataSource= catDataSet.Tables[0];
179
180         SqlCommand loadCompany = new SqlCommand(companyQuery, con);
181         SqlDataAdapter companyAdapter = new SqlDataAdapter(loadCompany);
182         DataSet companyDataSet= new DataSet();
183         companyAdapter.Fill(companyDataSet);
184
185         dataGridView3.DataSource = companyDataSet.Tables[0];
186
187         SqlCommand loadAppointment = new SqlCommand(appoQueary, con);
188         SqlDataAdapter appointmentAdapter = new SqlDataAdapter(loadAppointment);
189         DataSet appointmentDataSet= new DataSet();
190         appointmentAdapter.Fill(appointmentDataSet);
191
192         dataGridView4.DataSource = appointmentDataSet.Tables[0];
```

```
193
194             SqlCommand loadoppertunity = new SqlCommand(opportunityQueary, con);
195             SqlDataAdapter opportunityAdapter = new SqlDataAdapter(loadoppertunity);
196             DataSet oppertunityDataset = new DataSet();
197             opportunityAdapter.Fill(oppertunityDataset);
198
199             dataGridView5.DataSource = oppertunityDataset.Tables[0];
200
201             SqlCommand loadproduct = new SqlCommand(productQueary, con);
202             SqlDataAdapter productAdapter = new SqlDataAdapter(loadproduct);
203             DataSet productDataset = new DataSet();
204             productAdapter.Fill(productDataset);
205
206             dataGridView6.DataSource = productDataset.Tables[0];
207
208         }
209
                1 reference
210         private void button1_Click_1(object sender, EventArgs e)
211         {
212             tabControl1.SelectedIndex = 1;
213             button1.BackColor = Color.OrangeRed;
214             button2.BackColor = Color.White;
215             button3.BackColor = Color.White;
216             button4.BackColor = Color.White;
217             button5.BackColor = Color.White;
218             button6.BackColor = Color.White;
219         }
220
                1 reference
221         private void button2_Click(object sender, EventArgs e)
222         {
223             tabControl1.SelectedIndex = 2;
224             button1.BackColor = Color.White;
225             button2.BackColor = Color.OrangeRed;
226             button3.BackColor = Color.White;
227             button4.BackColor = Color.White;
228             button5.BackColor = Color.White;
229             button6.BackColor = Color.White;
230         }
231
```

```
                1 reference
232         private void button3_Click(object sender, EventArgs e)
233         {
234             tabControl1.SelectedIndex = 3;
235             button1.BackColor = Color.White;
236             button2.BackColor = Color.White;
237             button3.BackColor = Color.OrangeRed;
238             button4.BackColor = Color.White;
239             button5.BackColor = Color.White;
240             button6.BackColor = Color.White;
241         }
242
                1 reference
243         private void button4_Click(object sender, EventArgs e)
244         {
245             tabControl1.SelectedIndex = 4;
246             button1.BackColor = Color.White;
247             button2.BackColor = Color.White;
248             button3.BackColor = Color.White;
249             button4.BackColor = Color.OrangeRed;
250             button5.BackColor = Color.White;
251             button6.BackColor = Color.White;
252         }
253
                1 reference
254         private void button5_Click(object sender, EventArgs e)
255         {
256             tabControl1.SelectedIndex = 5;
257             button1.BackColor = Color.White;
258             button2.BackColor = Color.White;
259             button3.BackColor = Color.White;
260             button4.BackColor = Color.White;
261             button5.BackColor = Color.OrangeRed;
262             button6.BackColor = Color.White;
263         }
264
                1 reference
265         private void button6_Click(object sender, EventArgs e)
266         {
267             tabControl1.SelectedIndex = 6;
268             button1.BackColor = Color.White;
269             button2.BackColor = Color.White;
270             button3.BackColor = Color.White;
271             button4.BackColor = Color.White;
272             button5.BackColor = Color.White;
273             button6.BackColor = Color.OrangeRed;
```

```
1 reference
private void button6_Click(object sender, EventArgs e)
{
    tabControl1.SelectedIndex = 6;
    button1.BackColor = Color.White;
    button2.BackColor = Color.White;
    button3.BackColor = Color.White;
    button4.BackColor = Color.White;
    button5.BackColor = Color.White;
    button6.BackColor = Color.OrangeRed;
}

1 reference
private void tabControl1_SelectedIndexChanged(object sender, EventArgs e)
{
    if(tabControl1.SelectedIndex == 0)
    {
        button1.BackColor= Color.White;
        button2.BackColor = Color.White;
        button3.BackColor = Color.White;
        button4.BackColor = Color.White;
        button5.BackColor = Color.White;
        button6.BackColor = Color.White;
    }

    if(tabControl1.SelectedIndex == 1)
    {
        button1.BackColor = Color.OrangeRed;
        button2.BackColor = Color.White;
        button3.BackColor = Color.White;
        button4.BackColor = Color.White;
        button5.BackColor = Color.White;
        button6.BackColor = Color.White;
    }

    if(tabControl1.SelectedIndex == 2)
    {
        button1.BackColor = Color.White;
        button2.BackColor = Color.OrangeRed;
        button3.BackColor = Color.White;
        button4.BackColor = Color.White;
        button5.BackColor = Color.White;
        button6.BackColor = Color.White;
    }
```

Toolbox

dashboard.cs | Form1.cs* | Form1.cs [Design]* | dashboard.cs [Design]

es-all | es_all.dashboard

```
    if (tabControl1.SelectedIndex == 3)
    {
        button1.BackColor = Color.White;
        button2.BackColor = Color.White;
        button3.BackColor = Color.OrangeRed;
        button4.BackColor = Color.White;
        button5.BackColor = Color.White;
        button6.BackColor = Color.White;
    }

    if(tabControl1.SelectedIndex == 4)
    {
        button1.BackColor = Color.White;
        button2.BackColor = Color.White;
        button3.BackColor = Color.White;
        button4.BackColor = Color.OrangeRed;
        button5.BackColor = Color.White;
        button6.BackColor = Color.White;
    }

    if (tabControl1.SelectedIndex == 5)
    {
        button1.BackColor = Color.White;
        button2.BackColor = Color.White;
        button3.BackColor = Color.White;
        button4.BackColor = Color.White;
        button5.BackColor = Color.OrangeRed;
        button6.BackColor = Color.White;
    }

    if (tabControl1.SelectedIndex == 6)
    {
        button1.BackColor = Color.White;
        button2.BackColor = Color.White;
        button3.BackColor = Color.White;
        button4.BackColor = Color.White;
        button5.BackColor = Color.White;
```

```
1 reference
393     private void button10_Click(object sender, EventArgs e)
394     {
395         SqlCommand loadCustomers = new SqlCommand(customerQuery, con);
396         SqlDataAdapter customerAdapter = new SqlDataAdapter(loadCustomers);
397         DataSet customerDataSet = new DataSet();
398         customerAdapter.Fill(customerDataSet);
399
400         dataGridView1.DataSource = customerDataSet.Tables[0];
401     }
402
    1 reference
403     private void button11_Click(object sender, EventArgs e)
404     {
405         SqlCommand loadCompany = new SqlCommand(companyQuery, con);
406         SqlDataAdapter companyAdapter = new SqlDataAdapter(loadCompany);
407         DataSet companyDataSet = new DataSet();
408         companyAdapter.Fill(companyDataSet);
409
410         dataGridView3.DataSource = companyDataSet.Tables[0];
411     }
412
    1 reference
413     private void button12_Click(object sender, EventArgs e)
414     {
415         SqlCommand loadCategories = new SqlCommand(catQuery, con);
416         SqlDataAdapter catAdapter = new SqlDataAdapter(loadCategories);
417         DataSet catDataSet = new DataSet();
418         catAdapter.Fill(catDataSet);
419
420         dataGridView2.DataSource = catDataSet.Tables[0];
421
422
423     }
```

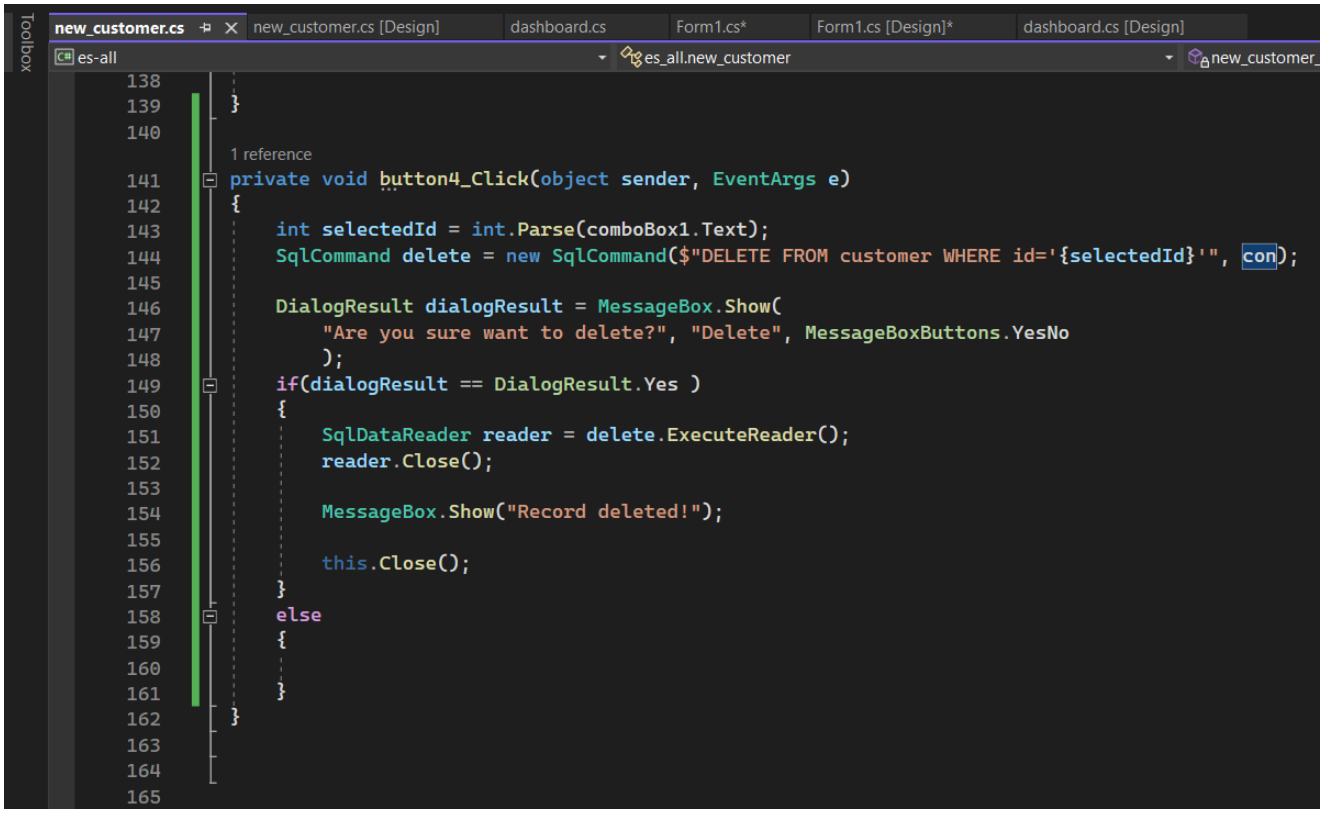
Toolbox | dashboard.cs | Form1.cs* | Form1.cs [Design]* | dashboard.cs [Design]

es-all | es_all.dashboard

```
448     private void button13_Click(object sender, EventArgs e)
449     {
450         SqlCommand loadAppointment = new SqlCommand(appoQueary, con);
451         SqlDataAdapter appointmentAdapter = new SqlDataAdapter(loadAppointment);
452         DataSet appointmentDataSet = new DataSet();
453         appointmentAdapter.Fill(appointmentDataSet);
454
455         dataGridView4.DataSource = appointmentDataSet.Tables[0];
456     }
457
    1 reference
458     private void button15_Click(object sender, EventArgs e)
459     {
460         new_opportunity new_opportunity = new new_opportunity();
461         new_opportunity.Show();
462
463     }
464
    1 reference
465     private void button16_Click(object sender, EventArgs e)
466     {
467         SqlCommand loadoppertunity = new SqlCommand(opportunityQueary, con);
468         SqlDataAdapter opportunityAdapter = new SqlDataAdapter(loadoppertunity);
469         DataSet oppertunityDataset = new DataSet();
470         opportunityAdapter.Fill(oppertunityDataset);
471
472         dataGridView5.DataSource = oppertunityDataset.Tables[0];
473     }
474
    1 reference
475     private void label7_Click(object sender, EventArgs e)
476     {
477
478     }
479
    1 reference
480     private void button17_Click(object sender, EventArgs e)
481     {
482         new_product new_Product = new new_product();
483         new_Product.Show();
484
485     }
```

```
59
                1 reference
60          private void new_customer_Load(object sender, EventArgs e)
61          {
62              con = new SqlConnection(conString);
63              con.Open();
64
65              string loadCompanyQuery = "SELECT companyName FROM company";
66              SqlCommand companyCommand = new SqlCommand(loadCompanyQuery, con);
67              SqlDataReader companyDataReader = companyCommand.ExecuteReader();
68              comboBox2.Items.Clear();
69
70              while (companyDataReader.Read())
71              {
72                  comboBox2.Items.Add(companyDataReader["companyName"].ToString());
73              }
74              companyDataReader.Close();
75
76
77              SqlCommand loadId= new SqlCommand("SELECT id FROM customer", con);
78              SqlDataReader idDataReader = loadId.ExecuteReader();
79
80              while (idDataReader.Read())
81              {
82                  comboBox1.Items.Add(idDataReader["id"]);
83              }
84              idDataReader.Close();
85
86          }
87
```

```
es-all                    es_all.new_customer                    new_customer_Load(object sender, EventArgs e)
93  private void comboBox1_SelectedIndexChanged(object sender, EventArgs e)
94  {
95      //load data when selected id is change in customer
96
97      int selectedId = int.Parse(comboBox1.Text);
98      SqlCommand loadCustomer = new SqlCommand($"SELECT * FROM customer WHERE id = {selectedId}",con);
99      SqlDataReader selectedCustomer = loadCustomer.ExecuteReader();
100
101     while (selectedCustomer.Read())
102     {
103         textBox5.Text = selectedCustomer[1].ToString();
104         textBox1.Text = selectedCustomer[2].ToString();
105         textBox2.Text = selectedCustomer[3].ToString();
106         textBox3.Text = selectedCustomer[4].ToString();
107         comboBox2.Text = selectedCustomer[5].ToString();
108     }
109     selectedCustomer.Close();
110 }
111
    1 reference
112 private void button1_Click_1(object sender, EventArgs e)
113 {
114     string fName = textBox5.Text;
115     string lName = textBox1.Text;
116     string email = textBox2.Text;
117     string mobile = textBox3.Text;
118     string company = comboBox2.Text;
119
120     SqlCommand saveCompany = new SqlCommand($"INSERT INTO customer(firstName,lastName,email,mobile,company) VALUES('{fName}','{lName}','{email}','{mobile}','{company}')", con);
121     saveCompany.ExecuteNonQuery();
122     textBox5.Clear();
123     textBox1.Clear();
124     textBox2.Clear();
125     textBox3.Clear();
126
127
128     MessageBox.Show("New customer added");
129 }
```



```
}

1 reference
private void button4_Click(object sender, EventArgs e)
{
    int selectedId = int.Parse(comboBox1.Text);
    SqlCommand delete = new SqlCommand($"DELETE FROM customer WHERE id='{selectedId}'", con);

    DialogResult dialogResult = MessageBox.Show(
        "Are you sure want to delete?", "Delete", MessageBoxButtons.YesNo
        );
    if(dialogResult == DialogResult.Yes )
    {
        SqlDataReader reader = delete.ExecuteReader();
        reader.Close();

        MessageBox.Show("Record deleted!");

        this.Close();
    }
    else
    {

    }
}
```

09. Database & tables

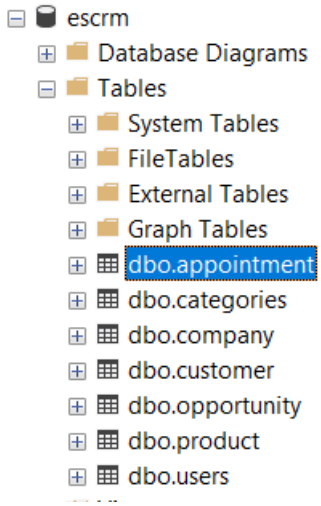


Figure 1 - Database

Column Name	Data Type	Allow Nulls
id	int	☐
username	varchar(50)	☐
password	varchar(50)	☐
		☐

Figure 2- User table

Column Name	Data Type	Allow Nulls
id	int	☐
firstName	varchar(50)	☑
lastName	varchar(50)	☑
email	varchar(50)	☑
mobile	int	☑
company	varchar(50)	☑
		☐

Figure 3 - Customer table

	Column Name	Data Type	Allow Nulls
▶	id	int	☐
	companyName	varchar(50)	☑
	address	varchar(50)	☑
	email	varchar(50)	☑
	telephone	int	☑
	category	varchar(50)	☑
			☐

Figure 4 - Company table

	Column Name	Data Type	Allow Nulls
▶	id	int	☐
	referanceNumber	int	☐
	venue	varchar(50)	☑
	dateTime	datetime	☑
	person	varchar(50)	☑
	topic	varchar(50)	☑
			☐

Figure 5 - Appointment table

	Column Name	Data Type	Allow Nulls
▶	id	int	☐
	referanceNumber	int	☐
	company	varchar(50)	☐
	person	varchar(50)	☐
	probability	int	☐
	description	varchar(50)	☐
			☐

Figure 6 - Opportunity table

	Column Name	Data Type	Allow Nulls
▶	id	int	☐
	productCode	int	☐
	productName	varchar(50)	☐
	supplier	varchar(50)	☐
	category	varchar(50)	☐
			☐

Figure 7 - Product table

	Column Name	Data Type	Allow Nulls
▶	id	int	☐
	category	varchar(50)	☑
			☐

Figure 8 - Category table

10. Group members

22744	BT Jayasuriya
23583	MKVI Medhani
22724	KWCNB Kobbekaduwa
22717	EGTD Egodage
22748	WBMNWN Basnayake
22749	WK Chandanayake
23648	NS Liyanage

End