

CRM SYSTEM

Project Report



IS204.3 Enterprise System
Lecturer: Mrs. Chalani Oruthotaarachchi
Group M

TABLE OF CONTENTS

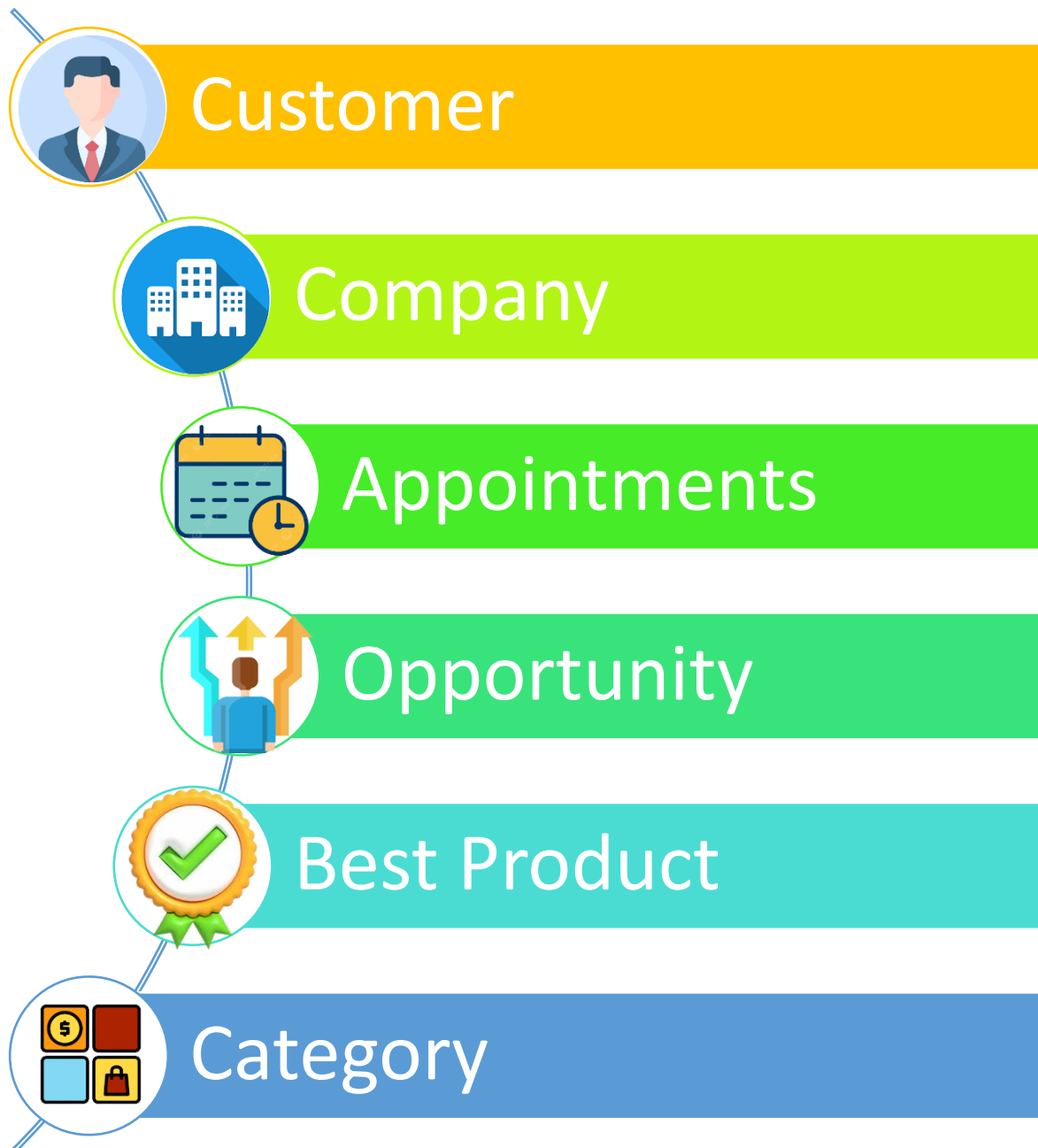
I System Introduction	3
II Main type of function in the system	4
III Identification of the system functions	5
IV How we build the system	6
V Flow of the system	7
VI Diagrams	8
VII User Interfaces design	12
VIII System backend	19
IX Database tablel	27
X Group members	30

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.

O2. MAIN TYPE OF FUNCTION IN THE SYSTEM



03. IDENTIFICATION OF THE SYSTEM FUNCTIONS

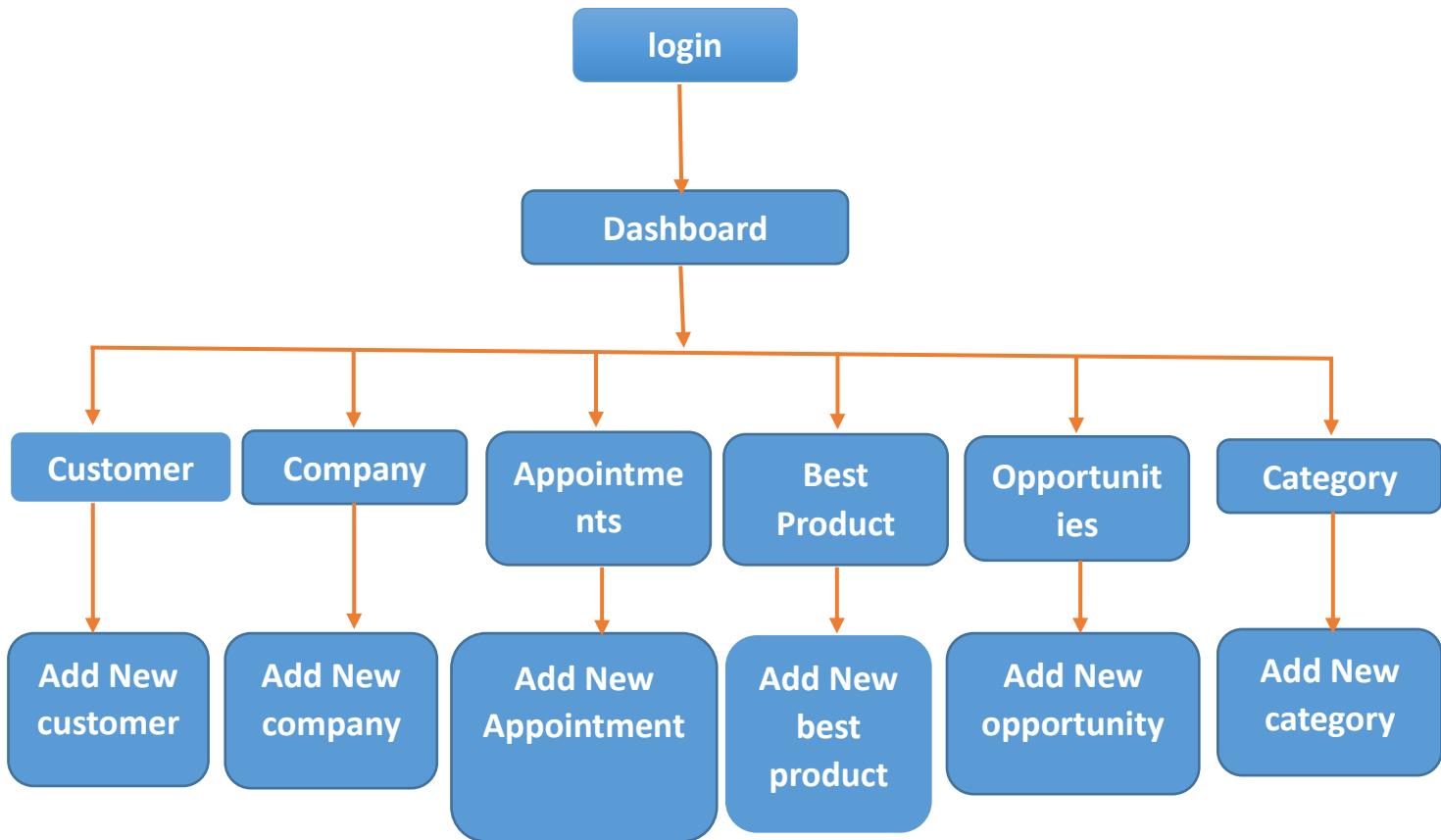
- ❖ **User** — The person who used the CRM system. The user has done every task in this system. He enters every data to the system.
- ❖ **customer**— The owners of the companies. This system can be used by many companies, so that many customers have in this system. In this system the customer can get 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** — That's can be used for any types of fields, such as transportation, health field, agricultural field etc.

O4. 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 coding 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.

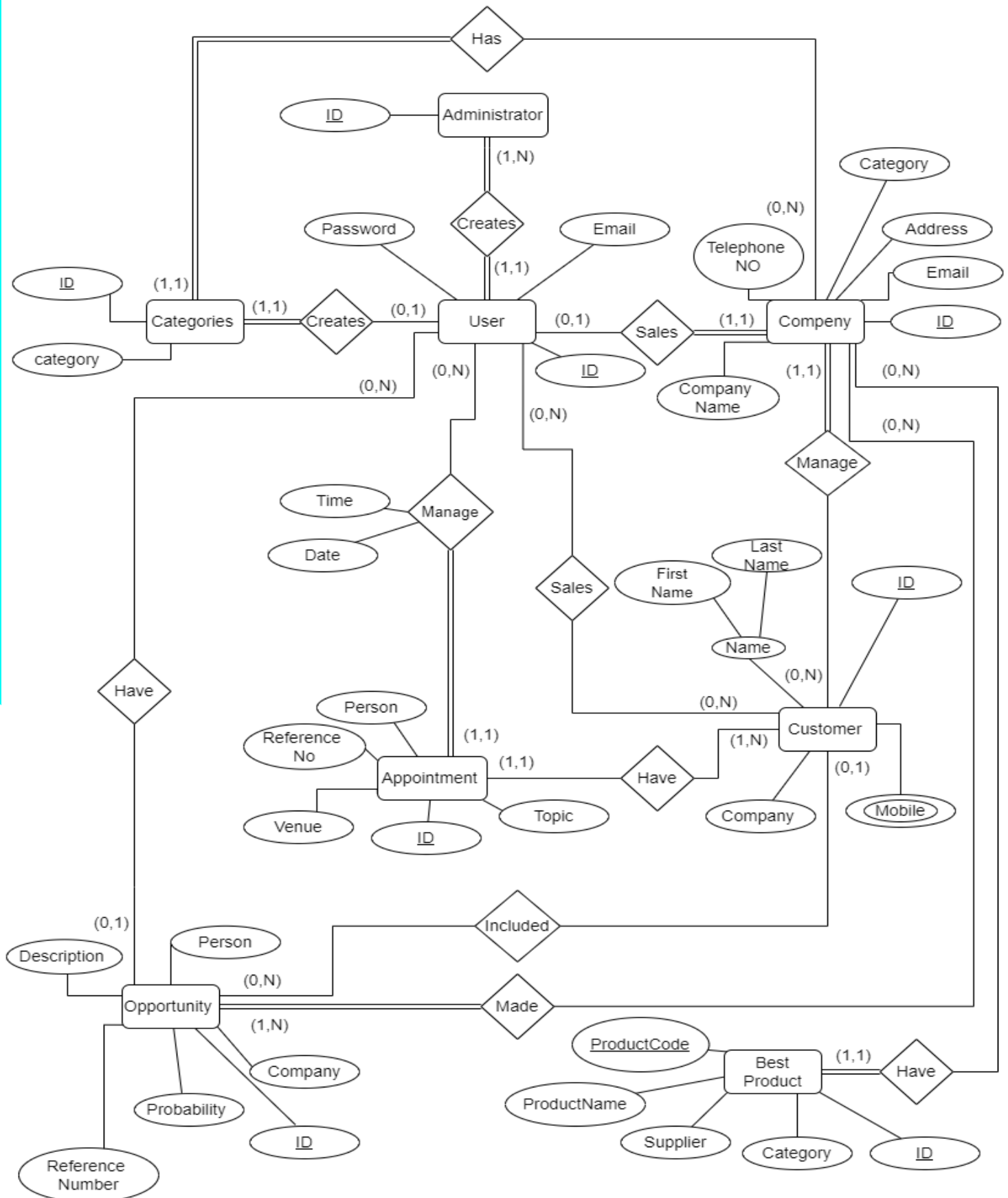


05. FLOW OF THE SYSTEM



06. DIAGRAMS

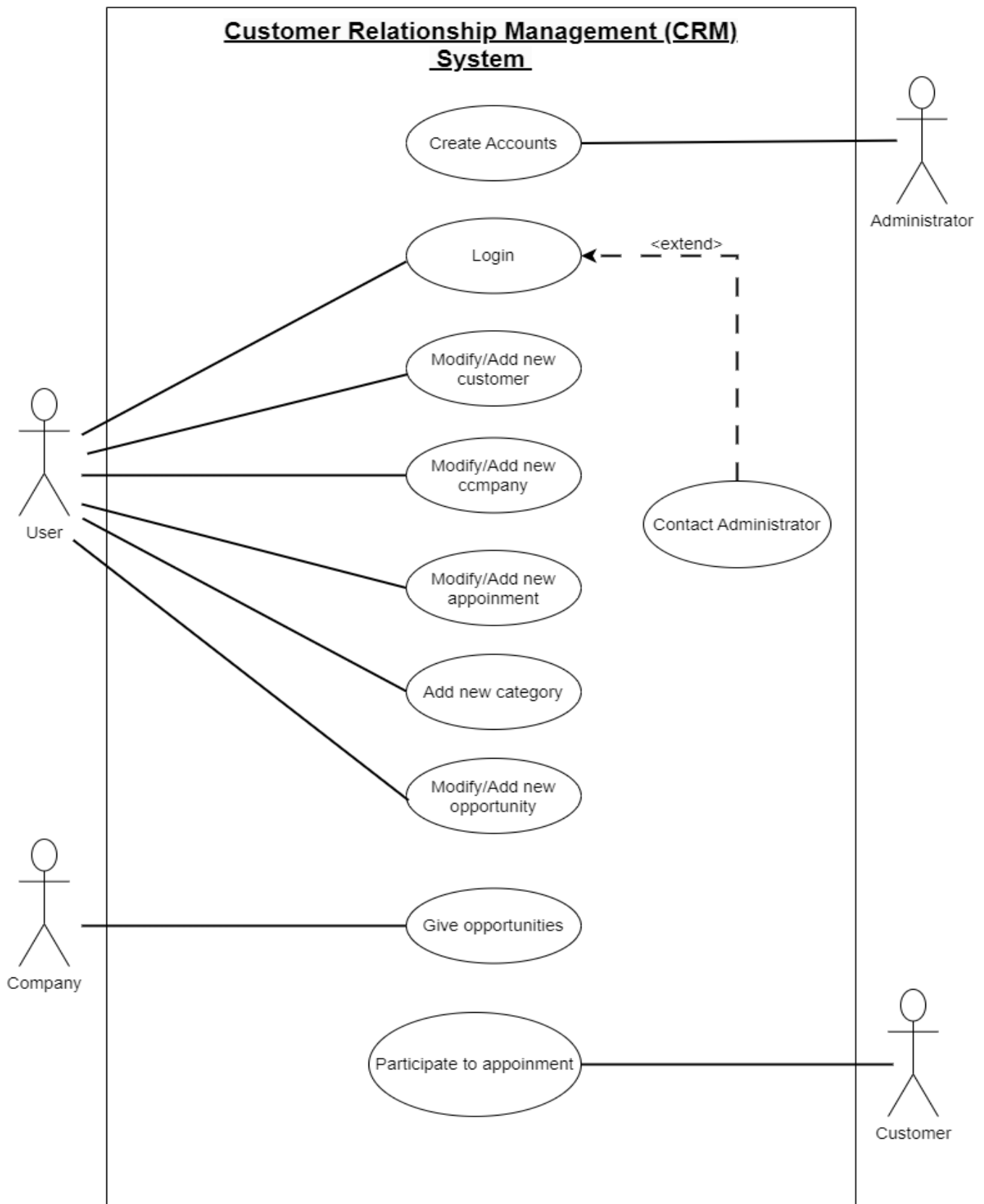
I. ER-DIAGRAM



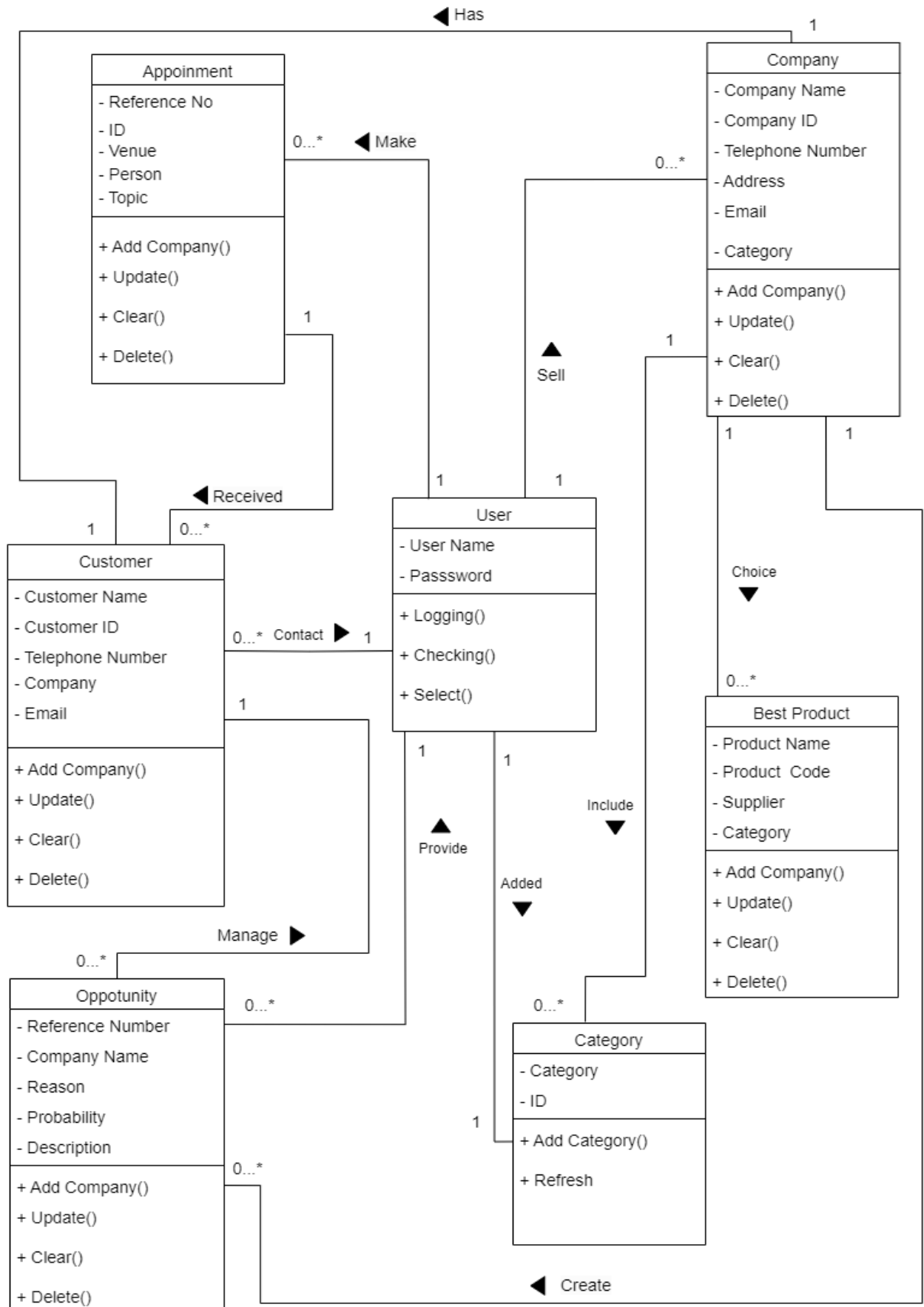
ASSUMPTIONS

- 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

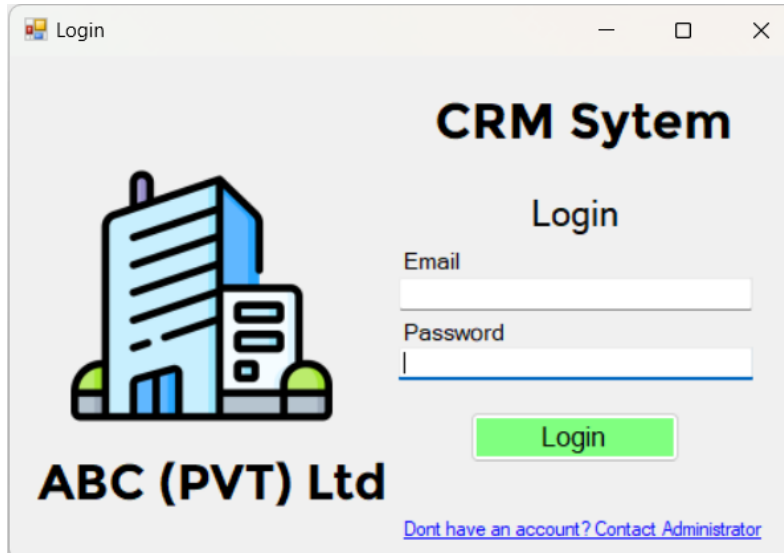


III. CLASS DIAGRAM



07. USER INTERFACES DESIGNS

Login page



The login page for the CRM System is displayed in a window titled 'Login'. It features a logo for 'ABC (PVT) Ltd' on the left, which includes a stylized building icon. The main heading is 'CRM System' followed by 'Login'. There are input fields for 'Email' and 'Password', and a green 'Login' button. A link at the bottom says 'Dont have an account? Contact Administrator'.

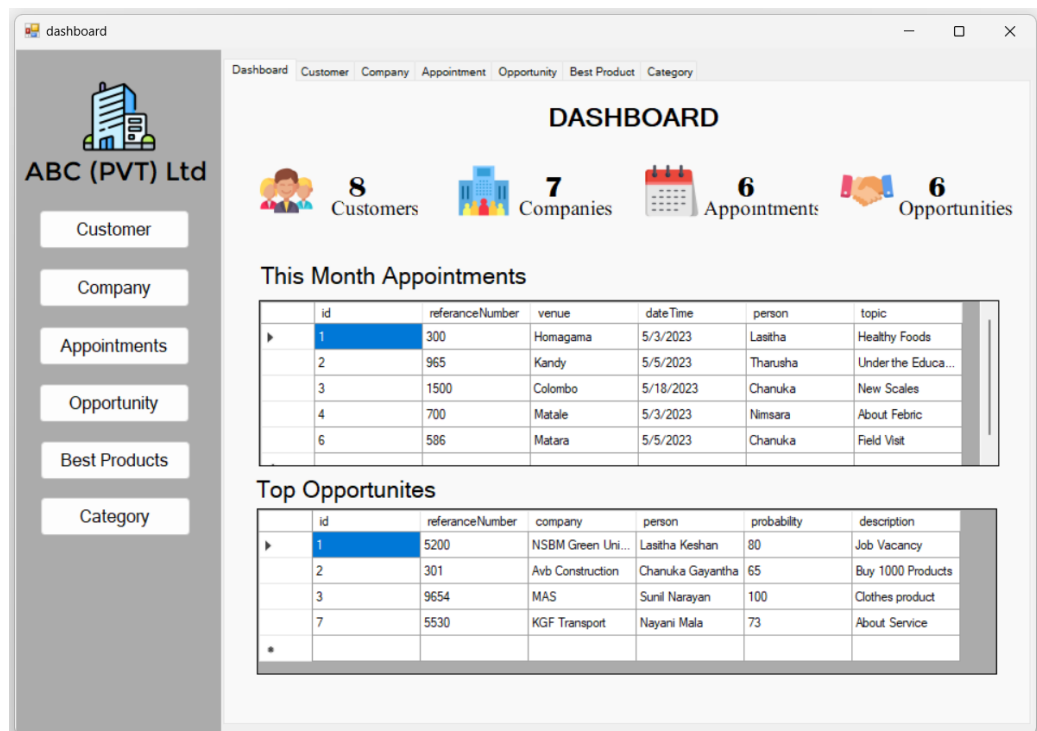
In this page user can login to the system using their password and mail. First of all, users want to create an account for this system by contacting the Administrator.

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.

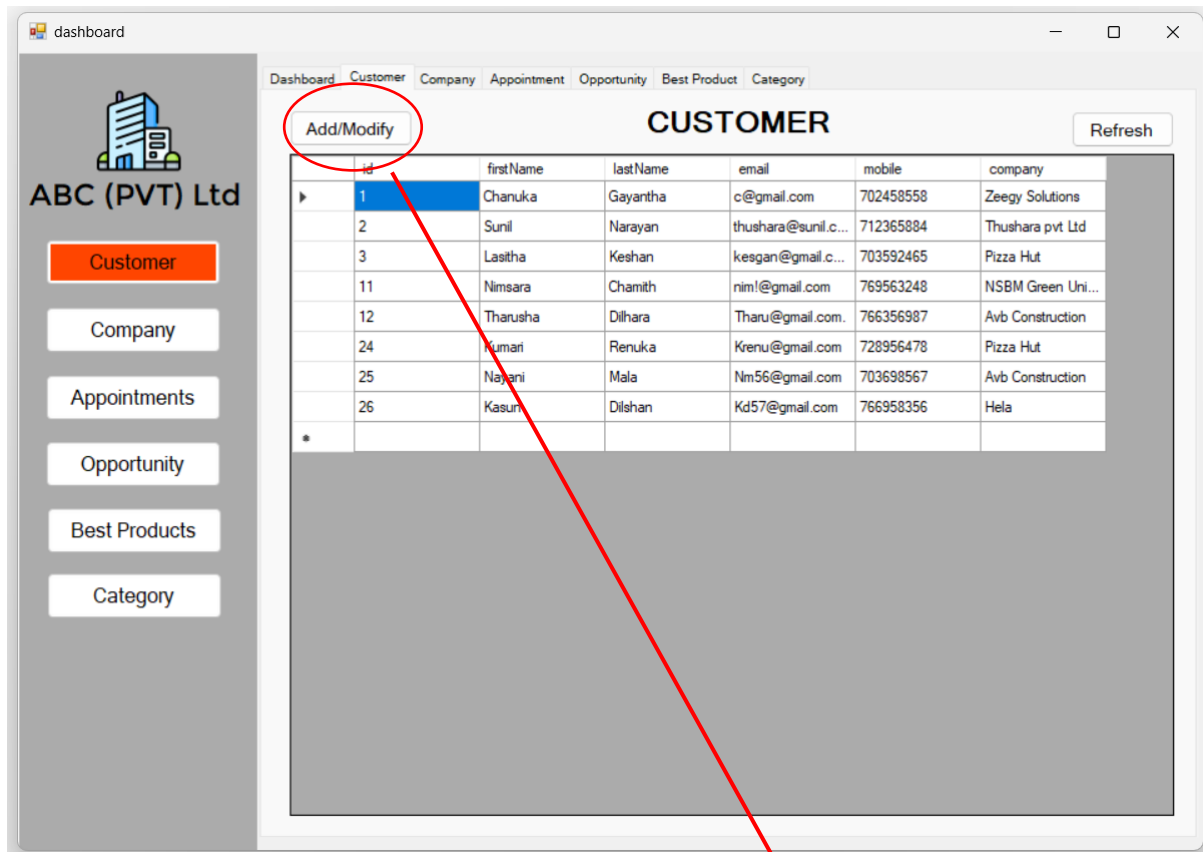


The dashboard is titled 'DASHBOARD' and shows various metrics and data tables. On the left, there is a sidebar for 'ABC (PVT) Ltd' with buttons for 'Customer', 'Company', 'Appointments', 'Opportunity', 'Best Products', and 'Category'. The main content area displays four summary cards: '8 Customers', '7 Companies', '6 Appointments', and '6 Opportunities'. Below these are two tables: 'This Month Appointments' and 'Top Opportunites'.

	id	referenceNumber	venue	dateTime	person	topic
▶	1	300	Homagama	5/3/2023	Lasitha	Healthy Foods
	2	965	Kandy	5/5/2023	Tharusha	Under the Educa...
	3	1500	Colombo	5/18/2023	Chanuka	New Scales
	4	700	Matale	5/3/2023	Nimsara	About Febtic
	6	586	Matara	5/5/2023	Chanuka	Field Visit

	id	referenceNumber	company	person	probability	description
▶	1	5200	NSBM Green Uni...	Lasitha Keshan	80	Job Vacancy
	2	301	Avb Construction	Chanuka Gayantha	65	Buy 1000 Products
	3	9654	MAS	Sunil Narayan	100	Clothes product
	7	5530	KGF Transport	Nayani Mala	73	About Service
*						

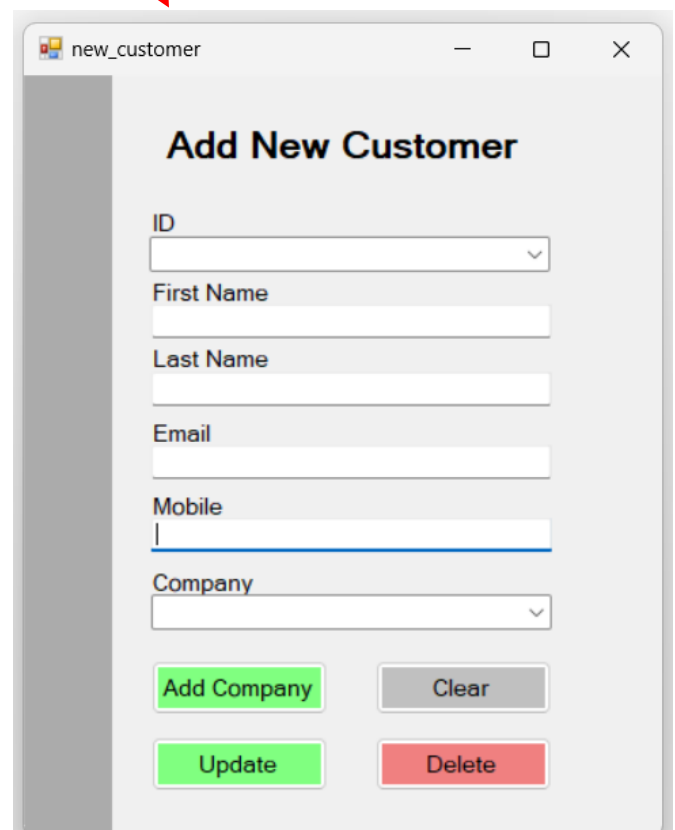
Customer page



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.

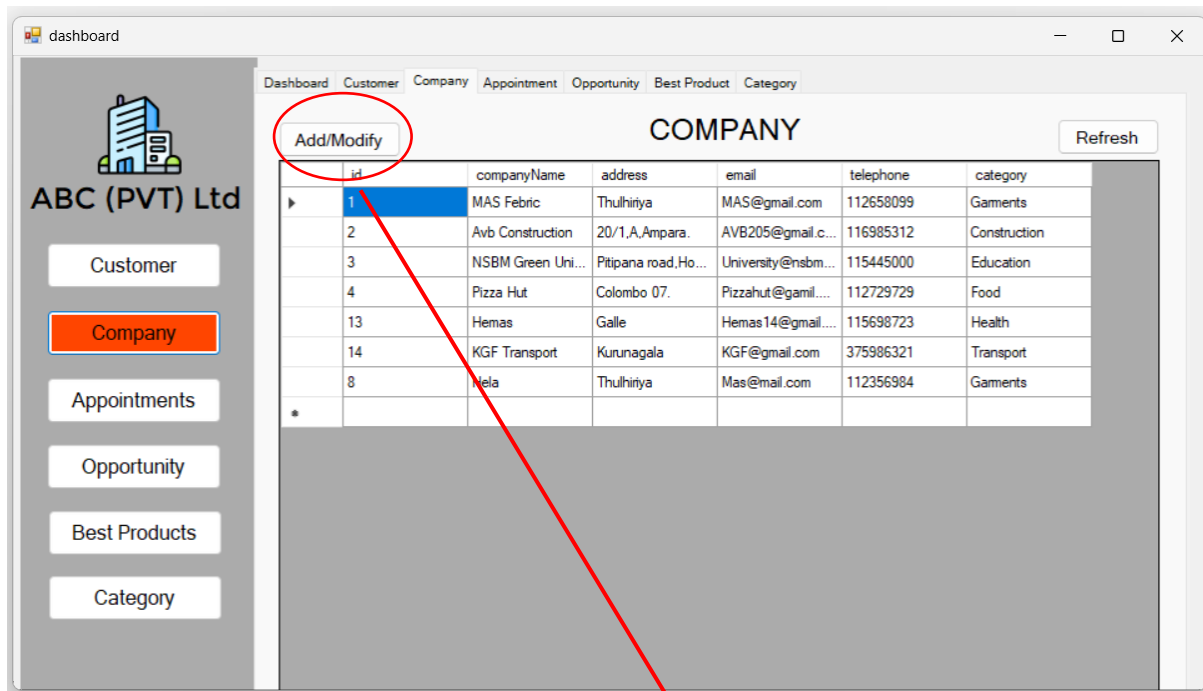
02



The screenshot shows a 'new_customer' window titled 'Add New Customer'. It contains the following fields and buttons:

- ID: A dropdown menu.
- First Name: A text input field.
- Last Name: A text input field.
- Email: A text input field.
- Mobile: A text input field.
- Company: A dropdown menu.
- Buttons: 'Add Company' (green), 'Clear' (grey), 'Update' (green), and 'Delete' (red).

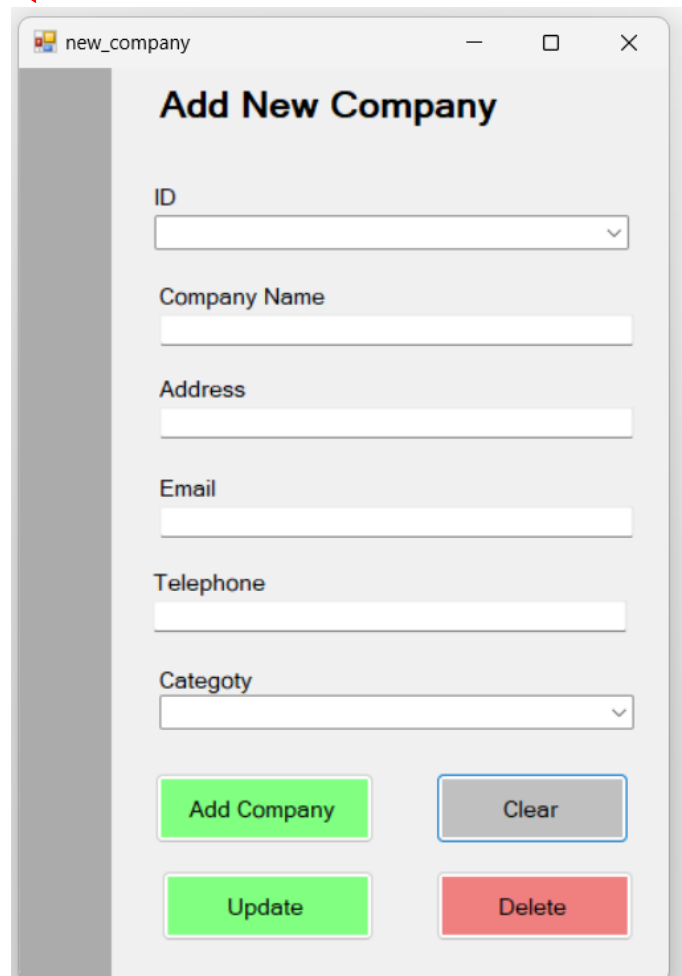
Company page



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



new_company

Add New Company

ID

Company Name

Address

Email

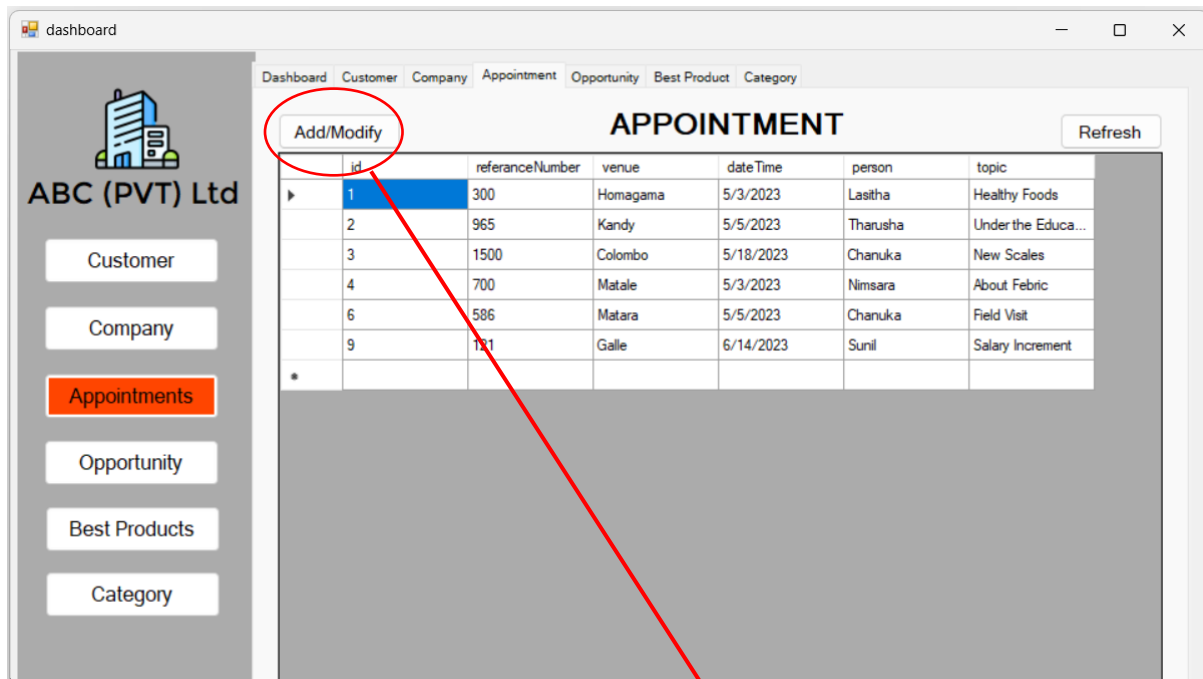
Telephone

Category

Add Company Clear

Update Delete

Appointment page

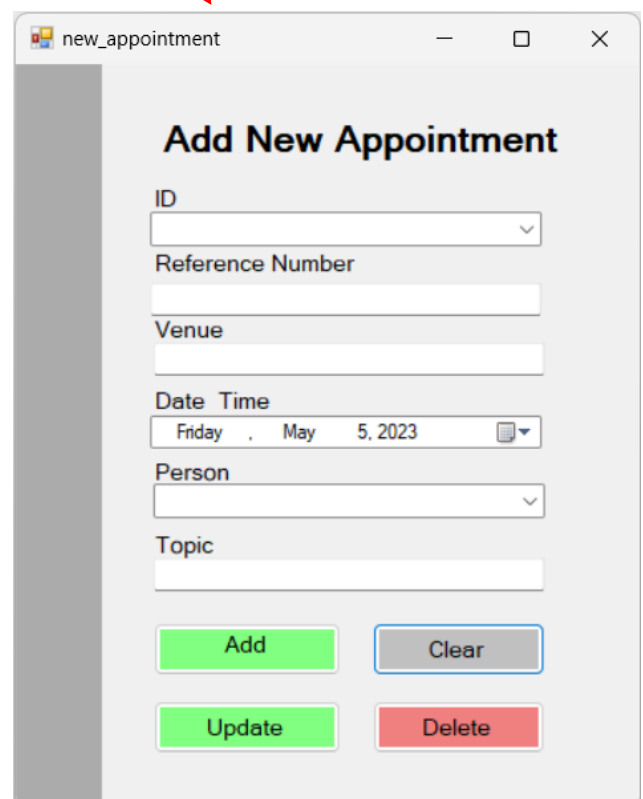


01

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.

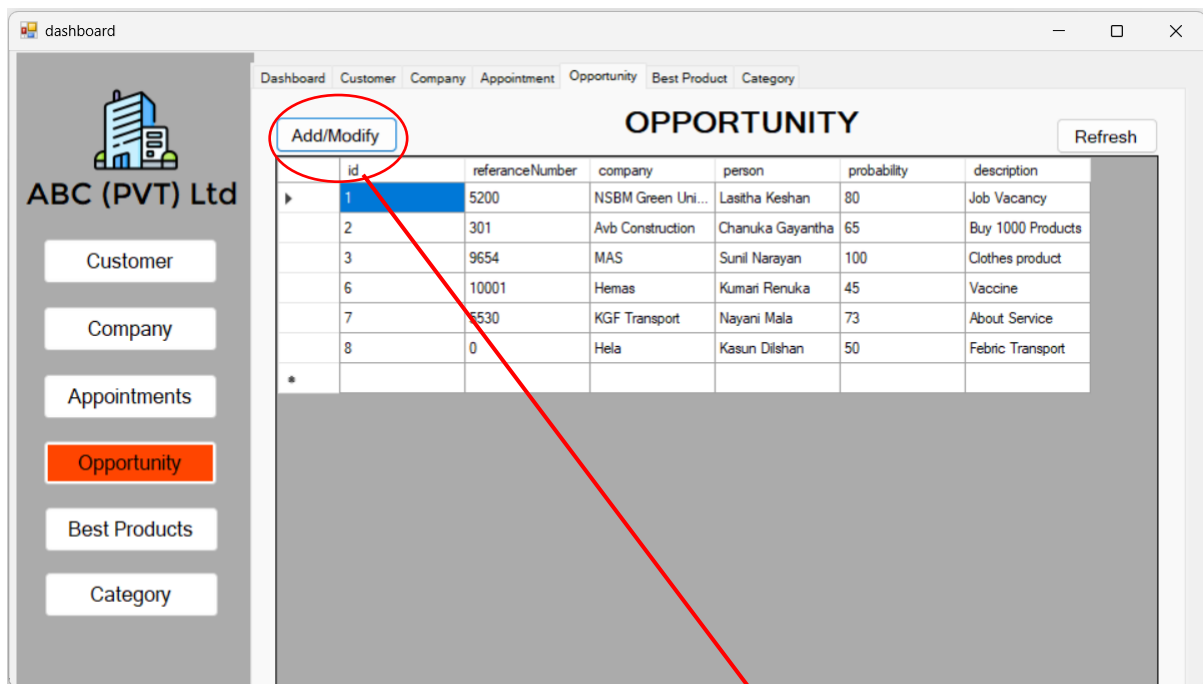
02



The 'new_appointment' form is titled 'Add New Appointment'. It contains the following fields and controls:

- ID**: A dropdown menu.
- Reference Number**: A text input field.
- Venue**: A text input field.
- Date Time**: A date and time picker showing 'Friday, May 5, 2023'.
- Person**: A dropdown menu.
- Topic**: A text input field.
- Buttons**: Four buttons at the bottom: 'Add' (green), 'Clear' (grey), 'Update' (green), and 'Delete' (red).

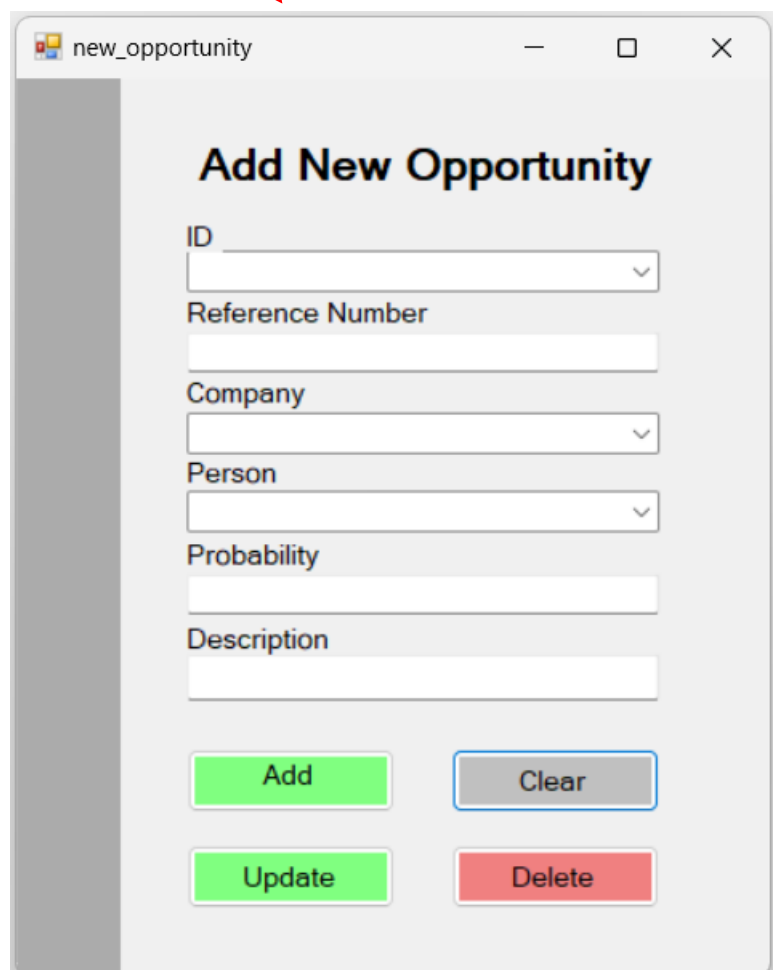
Opportunity page



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.

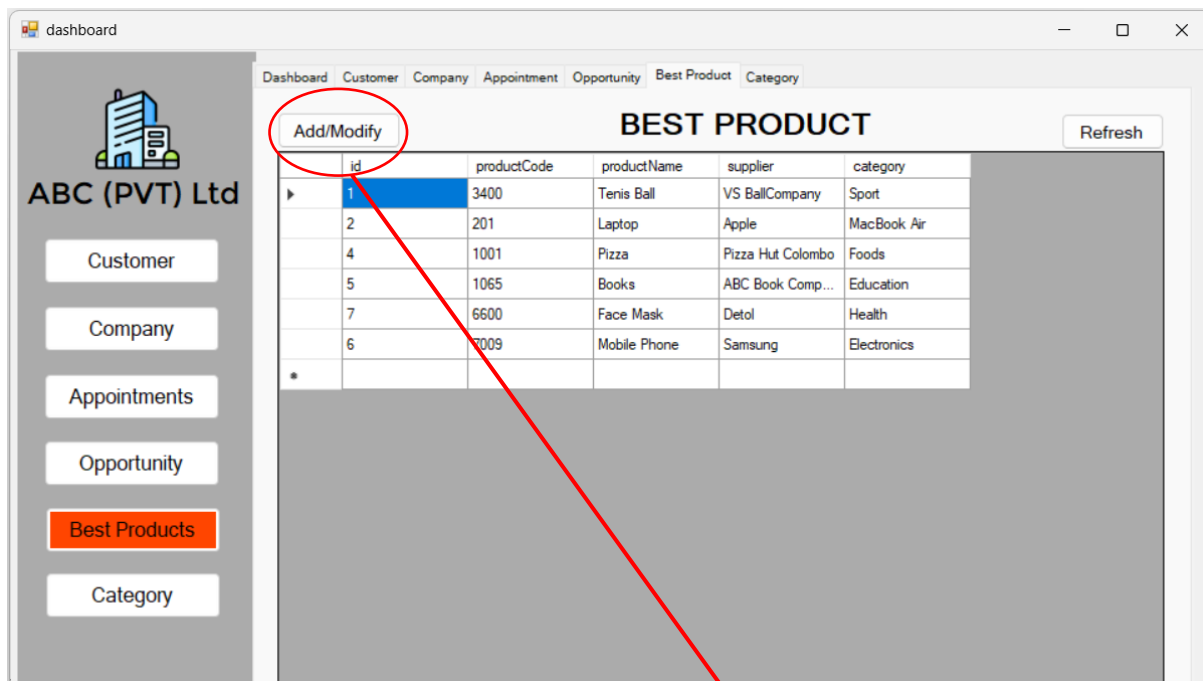
Top opportunity is shown on the dashboard.



The screenshot shows a 'new_opportunity' window with the title 'Add New Opportunity'. It contains several input fields: ID (a dropdown menu), Reference Number, Company, Person, Probability, and Description. At the bottom, there are four buttons: Add (green), Clear (grey), Update (green), and Delete (red).

02

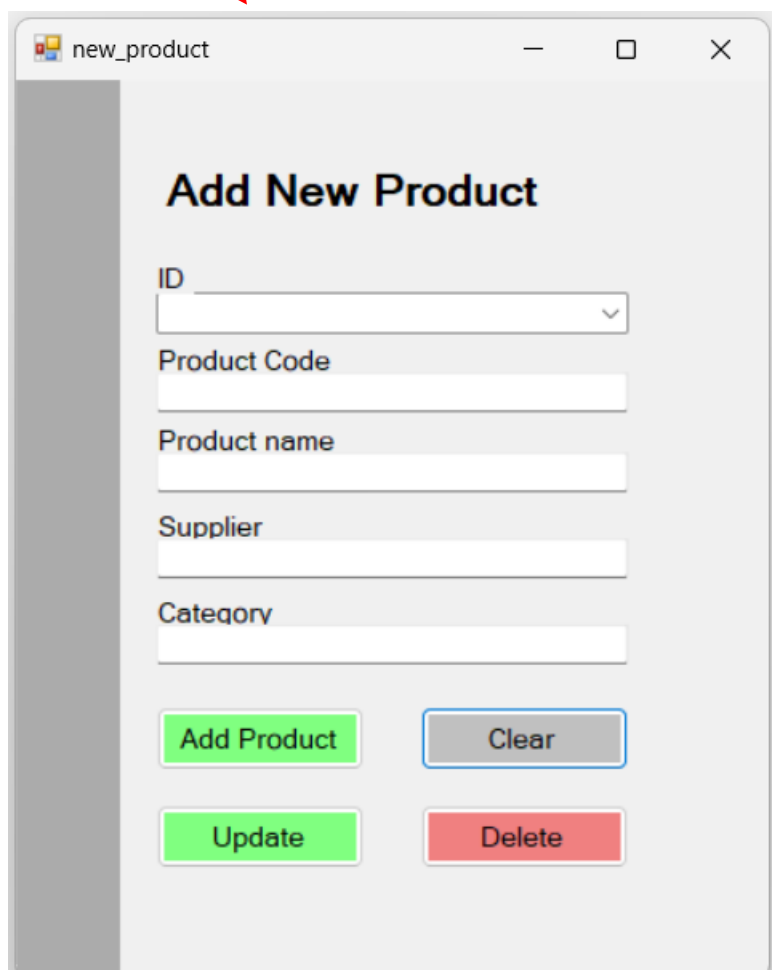
Best product



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.

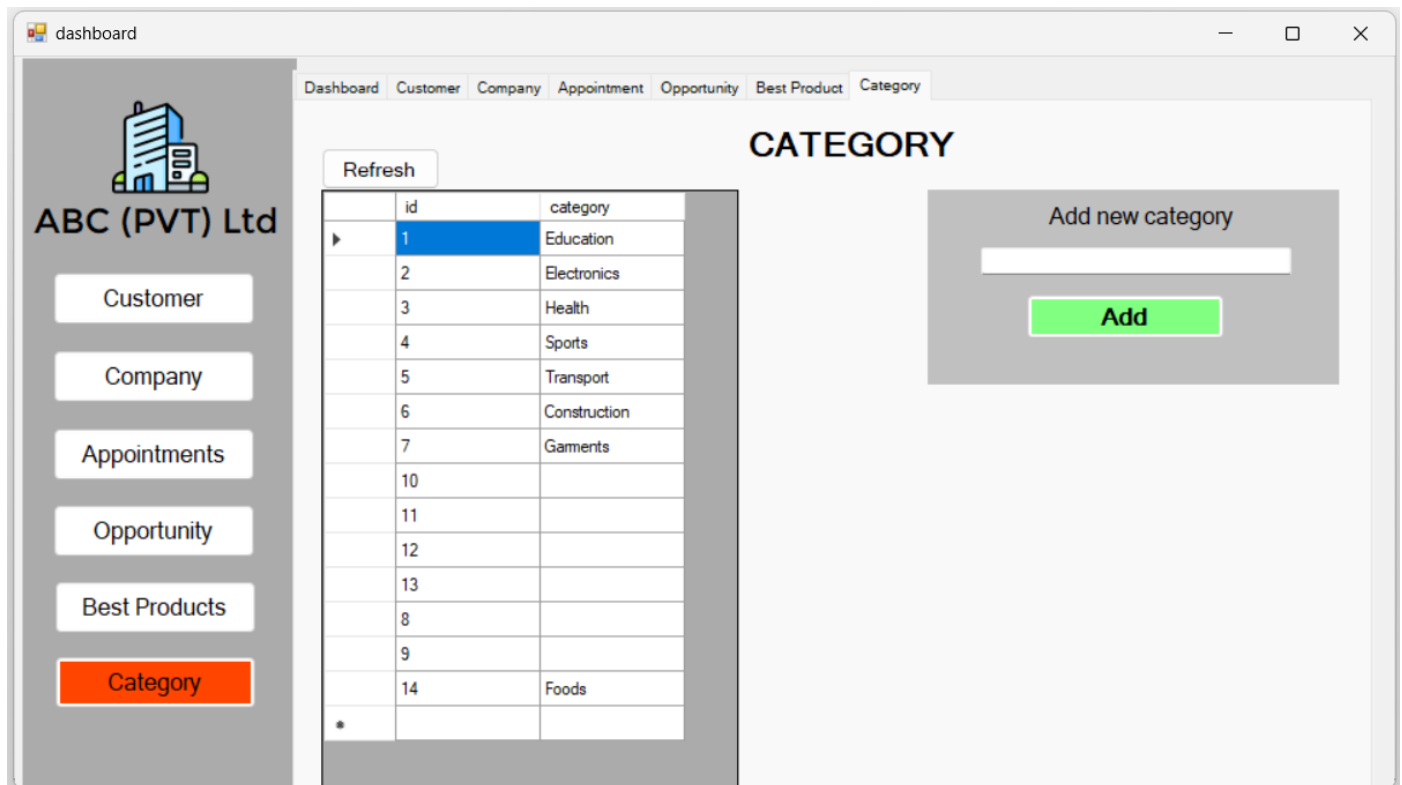
02



The 'Add New Product' form contains the following fields and buttons:

- ID: A dropdown menu.
- Product Code: A text input field.
- Product name: A text input field.
- Supplier: A text input field.
- Category: A text input field.
- Buttons: 'Add Product' (green), 'Clear' (grey), 'Update' (green), and 'Delete' (red).

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

The screenshot displays the Visual Studio IDE with the following components:

- Editor Window:** Shows the code for `Form1.cs` in Design mode. The code includes a `textBox1_TextChanged` event handler and a `button1_Click` event handler. The `button1_Click` method performs a database query to check user credentials and displays a message box. The status bar at the bottom indicates "83 %", "No issues found", and line 63, column 28.
- Solution Explorer:** Located on the right, it shows the project structure. The project is named "es-all" and contains several files, including `Form1.cs`, `new_appointment.cs`, `new_company.cs`, `new_customer.cs`, `new_opportunity.cs`, `new_product.cs`, `Program.cs`, and `register.cs`.
- Properties Window:** Also on the right, it shows the properties of the selected file, `Form1.cs`.
- Output Window:** At the bottom, it shows the output of the application, including the error list and code lens.

The screenshot displays the Visual Studio IDE with the following components:

- Menu Bar:** File, Edit, View, Git, Project, Build, Debug, Test, Analyze, Tools, Extensions, Window, Help.
- Search Bar:** Search (Ctrl+F) with a magnifying glass icon.
- Toolbar:** Includes icons for undo, redo, save, and other standard development actions.
- Code Editor:**
 - File: **Form1.cs***
 - View: **Form1.cs [Design]***
 - Content: **dashboard.cs [Design]**
 - Current Selection: **es_allForm1**
 - Current Line: **button1_Click(object sender, EventArgs e)**
- Code Snippet:**

```

28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
}

1 reference
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!");
}
}

```
- Solution Explorer:**
 - Search: **Search Solution Explorer (Ctrl+;)**
 - Project: **es-all**
 - Files: Properties, References, App.config, dashboard.cs, **Form1.cs**, new_appointment.cs, new_company.cs, new_customer.cs, new_opportunity.cs, new_product.cs, Program.cs, register.cs.
- Properties Window:**
 - Tab: **Properties**
 - Content: Empty.
- Status Bar:**
 - 75% zoom
 - No issues found
 - Ln: 71, Ch: 1, SPC, CRLF

dashboard.cs

Form1.cs*

Form1.cs [Design]*

dashboard.cs [Design]

C# es-all

es_all.dashboard

75

1 reference

76 private void dashboard_Load(object sender, EventArgs e)

77 {

78 con = new SqlConnection(conString);

79 con.Open();

80

81 //dashboard

82

83 SqlCommand customerCountQ = new SqlCommand("SELECT * FROM customer", con);

84 SqlDataReader customerRead = customerCountQ.ExecuteReader();

85

86 int customerCount = 0;

87

88 while (customerRead.Read())

89 {

90 customerCount++;

91 }

92 label13.Text = "";

93 label13.Text = customerCount.ToString();

94

95 customerRead.Close();

96

97 SqlCommand companyCountQ = new SqlCommand("SELECT * FROM company", con);

98 SqlDataReader companyRead = companyCountQ.ExecuteReader();

99

100 int companyCount = 0;

101

102 while (companyRead.Read())

103 {

104 companyCount++;

105 }

106 label14.Text = "";

107 label14.Text = companyCount.ToString();

108

109 companyRead.Close();

110

111 SqlCommand appointmentCountQ = new SqlCommand("SELECT * FROM appointment", con);

112 SqlDataReader appointmentRead = appointmentCountQ.ExecuteReader();

113

```

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  dashboard
152     //Top oppt
153
154     string topOppotunities = $"SELECT * FROM opportunity WHERE probability > 50";
155     SqlCommand loadtopOppotunities = new SqlCommand(topOppotunities, con);
156     SqlDataAdapter topoppportunityAdapter = new SqlDataAdapter(loadtopOppotunities);
157     DataSet topoppportunityDataset = new DataSet();
158     topoppportunityAdapter.Fill(topoppportunityDataset);
159
160     dataGridView8.DataSource = topoppportunityDataset.Tables[0];
161
162     //End dashboard
163
164
165     SqlCommand loadCustomers = new SqlCommand(customerQuery, con);
166     SqlDataAdapter customerAdapter = new SqlDataAdapter(loadCustomers);
167     DataSet customerDataSet= new DataSet();
168     customerAdapter.Fill(customerDataSet);
169
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 loadopportunity = new SqlCommand(opportunityQuery, con);
195 SqlDataAdapter opportunityAdapter = new SqlDataAdapter(loadopportunity);
196 DataSet opportunityDataset = new DataSet();
197 opportunityAdapter.Fill(opportunityDataset);
198
199 dataGridView5.DataSource = opportunityDataset.Tables[0];
200
201 SqlCommand loadproduct = new SqlCommand(productQuery, 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
210 1 reference
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
221 1 reference
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

```

```

232 1 reference
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
243 1 reference
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
254 1 reference
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
265 1 reference
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;

```

```

265     1 reference
266     private void button6_Click(object sender, EventArgs e)
267     {
268         tabControl1.SelectedIndex = 6;
269         button1.BackColor = Color.White;
270         button2.BackColor = Color.White;
271         button3.BackColor = Color.White;
272         button4.BackColor = Color.White;
273         button5.BackColor = Color.White;
274         button6.BackColor = Color.OrangeRed;
275     }
276
277     1 reference
278     private void tabControl1_SelectedIndexChanged(object sender, EventArgs e)
279     {
280         if(tabControl1.SelectedIndex == 0)
281         {
282             button1.BackColor= Color.White;
283             button2.BackColor = Color.White;
284             button3.BackColor = Color.White;
285             button4.BackColor = Color.White;
286             button5.BackColor = Color.White;
287             button6.BackColor = Color.White;
288         }
289
290         if(tabControl1.SelectedIndex == 1)
291         {
292             button1.BackColor = Color.OrangeRed;
293             button2.BackColor = Color.White;
294             button3.BackColor = Color.White;
295             button4.BackColor = Color.White;
296             button5.BackColor = Color.White;
297             button6.BackColor = Color.White;
298         }
299
300         if(tabControl1.SelectedIndex == 2)
301         {
302             button1.BackColor = Color.White;
303             button2.BackColor = Color.OrangeRed;
304             button3.BackColor = Color.White;
305             button4.BackColor = Color.White;
306             button5.BackColor = Color.White;
307             button6.BackColor = Color.White;
308         }
309
310         if (tabControl1.SelectedIndex == 3)
311         {
312             button1.BackColor = Color.White;
313             button2.BackColor = Color.White;
314             button3.BackColor = Color.OrangeRed;
315             button4.BackColor = Color.White;
316             button5.BackColor = Color.White;
317             button6.BackColor = Color.White;
318         }
319
320         if(tabControl1.SelectedIndex == 4)
321         {
322             button1.BackColor = Color.White;
323             button2.BackColor = Color.White;
324             button3.BackColor = Color.White;
325             button4.BackColor = Color.OrangeRed;
326             button5.BackColor = Color.White;
327             button6.BackColor = Color.White;
328         }
329
330         if (tabControl1.SelectedIndex == 5)
331         {
332             button1.BackColor = Color.White;
333             button2.BackColor = Color.White;
334             button3.BackColor = Color.White;
335             button4.BackColor = Color.White;
336             button5.BackColor = Color.OrangeRed;
337             button6.BackColor = Color.White;
338         }
339
340         if (tabControl1.SelectedIndex == 6)
341         {
342             button1.BackColor = Color.White;
343             button2.BackColor = Color.White;
344             button3.BackColor = Color.White;
345             button4.BackColor = Color.White;
346             button5.BackColor = Color.White;
347             button6.BackColor = Color.OrangeRed;
348         }
349     }

```

Toolbox

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

C# es-all es_all.dashboard

```

308     if (tabControl1.SelectedIndex == 3)
309     {
310         button1.BackColor = Color.White;
311         button2.BackColor = Color.White;
312         button3.BackColor = Color.OrangeRed;
313         button4.BackColor = Color.White;
314         button5.BackColor = Color.White;
315         button6.BackColor = Color.White;
316     }
317
318     if(tabControl1.SelectedIndex == 4)
319     {
320         button1.BackColor = Color.White;
321         button2.BackColor = Color.White;
322         button3.BackColor = Color.White;
323         button4.BackColor = Color.OrangeRed;
324         button5.BackColor = Color.White;
325         button6.BackColor = Color.White;
326     }
327
328     if (tabControl1.SelectedIndex == 5)
329     {
330         button1.BackColor = Color.White;
331         button2.BackColor = Color.White;
332         button3.BackColor = Color.White;
333         button4.BackColor = Color.White;
334         button5.BackColor = Color.OrangeRed;
335         button6.BackColor = Color.White;
336     }
337
338     if (tabControl1.SelectedIndex == 6)
339     {
340         button1.BackColor = Color.White;
341         button2.BackColor = Color.White;
342         button3.BackColor = Color.White;
343         button4.BackColor = Color.White;
344         button5.BackColor = Color.White;
345         button6.BackColor = Color.OrangeRed;
346     }
347
348     }
349

```



```

393     1 reference
394     private void button10_Click(object sender, EventArgs e)
395     {
396         SqlCommand loadCustomers = new SqlCommand(customerQuery, con);
397         SqlDataAdapter customerAdapter = new SqlDataAdapter(loadCustomers);
398         DataSet customerDataSet = new DataSet();
399         customerAdapter.Fill(customerDataSet);
400
401         dataGridView1.DataSource = customerDataSet.Tables[0];
402     }
403
404     1 reference
405     private void button11_Click(object sender, EventArgs e)
406     {
407         SqlCommand loadCompany = new SqlCommand(companyQuery, con);
408         SqlDataAdapter companyAdapter = new SqlDataAdapter(loadCompany);
409         DataSet companyDataSet = new DataSet();
410         companyAdapter.Fill(companyDataSet);
411
412         dataGridView3.DataSource = companyDataSet.Tables[0];
413     }
414
415     1 reference
416     private void button12_Click(object sender, EventArgs e)
417     {
418         SqlCommand loadCategories = new SqlCommand(catQuery, con);
419         SqlDataAdapter catAdapter = new SqlDataAdapter(loadCategories);
420         DataSet catDataSet = new DataSet();
421         catAdapter.Fill(catDataSet);
422
423         dataGridView2.DataSource = catDataSet.Tables[0];
424     }

```

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(appoQuery, 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
458     1 reference
459     private void button15_Click(object sender, EventArgs e)
460     {
461         new_opportunity new_opportunity = new new_opportunity();
462         new_opportunity.Show();
463     }
464
465     1 reference
466     private void button16_Click(object sender, EventArgs e)
467     {
468         SqlCommand loadopportunity = new SqlCommand(opportunityQuery, con);
469         SqlDataAdapter opportunityAdapter = new SqlDataAdapter(loadopportunity);
470         DataSet opportunityDataset = new DataSet();
471         opportunityAdapter.Fill(opportunityDataset);
472
473         dataGridView5.DataSource = opportunityDataset.Tables[0];
474     }
475
476     1 reference
477     private void label7_Click(object sender, EventArgs e)
478     {
479     }
480
481     1 reference
482     private void button17_Click(object sender, EventArgs e)
483     {
484         new_product new_Product = new new_product();
485         new_Product.Show();
486     }

```



```

59
60 1 reference
61 private void new_customer_Load(object sender, EventArgs e)
62 {
63     con = new SqlConnection(conString);
64     con.Open();
65
66     string loadCompanyQuery = "SELECT companyName FROM company";
67     SqlCommand companyCommand = new SqlCommand(loadCompanyQuery, con);
68     SqlDataReader companyDataReader = companyCommand.ExecuteReader();
69     comboBox2.Items.Clear();
70
71     while (companyDataReader.Read())
72     {
73         comboBox2.Items.Add(companyDataReader["companyName"].ToString());
74     }
75     companyDataReader.Close();
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

```

```

93
94
95 //Load data when selected id is change in customer
96
97 int selectedId = int.Parse(comboBox1.Text);
98 SqlCommand loadCustomer = new SqlCommand($"SELECT * FROM customer WHERE id = {selectedId}", con);
99 SqlDataReader selectedCustomer = loadCustomer.ExecuteReader();
100
101 while (selectedCustomer.Read())
102 {
103     textBox5.Text = selectedCustomer[1].ToString();
104     textBox1.Text = selectedCustomer[2].ToString();
105     textBox2.Text = selectedCustomer[3].ToString();
106     textBox3.Text = selectedCustomer[4].ToString();
107     comboBox2.Text = selectedCustomer[5].ToString();
108 }
109 selectedCustomer.Close();
110
111
112 1 reference
113 private void button1_Click_1(object sender, EventArgs e)
114 {
115     string fName = textBox5.Text;
116     string lName = textBox1.Text;
117     string email = textBox2.Text;
118     string mobile = textBox3.Text;
119     string company = comboBox2.Text;
120
121     SqlCommand saveCompany = new SqlCommand($"INSERT INTO customer(firstName,lastName,email,mobile,company) VALUES('{fName}','{lName}','{email}','{mobile}','{company}');", con);
122     saveCompany.ExecuteNonQuery();
123     textBox5.Clear();
124     textBox1.Clear();
125     textBox2.Clear();
126     textBox3.Clear();
127
128     MessageBox.Show("New customer added");
129 }
130

```

```

138
139
140
141 1 reference
142 private void button4_Click(object sender, EventArgs e)
143 {
144     int selectedId = int.Parse(comboBox1.Text);
145     SqlCommand delete = new SqlCommand($"DELETE FROM customer WHERE id='{selectedId}'", con);
146
147     DialogResult dialogResult = MessageBox.Show(
148         "Are you sure want to delete?", "Delete", MessageBoxButtons.YesNo
149     );
150     if(dialogResult == DialogResult.Yes )
151     {
152         SqlDataReader reader = delete.ExecuteReader();
153         reader.Close();
154
155         MessageBox.Show("Record deleted!");
156
157         this.Close();
158     }
159     else
160     {
161     }
162 }
163
164
165

```

09. DATABASE & TABLES

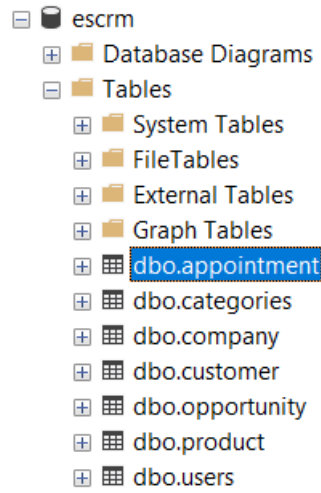


Figure 1 - Database

	Column Name	Data Type	Allow Nulls
▶	id	int	<input type="checkbox"/>
	username	varchar(50)	<input type="checkbox"/>
	password	varchar(50)	<input type="checkbox"/>
			<input type="checkbox"/>

Figure 2- User table

	Column Name	Data Type	Allow Nulls
▶	id	int	<input type="checkbox"/>
	firstName	varchar(50)	<input checked="" type="checkbox"/>
	lastName	varchar(50)	<input checked="" type="checkbox"/>
	email	varchar(50)	<input checked="" type="checkbox"/>
	mobile	int	<input checked="" type="checkbox"/>
	company	varchar(50)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Figure 3 - Customer table

	Column Name	Data Type	Allow Nulls
▶	id	int	<input type="checkbox"/>
	companyName	varchar(50)	<input checked="" type="checkbox"/>
	address	varchar(50)	<input checked="" type="checkbox"/>
	email	varchar(50)	<input checked="" type="checkbox"/>
	telephone	int	<input checked="" type="checkbox"/>
	category	varchar(50)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Figure 4 - Company table

	Column Name	Data Type	Allow Nulls
▶	id	int	<input type="checkbox"/>
	referenceNumber	int	<input type="checkbox"/>
	venue	varchar(50)	<input checked="" type="checkbox"/>
	dateTime	datetime	<input checked="" type="checkbox"/>
	person	varchar(50)	<input checked="" type="checkbox"/>
	topic	varchar(50)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Figure 5 - Appointment table

	Column Name	Data Type	Allow Nulls
▶	id	int	<input type="checkbox"/>
	referenceNumber	int	<input type="checkbox"/>
	company	varchar(50)	<input type="checkbox"/>
	person	varchar(50)	<input type="checkbox"/>
	probability	int	<input type="checkbox"/>
	description	varchar(50)	<input type="checkbox"/>
			<input type="checkbox"/>

Figure 6 - Opportunity table

	Column Name	Data Type	Allow Nulls
▶	id	int	<input type="checkbox"/>
	productCode	int	<input type="checkbox"/>
	productName	varchar(50)	<input type="checkbox"/>
	supplier	varchar(50)	<input type="checkbox"/>
	category	varchar(50)	<input type="checkbox"/>
			<input type="checkbox"/>

Figure 7 - Product table

	Column Name	Data Type	Allow Nulls
▶	id	int	<input type="checkbox"/>
	category	varchar(50)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

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