

# RESTAURANT MANAGEMENT SYSTEM

## TECHNICAL MANUAL

*Group ID: 01*

*Members: Akriti Kumari (120101006)*

*Bharat Nayaran Gupta (120101017)*

*Chandragiri Vinay (120101018)*

*Siddhesh Khandelwal (120101069)*

*Suraj Rautela (120101072)*

# TABLE OF CONTENTS

- Introduction
- Design Consideration
  - I. Assumption and Dependencies
  - II. Detailed Design of the project
- Coding Convention and Tactics

## INTRODUCTION

This project aims at the efficient and effective management of a restaurant with flexibility depending upon the requirements of the user. Given the increase in the number of people visiting restaurants, limited service time and competitiveness, there is a need to manage the activities in a restaurant efficiently in order to increase consumer satisfaction and profits

Various benefits for using this software includes:

- a) better control of food costs->Measures ,control and provides various offers to the customers based on the frequency of their visit and special occasion .
- b) Decrease labor cost->since maximum work is online using this software it cuts the labor cost and enhance profit of restaurant.

c) Smart business decisions ->with the increase in the number of people visiting restaurants, limited service time and competitiveness, there is a need to manage the activities and make smart business decisions in the restaurant in order to increase consumer satisfaction and profits.

## DESIGN CONSIDERATION

This project is developed in Visual Studio 2010 using Visual C++ and MySQL.

- i) **Assumption and Dependencies**-> - It can be installed only on windows based operating system.
- ii) **Detailed Design of the Project**->There are 18 forms and several message boxes in the project .
  - Main Window is the 'login' which consists of 3 labels,2 textboxes and 1 checkbox where the employees and customers are required to provide their username ,password and designation respectively . It also contain 2 link-labels, first link to ' NewCustomer ' and second 'ChangePass' where the customers or employee change their password.There is a button 'login' and on click ,links to any one of 'Manager', 'chef' or 'MenuCard' based on the designation provided in textbox .
  - 'NewCustomer' is the next form having 2 textboxes to give name and mobile no,4 labels and 1 date and time picker to give the date of birth of customer and a button and click on it ,opens a message box having Username and password.
  - 'Changepass' is another form where any employee or customer change their password. It contains 5 labels ,4 textboxes ,to provide username ,oldpassword,new password and reenter password and 1 checkboxes to select their designation and a button .then button when clicked opens a message-box informing whether the password changed or details are not correct.
  - 'MenuCard' is the 4<sup>th</sup> form that contain 13 labels ,5 for item-category

2 for the item-bill and discount –bill and others for showing price ,no of items and packing status. It also contains 3 combo-boxes .Based on the item-category menu appears in the 1<sup>st</sup> combo-box and the other two combo-boxes are used to select no. of items and their packing mode. There is a button 'Add to cart', on clicking whatever is selected in the 3 checkboxes get displayed in the listBox1.Customer can either place or remove orders using two buttons-'Remove' and 'Order'.

- 'Chef' is another form which contains 9 labels, a list-view to display bill number, item name, amount ordered and packing option. There is a button 'accept Order' which on click displays item name, packing and amount in labels and ingredients in the listBox. The form also contain a button 'orders completed' which delivers orders to the customers .
- Manager is another important form having 1 menustrip with Home ,Log Out ,Help( suboptions -i)About, which links to 'About ' form ii)Guide) and Quit options. There are 8 more buttons-Workers list(links to 'workerList' form ),Stock Data(links to 'stockData' form),Menu Items(links to 'menu' form),Sales(links to 'sales'form),Customer List(links to 'customerList' form),Expenditure(links to 'expenditure' form),Discount and Tax Management(links to 'discount' form) and Profit Graph(links to 'profit' form) in the form.
- About is another form which describes about the management system consists of 12 labels , a rich-textbox and a button which when clicked closes the form.
- 'Worklist' is another form having a toolstrip having same function as 'Manager' form. It also contains a listView showing the records of existing employee name and post. Details of the employee like Name,Post,Address,Salary ,Phone Number appears in the labels .There are 3 buttons namely 'Add Employee'which opens another form-'addEmployee' , 'Edit' which opens 'employeeEdit' form and 'delete' removes the employee details from the database.

- 'Add Employee' form adds new employee details in the database, contains 5 levels, 3 text-boxes and a richtext-box to enter the details of employee, a button 'add' which add the details in database if they are correct.
- 'employeeEdit'-provide facilities to manager to edit employee record, contains 6 labels, 4 text-boxes( to enter Name, Salary, Phone Number, Password), a richtextbox to enter address and 'edit' button to save the information to the database.
- 'genbill'-20 labels, 1 listBox and 1 button
- 'StockData'-contains a toolstrip having same menu as in the 'Manager' form. Contains a 'Add New item' button which allows manager to add new items in the stock by providing details (Item Name, Price Per Kg, Units to be Ordered and cost) in the 4 textboxes, a textbox to search by name, listBox showing all the current item-Names, 2 buttons-'Update' (manage the cost) and 'Delete'(remove the selected items from the database) and a chart-box which shows stock Used and Stock Remaining and a 'update' or 'Add items' button to Update stock table of the database.
- 'menu'-contains a toolstrip same as the 'Manager' form, a listView showing the Name, Description, Price, Type of the menu.
  - 8labels, 'Add New Item' button, after click manager can add new items.
  - 3 Textboxes(Name of the item, price, Amount of the ingredient)
  - 1 richtextbox(add description of the items)
  - 2 labels(show all ingredients and select them and second to show the selected ingredients and amount)
  - 1 Checkbox(show and select type of the food items)
  - 3 buttons('Add'-add to list-box and 'Remove'-remove selected items from list-box and 'Add to Menu List'-add enteries to the database).

- 'Sales'-contains a toolstrip same as the 'Manager' form, a listView (showing the Name, Description, Price, Amount Ordered, Total Earning and Type)
  - 1 date and time-picker (to view earnings on that day)
  - 1 chart(total earnings corresponding to each dish)
- 'expenditure'- contains a toolstrip same as the 'Manager' form,
  - listView (showing the Type, Purchase ,Amount, Date)
  - chart1 (shows the amount corresponding to each date in graphical format)
- 'discount'-12 labels
  - 6 numeric up-down(to set discount on frequency of the customer's visit , birthday set VAT,Service Tax, Packing cost )
  - 1 button(save changes in the discount table of the database)
- 'Profit'-1 chart (shows profit)

## CODING CONVENTION AND TACTICS

I am explaining the logic of my code in some of the major forms:

- Login Form

```

2  #include "stdafx.h"
3  #include "NewCustomer1.h"
4  #include "ChangePass.h"
5  #include "MenuCard.h"
6  #include "Form1.h"
7  #include "chef.h"

```

---

line 2-7:includes all the header files

```

192 private: System::Void Login_button_Click(System::Object^ sender, System::EventArgs^ e) {
193     String^ constring=L"datasource=localhost;port=3306;username=root;password=naman";
194     MySqlConnection^ conDataBase=gcnew MySqlConnection(constring);
195     MySqlCommand^ cmdDataBase=gcnew MySqlCommand("select * from restaurant.employee_id where username='"+this->username_txt->Text+"' and password='"+this->password_txt->Text+"'");
196     MySqlDataReader^ myReader;
197     try{
198         conDataBase->Open();
199         myReader=cmdDataBase->ExecuteReader();
200         int count=0;
201         while(myReader->Read()){
202             count=count+1;
203         }
204         if (count==1){
205             if(designation_list->Text=="Manager"){
206                 Form1^ form = gcnew Form1();
207                 form->Show();
208             }
209         }
210     } catch { }
211 }

```

Line192-207:initialize MySql connection ,query to select all from the database-restaurant and employee\_id table where the username ,password and designation are taken from textbox and combobox. If the condition matches then mentioned form open or messagebox .

- Line 234-238:link ' NewCustomer ' form and hide current form using hide() command. Same logic in the line 239-242.

```

234 private: System::Void newuser_link_LinkClicked(System::Object^ sender, System::Windows::Forms::LinkLabelLinkClickedEventArgs^ e) {
235     this->Hide();
236     NewCustomer^ f4 = gcnew NewCustomer();
237     f4->Show();
238 }
239 private: System::Void cngpass_link_LinkClicked(System::Object^ sender, System::Windows::Forms::LinkLabelLinkClickedEventArgs^ e) {
240     this->Hide();
241     ChangePass^ f3 = gcnew ChangePass();
242     f3->Show();
243 }

```

- 'Chef' form :

need to show all the orders that are not delivered , query to select distinct bill no from the database restaurant and from order table whose status is not\_delivered and show in the combobox. In the next function ,details corresponding to the selected bill no is displayed in the datagridview. When deliver button is clicked bill no in the combobox and details in the dataGridView gets removed the status gets updated in the orders\_placed table.

```

MySQLDataReader^ myReader;
String^ constring=L"datasource=localhost;port=3306;username=root;password=abc123";
MySQLConnection^ conDataBase=gcnew MySqlConnection (constring);

MySQLCommand^ cmdDataBase=gcnew MySqlCommand("select DISTINCT bill_no from restaurant.order where status='not_delivered' ";", conDataB;

try{
    conDataBase->Open();
    myReader = cmdDataBase->ExecuteReader();
    while(myReader->Read()){
        String ^ name;
        name=myReader->GetString("bill_no");
        comboBox1->Items->Add(name);
    }
}

catch(Exception^ex){
    MessageBox::Show(ex->Message);
}

private: System::Void button2_Click(System::Object^ sender, System::EventArgs^ e) {
    MySQLDataReader^ myReader;
    int b=System::Convert::ToInt32(this->comboBox1->Text);
    String^ constring=L"datasource=localhost;port=3306;username=root;password=abc123";
    MySQLConnection^ conDataBase=gcnew MySqlConnection (constring);
    //WHERE bil_no ="+"this->comboBox1->Text+"
    MySQLCommand^ cmdDataBase=gcnew MySqlCommand("select bill_no,`items`, `quantity`, `packing` from restaurant.order where bill_no="+this->comboBox1->Text+"");

    try{
        MySQLDataAdapter ^sda=gcnew MySQLDataAdapter();
        sda->SelectCommand =cmdDataBase;
        DataTable^ dbdataset = gcnew DataTable();
        sda->Fill(dbdataset);
        BindingSource^ bSource =gcnew BindingSource();

        bSource->DataSource =dbdataset;
        dataGridView1->DataSource =bSource;
        sda->Update(dbdataset);

    }

    catch(Exception^ex){
        MessageBox::Show(ex->Message);
    }
}

```



```
MySqlCommand^ cmdDataBase=gcnew MySqlCommand("UPDATE restaurant.order SET status='delivered' where bill_no= '"+this->comboBox1->Text+"'");
```

```
try{
    conDataBase->Open();
    myReader = cmdDataBase->ExecuteReader();
    comboBox1->Items->Remove(this->comboBox1->Text);
    this->dataGridView1->DataSource = "";
}
```

```
catch(Exception^ex){
```

```
    MessageBox::Show(ex->Message);
```

- 'NewCustomer'-New customer provide details in the text boxes and date and time picker. Then check if the mobile no is of 10 digit or not, if yes then all the entries are added in the customer\_id table and employee\_id table.

```
199 MySqlCommand^ cmdDataBase2=gcnew MySqlCommand("select * from restaurant.customer_id where mobile='"+this->mobile_txt->Text+"'";,conDataBase);
200 MySqlDataReader^ myReader2;
201 try{
202     conDataBase->Open();
203     myReader2=cmdDataBase2->ExecuteReader();
204     while(myReader2->Read()){
205         count=count+1;
206     }
207 }catch(Exception^ex){
208
209     MessageBox::Show(ex->Message);
210 }
211 conDataBase->Close();}
212 count==0){
213 if(name_txt->Text!=""){
214     MySqlCommand^ cmdDataBase=gcnew MySqlCommand("INSERT INTO restaurant.customer_id (idcustomer_id,name,mobile,birthdate,password,last_visit,f
215     MySqlDataReader^ myReader;
216     try{
217         conDataBase->Open();
218         myReader=cmdDataBase->ExecuteReader();
219
220     }catch(Exception^ex){
221
222         MessageBox::Show(ex->Message);
223     }
```

```

219
220     }catch(Exception^ex){
221
222         MessageBox::Show(ex->Message);
223     }
224     conDataBase->Close();
225     cmdDataBase=gcnew MySqlCommand("INSERT INTO restaurant.employee_id (username,name,designation,password) values '"+System::Convert::ToString
226     try{
227         conDataBase->Open();
228         myReader=cmdDataBase->ExecuteReader();
229
230     }catch(Exception^ex){
231
232         MessageBox::Show(ex->Message);
233     }
234     conDataBase->Close();
235     MessageBox::Show("Your username is : "+var_id+" and password is : "+mobile_txt->Text);
236     showLogin();
237     this->Hide();
238 }
239 else MessageBox::Show("Please enter a name");
240
241 MessageBox::Show("You already have a registered mobile number.....Try Again");
242
243 MessageBox::Show("Please enter a 10 digit mobile number");

```

‘Add Employee’ Form: Manager adds new employee in the employee table. (line 239-254) Generate username and password and (255-263) update employee table and (265-275) insert entries in ‘employee\_id’ table.

```

229 MySqlCommand^ cmdDatabase= gcnw MySqlCommand("INSERT INTO `restaurant`.`employee`(`EmployeeName`,`EmployeePost`,`EmployeeAddress`,`EmployeeP
230 MySqlDataReader^ myReader;
231 try{
232     conDatabase->Open();
233     myReader = cmdDatabase->ExecuteReader();
234
235 }
236 catch(Exception^ ex){
237     MessageBox::Show(ex->Message);
238 }
239 conDatabase->Close();
240 cmdDatabase = gcnw MySqlCommand("SELECT * FROM `restaurant`.`employee` WHERE `UserName` = '"+textBox1->Text+textBox3->Text+"'";conDatabase)
241 String^ userN;
242 int id;
243 try{
244     conDatabase->Open();
245     myReader = cmdDatabase->ExecuteReader();
246     while(myReader->Read()){
247         id = myReader->GetInt32(0);

```

```

256 cmdDatabase = gcnw MySqlCommand("UPDATE `restaurant`.`employee` SET `UserName` = '"+userN+"',Password = '"+userN+"` WHERE `idEmployee` = "+i
257 try{
258     conDatabase->Open();
259     myReader = cmdDatabase->ExecuteReader();
260 }
261 catch(Exception^ ex){
262     MessageBox::Show(ex->Message);
263 }
264 conDatabase->Close();
265 cmdDatabase = gcnw MySqlCommand("INSERT INTO `restaurant`.`employee_id`(`username`,`name`,`designation`,`password`) VALUES ('"+userN+"','"+t
266 try{
267     conDatabase->Open();
268     myReader = cmdDatabase->ExecuteReader();
269 }
270 catch(Exception^ ex){
271     MessageBox::Show(ex->Message);
272 }
273 conDatabase->Close();

```

- ‘ChangePass’:In the first query we select that entry from the employee\_id table based on the input of the textboxes and change password .2<sup>nd</sup> and 3<sup>rd</sup> query we update customer\_id and employee\_id table of the database.

```

207 MySqlConnection^ conDataBase=gcnw MySqlConnection(constring);
208 MySqlCommand^ cmdDataBase=gcnw MySqlCommand("select * from restaurant.employee_id where username='"+this->user_txt->Text+"' and password='";
209 MySqlDataReader^ myReader;
210 try{
211     conDataBase->Open();
212     myReader=cmdDataBase->ExecuteReader();
213     int count=0;
214     while(myReader->Read()){
215         count=count+1;
216     }
217     if (count==1){
218
219         MessageBox::Show("Your password is successfully changed");
220         conDataBase->Close();
221
222         //MessageBox::Show("Your password is successfully changed");
223         conDataBase->Close();
224         MySqlCommand^ cmdDataBase1=gcnw MySqlCommand("update restaurant.employee_id set password='"+this->newpass_txt->Text+"'where usernam
225         MySqlDataReader^ myReader1;
226         try{
227             conDataBase->Open();
228             myReader1=cmdDataBase1->ExecuteReader();
229         }catch(Exception^ex){
230
231             MessageBox::Show(ex->Message);
232         }
233         conDataBase->Close();
234
235         cmdDataBase1=gcnw MySqlCommand("update restaurant.customer_id set password='"+this->newpass_txt->Text+"'where idcustomer_id="+Syste
236         try{
237             conDataBase->Open();
238             myReader1=cmdDataBase1->ExecuteReader();
239         }catch(Exception^ex){
240
241             MessageBox::Show(ex->Message);
242         }
243         conDataBase->Close();
244         showLogin();
245         this->Hide();
246     }

```

- 'profit' Form: Based on the calculated mindate ,first query(125-132) we calculate profit using 'bill' table and second query(133-140) calculate loss from 'expense' table. The net profit is calculated (142-145) and displayed on the chart.

■

```

123 String^ connectionString="datasource=127.0.0.1;port=3306;username=root;password=abc123";
124 MySqlConnection^ conDatabase= gcnew MySqlConnection(connectionString);
125 MySqlCommand^ cmdDatabase= gcnew MySqlCommand("SELECT `amount` FROM `restaurant`.`bill` WHERE `bill`.`date` LIKE '%" + mdate + "%'", conDatabase);
126 MySqlDataReader^ myReader;
127 conDatabase->Open();
128 myReader = cmdDatabase->ExecuteReader();
129 double prof = 0;
130 while(myReader->Read())

134 cmdDatabase= gcnew MySqlCommand("SELECT `amount` FROM `restaurant`.`expenses` WHERE `expenses`.`date` LIKE '%" + mdate + "%'", conDatabase);
135 conDatabase->Open();
136 myReader = cmdDatabase->ExecuteReader();
137 double loss = 0;
138 while(myReader->Read()){
139     loss = loss + myReader->GetDouble(0);
140 }
141 conDatabase->Close();
142 this->chart1->Series["Profit"]->Points->AddXY(mdate, prof+loss);
143 pmdate = pmdate->AddDays(1);
144 while(pmdate->Day != tdate->Day || pmdate->Month != tdate->Month || pmdate->Year != tdate->Year );
145

```

- 'sales' form: Display Name, Description ,Price and Amount Ordered ,total earning,type in date selected in the datepicker.

```

268 MySqlConnection^ conDatabase= gcnew MySqlConnection(connectionString);
269 MySqlCommand^ cmdDatabase= gcnew MySqlCommand("SELECT * FROM `restaurant`.`menu`", conDatabase);
270 MySqlDataReader^ myReader;

```

```

289 DateTime^ date = dateTimePicker1->Value.Date;
290 String^ sqldate = System::Convert::ToString(dateTimePicker1->Value.Year)+"-"+System::Convert::ToString(dateTimePicker1->Value.Month)+"-"+System::Convert::ToString(dateTimePicker1->Value.Day);
291 conDatabase->Close();
292 connectionString=L"datasource=127.0.0.1;port=3306;username=root;password=abc123";
293 conDatabase= gcnew MySqlConnection(connectionString);
294 cmdDatabase= gcnew MySqlCommand("SELECT `items`,`quantity` FROM `restaurant`.`order` WHERE `Date` = '"+sqldate+"'",conDatabase);
295 try{
296     conDatabase->Open();
297     myReader = cmdDatabase->ExecuteReader();
298     while(myReader->Read()){

```

- **EmployeeEdit:**In the first query update employee table based on data provided and in the 2<sup>nd</sup> query update employee\_id table.

```

235 if(textBox1->Text != "" && textBox3->Text != "" && comboBox1->Text != "" && richTextBox1->Text != "" && textBox2->Text != "" && is_number(textBox2->Text))
236     if(MessageBox::Show("Are you sure you want to edit the entry to the database?", "Edit Entry", MessageBoxButtons::YesNo, MessageBoxIconQuestion) == DialogResult::Yes)
237     {
238         String^ connectionString=L"datasource=127.0.0.1;port=3306;username=root;password=abc123";
239         MySqlConnection^ conDatabase= gcnew MySqlConnection(connectionString);
240         MySqlCommand^ cmdDatabase= gcnew MySqlCommand("UPDATE `restaurant`.`employee` SET `EmployeeName`='"+textBox1->Text+"',`EmployeePost`='"+textBox2->Text+"',`EmployeeId`='"+textBox3->Text+"',`Designation`='"+comboBox1->Text+"' WHERE `EmployeeId`='"+textBox4->Text+"'",conDatabase);
241         MySqlDataReader^ myReader;
242         try{
243             conDatabase->Open();
244             myReader = cmdDatabase->ExecuteReader();

```

```

239 MySqlCommand^ cmdDatabase= gcnew MySqlCommand("UPDATE `restaurant`.`employee` SET `EmployeeName`='"+textBox1->Text+"',`EmployeePost`='"+textBox2->Text+"',`EmployeeId`='"+textBox3->Text+"',`Designation`='"+comboBox1->Text+"' WHERE `EmployeeId`='"+textBox4->Text+"'",conDatabase);
240 MySqlDataReader^ myReader;
241 try{
242     conDatabase->Open();
243     myReader = cmdDatabase->ExecuteReader();
244 }
245 catch(Exception^ ex){
246     MessageBox::Show(ex->Message);
247 }
248 conDatabase->Close();
249 cmdDatabase = gcnew MySqlCommand("UPDATE restaurant.employee_id SET name = '"+textBox1->Text+"', designation = '"+comboBox1->Text+"', password = '"+textBox2->Text+"' WHERE employee_id = '"+textBox3->Text+"'",conDatabase);
250 try{
251     conDatabase->Open();
252     myReader = cmdDatabase->ExecuteReader();
253 }

```

- Menu Form: Based on the category of the items selected menu appears in the combobox 1 and a query is made to check whether the items ordered is available in stock table or not .Calculate discount and update stock table if order is placed.

```

550         MySqlCommand^ cmdDataBase1=gcnew MySqlCommand("select itemRem from restaurant.stock where itemName='"+itemName+"";,con
551         MySqlDataReader^ myreader1;
552         try{
553             conDataBase1->Open();
554             myreader1 = cmdDataBase1->ExecuteReader();
555             while(myreader1->Read()){
556                 int possNum = 0;
557                 double stockLeft = myreader1->GetDouble("itemRem");
558

```

```

486         MySqlCommand^ cmdDataBase=gcnew MySqlCommand("select * from restaurant.menu where itemName='"+comboBox1+"";,conDataBase
487         MySqlDataReader^ myreader;
488
489
490         try{
491
492             conDataBase->Open();
493             myreader=cmdDataBase->ExecuteReader();
494             while(myreader->Read())
495
496
497             {
498                 label2->Text=myreader->GetDouble("itemPrice").ToString();
499             }
500
501

```

```

1145         MySqlCommand^ cmdDataBase2=gcnew MySqlCommand("select * from restaurant.discount where name='Service Tax';",conDataBase);
1146         MySqlDataReader^ myreader2;
1147
1148
1149         try{
1150
1151             conDataBase->Open();
1152             myreader2=cmdDataBase2->ExecuteReader();
1153             while(myreader2->Read())

```

---

```

816 MySqlCommand^ cmdDataBase3 = gcnew MySqlCommand("UPDATE `restaurant`.`stock` SET `itemRem` = `itemRem` - "+amt+",`itemUsed` = `itemUsed` +
817 MySqlCommand^ myreader3;
818 try{
819     conDataBase1->Open();
820     myreader3 = cmdDataBase3->ExecuteReader();
821 }
822 catch(Exception^ ex){
823     MessageBox::Show(ex->Message,"Error");
824 }
825 conDataBase1->Close();
826
827

```

```

840         /*****GIVING DISCOUNT*****/
841         double b=System::Convert::ToDouble(label4->Text);
842         //int birthdaydisc=0;
843         double c=b;
844         System::DateTime^ no = System::DateTime::Now;
845         String^ date=System::Convert::ToString(no->Month) + "/" + System::Convert
846
847
848         int freq=give_frequency();
849         int bd=0;
850         int ap=0;
851         int st=0;
852         int vt=0;
853         int fq = 0;
854         int q = 0;
855         DateTime^ birth=get_birthday());

```



- 'discount' Form: Manager can change the discount-rate. based on the input data Update discount table.

```

403 conDatabase->Close();
404 cmdDatabase= gcnew MySqlCommand("UPDATE `restaurant`.`discount` SET `amount` = "+System::Convert::ToInt32(numericUpDown2->Text)+
405 try{
406     conDatabase->Open();
407     myReader = cmdDatabase->ExecuteReader();
408 }
409 catch(Exception^ ex)
410 {
411     String^ q = ex->Message;
412     MessageBox::Show(q, "Error");
413 }
414 conDatabase->Close();
415 cmdDatabase= gcnew MySqlCommand("UPDATE `restaurant`.`discount` SET `amount` = "+System::Convert::ToInt32(numericUpDown3->Text)+
416 try{
417     conDatabase->Open();
418     myReader = cmdDatabase->ExecuteReader();
419 }
420 catch(Exception^ ex)
421 {
422     String^ q = ex->Message;
423     MessageBox::Show(q, "Error");
424 }
425 conDatabase->Close();
426 cmdDatabase= gcnew MySqlCommand("UPDATE `restaurant`.`discount` SET `amount` = "+System::Convert::ToInt32(numericUpDown4->Text)+
427
428
429 cmdDatabase= gcnew MySqlCommand("UPDATE `restaurant`.`discount` SET `amount` = "+System::Convert::ToInt32(numericUpDown5->Text)+
430 try{
431     conDatabase->Open();
432     myReader = cmdDatabase->ExecuteReader();
433 }
434 catch(Exception^ ex)
435 {
436     String^ q = ex->Message;
437     MessageBox::Show(q, "Error");
438 }

```

- 'WorkerList' Form: first query delete employee record from 'employee' table and other shows the details of the selected employee in the labels.

```

431 MySqlCommand^ cmdDatabase= gcnew MySqlCommand("DELETE FROM `restaurant`.`employee` WHERE `employee`.`UserName` = '" + label12->Text + "'")
432 MySqlDataReader^ myReader;
433 try{
434     conDatabase->Open();
435     myReader = cmdDatabase->ExecuteReader();
436
437
438 }
439 catch(Exception^ ex){
440     MessageBox::Show(ex->Message);
441 }

```

```

507 label10->Visible = true;
508 button3->Visible = true;
509 button4->Visible = true;
510 label11->Visible = true;
511 label12->Visible = true;
512 ListViewItem^ objName = listView1->Items[listView1->FocusedItem->Index];
513 String^ itName = objName->SubItems[0]->Text;
514 String^ connectionString=L"datasource=127.0.0.1;port=3306;username=root;password=abc123";
515 MySqlConnection^ conDatabase= gcnew MySqlConnection(connectionString);
516 MySqlCommand^ cmdDatabase= gcnew MySqlCommand("SELECT * FROM `restaurant`.`employee` WHERE `employee`.`EmployeeName` = '" + itName + "'");
517 MySqlDataReader^ myReader;
518 button4->Visible = true;
519 try{
520     conDatabase->Open();
521     myReader = cmdDatabase->ExecuteReader();
522
523     while(myReader->Read()){
524         label5->Text = (myReader->GetString(1));
525         label6->Text = (myReader->GetString(2));
526         label7->Text = (myReader->GetString(3));
527         label8->Text = (myReader->GetString(4));

```