C# Project

Wykonawca: Roman Uliashyn

Temat: „Restaurant Billing System”

Treść

1. Cel
2. Jak Działa
3. Środowisko programowe
4. Interfacy
5. Stan Programu – “discounts”
6. Kod Żriódłowy

Cel:

System zamówienia w restauracji.  
Naprzykład w restauracji lub na barze kelner robi zamówienie. Za tym trzeba wbić wszystko do systemu , żeby wydrukować paragon. Ten program reprezentuję tą sytuacje.

Jak dziala

Po otwieraniu pliku CafeManagement\bin\Debug\CafeManagmen.exe – otwiera się okienko z interfejsem Aplikacji.   
Dalej użytkownik muśi wypełniać formatkę intyitywnie.

Algorythm:

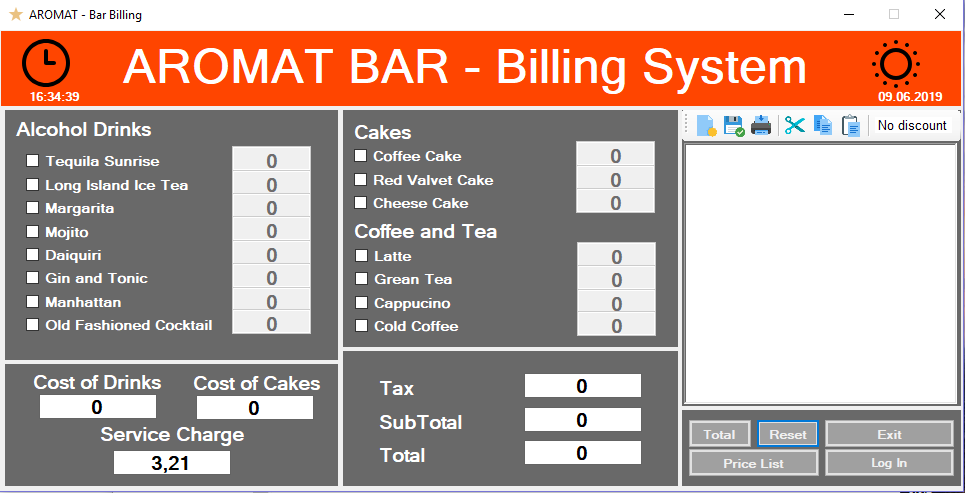
1. Zaznaczyć produkt
2. Ustawić ilość produktów
3. Kliknąć na przycisc TOTAL
   1. Jeśli nie było przed tym zalogowanej osoby – proram poprosi zalogować się. Jeśli logowanie przeszło przed nacisknięciem – pojawi się paragon w Text Boxie
4. Po pojawieniu paragonu w TextBoxsie , wybrać potrzebną funkcje w ToolBoxie.
   1. Nowy Arkusz
   2. Zapisz
   3. Drukuj
   4. Wycnij
   5. Copja
   6. Wstawić
5. Sprawdżić ceny
6. Zmienić osobę

Śriodowisko programowe

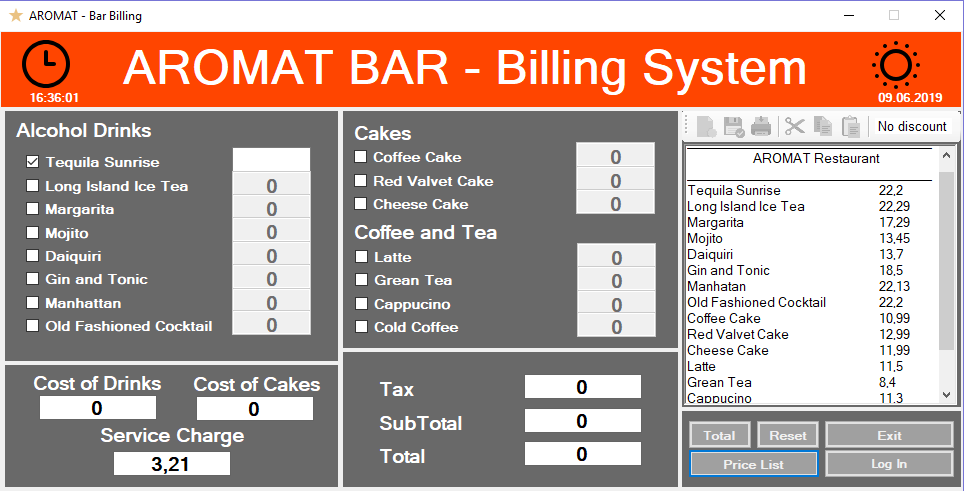
Visual Studio 15.

Interfacy

-Po uruchameniu

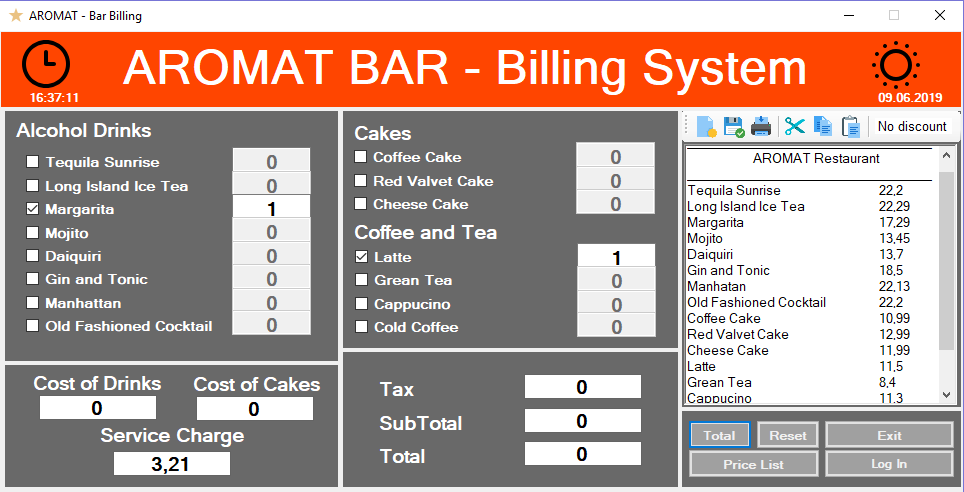


-Przycisk Price list

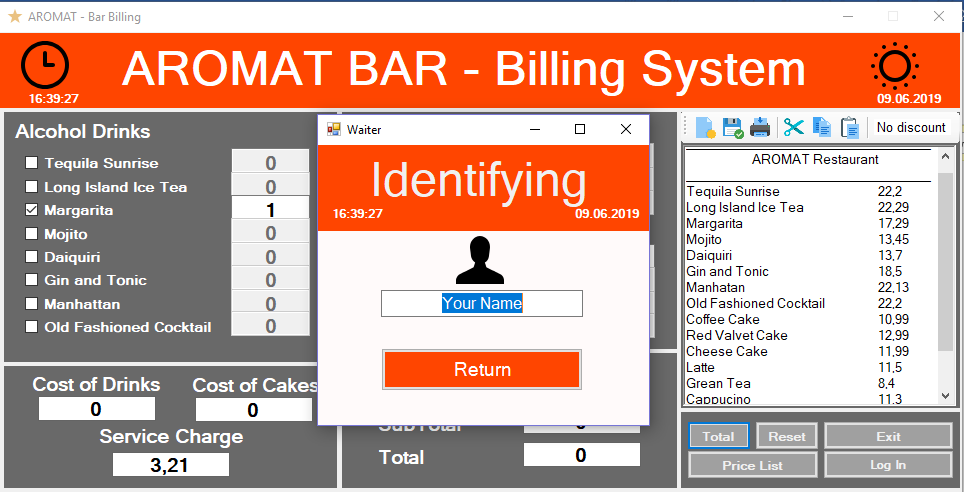


Toolstrip w tym momęcie jest zablokowany, żeby nie można było kopiować,zmieniać dane cen.

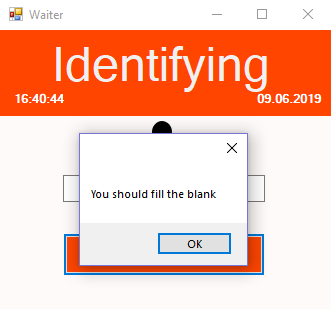
-Wybór produktu oraz ustawienie ilości przez klawiaturę



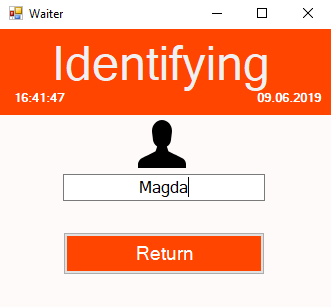
-Po naćisku na total , program wyświetli okienko z logowaniem , skoro nie zrobione to przed tym



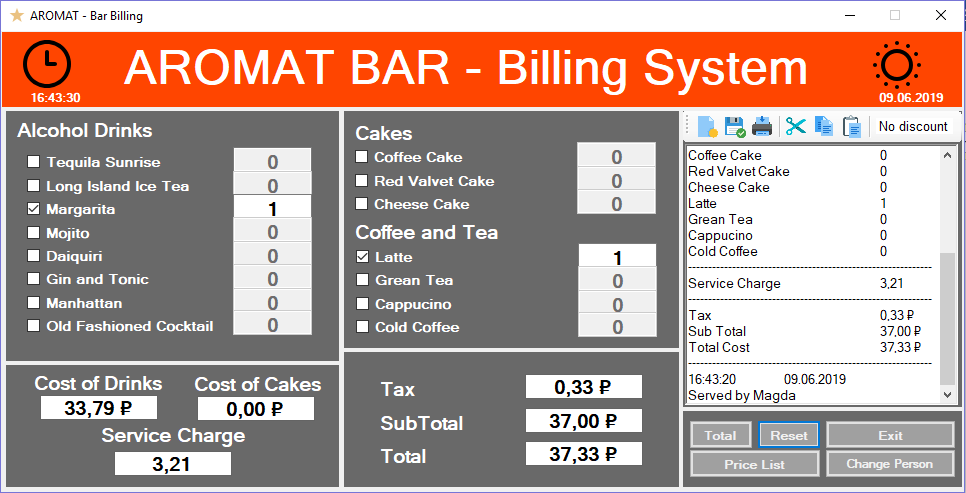
-Klikając odrazu na return – program powie, że trzeba wpisać imię:



-Klikając ok , zwróci się stan wypełnienia. Wypełnając można kliknąc od razu enter – zadziała zamknięcie okienka

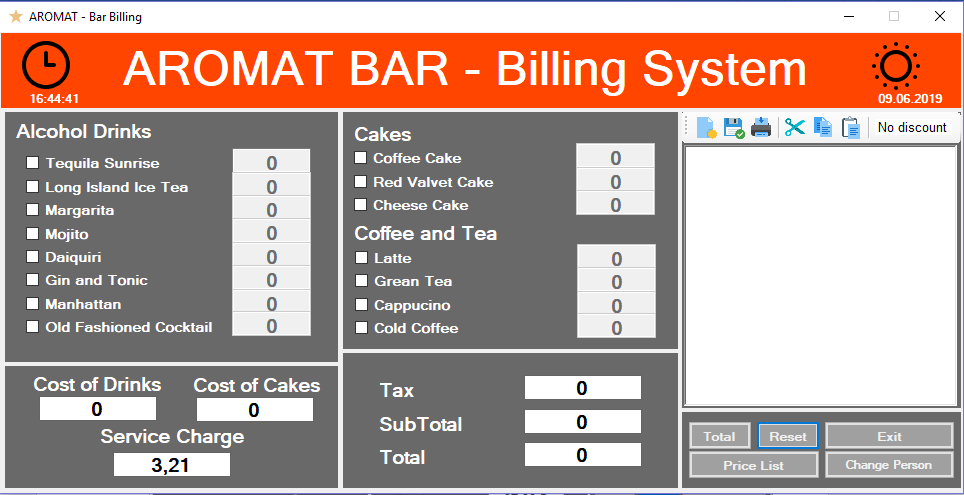


-Teraz musimy kliknąć na przycisk „total” – ponownie



Pokazany jest paragon + imię kelnera

-Podświetlony jest przycisk Reset, oczyszcza on wszystkie ustawienia (wybrane produkty,textBox z paragonem , pola z obliczeniami):



Stan programu

Po za intuitywnymi działaniami , program ma stan z rabatem na Drinki oraz Cakes.

Działa to w taki sposób:

1. Program po uruchamieniu sprawdza status discounta. Ten status zależny od periodu w czasie. Mam takie ograniczenia:
   1. Jeśli teraz jesteśmy pomiędzy dziesiątą a czternastą > 0 , < 13 , rabat jest dla Cakes.
   2. Jeśli teraz jesteśmy w periodzie on 19 do 24 , to mamy rabat na wszystkie napoje
2. Wypisuje status a oraz szczegóły rabatów w TextBox
3. Ma wpłył na cene.

Rabat ustawiony ny 15 procentów. Formuła:

private double percentoff = 0.15;

if (discount\_cake)

{

cakeCosts \*= 1 - percentoff;

}

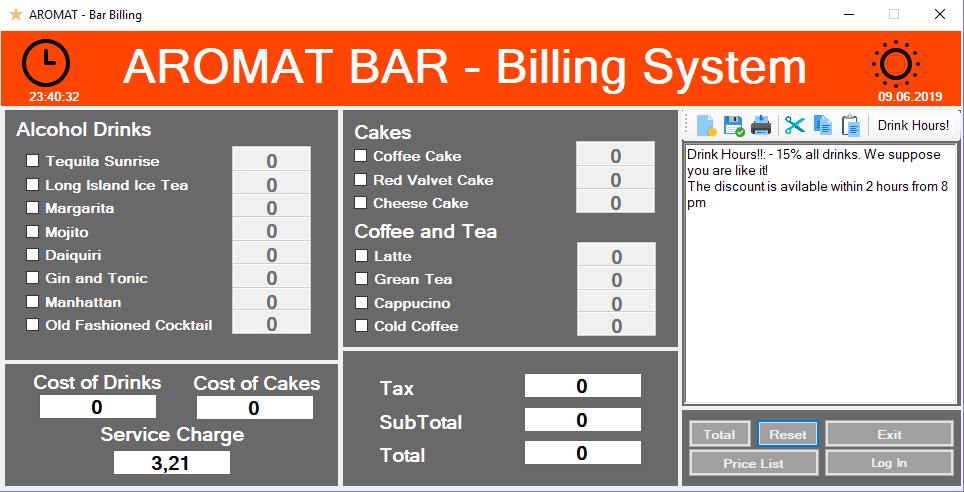
else

{

drinkCosts \*= 1 - percentoff;

}

1. Jeśli jest już po czasie, przy każdym czisknięciu Rest , aktualizuje się program w zwyykły stan



Ważne funkcje:

Check\_Discount

private void Check\_Discount()

{

if (clone\_disc\_status && disc\_status)

{

if (discount\_cake)

{

lbDiscount.Text = "Cake Hour!";

rtfReceipt.AppendText("Happy Hour: - 15% on cake!" + Environment.NewLine);

rtfReceipt.AppendText("The discount is avilable within 3 hour from 9 am" + Environment.NewLine);

clone\_disc\_status = false;

}

else

{

lbDiscount.Text = "Drink Hours!";

rtfReceipt.AppendText("Drink Hours!!: - 15% all drinks. We suppose you are like it!" + Environment.NewLine);

rtfReceipt.AppendText("The discount is avilable within 2 hours from 8 pm" + Environment.NewLine);

clone\_disc\_status = false;

}

}

else

{

lbDiscount.Text = "No discount";

}

clone\_disc\_status = true;

}

CheckStatus:

private bool CheckStatus(DateTime now)

{

if (now.Hour > 9 && now.Hour < 13)

{

disc\_status = true;

discount\_cake = true;

}

else if (now.Hour > 19 && now.Hour < 24)

{

disc\_status = true;

discount\_alc = true;

}

else

{

discount\_alc = false;

discount\_cake = false;

disc\_status = false;

}

return disc\_status;

}

Kod Żriódłowy

**USING CASE**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

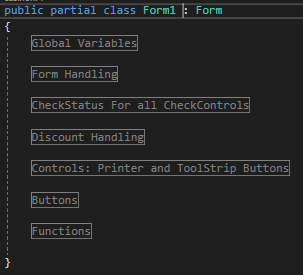
using System.Threading.Tasks;

using System.Windows.Forms;

Dwie klasy:

**Form1** – Główna formatka a **Waiter** Form – logowanie kelnera.

**Form1** – ma takie regiony



Dalej reprezentowany kod w formacie: Region – Kod

#region Global Variables

public static string persons;

public double Teq, Long, Margar, Mojito, Daiquiri, GinTon, Manhatt, OldFash;

public double cCake, rValvet, cheeseCake;

public double Latte, GreenTea, Cappucino, ColdCoffee;

private bool disc\_status , clone\_disc\_status = true;

private bool discount\_cake = false, discount\_alc = false;

private double percentoff = 0.15;

public bool log = false;

#endregion

#region Form Handling

public Form1()

{

InitializeComponent();

button1.Focus();

}

private void Form1\_Load(object sender, EventArgs e)

{

var now = DateTime.Now;

lblDate.Text = DateTime.Now.ToShortDateString();

timer1.Start();

Enabling();

rtfReceipt.Clear();

CheckStatus(now);

Check\_Discount();

SetVar();

}

#endregion

//Jeśli punkt został zaznaczony – możesz wpisać ilość produktu , jak nie ustawia w zwykły stan

#region CheckStatus For all CheckControls

private void chkTequila\_CheckedChanged(object sender, EventArgs e)

{

if (chkTequilaSun.Checked==true)

{

txtTequilaSun.Enabled = true;

}

else

{

txtTequilaSun.Enabled = false;

txtTequilaSun.Text = "0";

}

}

private void txtTequila\_Click(object sender, EventArgs e)

{

txtTequilaSun.Text = "";

txtTequilaSun.Focus();

}

private void chkLong\_CheckedChanged(object sender, EventArgs e)

{

if (chkLong.Checked == true)

{

txtLong.Enabled = true;

}

else

{

txtLong.Enabled = false;

txtLong.Text = "0";

}

}

private void txtLong\_Click(object sender, EventArgs e)

{

txtLong.Text = "";

txtLong.Focus();

}

private void chkMargarita\_CheckedChanged(object sender, EventArgs e)

{

if (chkMargarita.Checked == true)

{

txtMargarita.Enabled = true;

}

else

{

txtMargarita.Enabled = false;

txtMargarita.Text = "0";

}

}

private void txtMargarita\_Click(object sender, EventArgs e)

{

txtMargarita.Text = "";

txtMargarita.Focus();

}

private void chkMojito\_CheckedChanged(object sender, EventArgs e)

{

if (chkMojito.Checked == true)

{

txtMojito.Enabled = true;

}

else

{

txtMojito.Enabled = false;

txtMojito.Text = "0";

}

}

private void txtMojito\_Click(object sender, EventArgs e)

{

txtMojito.Text = "";

txtMojito.Focus();

}

private void chkDaiquiri\_CheckedChanged(object sender, EventArgs e)

{

if (chkDaiquiri.Checked == true)

{

txtDaiquiri.Enabled = true;

}

else

{

txtDaiquiri.Enabled = false;

txtDaiquiri.Text = "0";

}

}

private void txtDaiquiri\_Click(object sender, EventArgs e)

{

txtDaiquiri.Text = "";

txtDaiquiri.Focus();

}

private void chkGinTon\_CheckedChanged(object sender, EventArgs e)

{

if (chkGinTon.Checked == true)

{

txtGinTon.Enabled = true;

}

else

{

txtGinTon.Enabled = false;

txtGinTon.Text = "0";

}

}

private void txtGinTon\_Click(object sender, EventArgs e)

{

txtGinTon.Text = "";

txtGinTon.Focus();

}

private void chkManhattan\_CheckedChanged(object sender, EventArgs e)

{

if (chkManhattan.Checked == true)

{

txtManhattan.Enabled = true;

}

else

{

txtManhattan.Enabled = false;

txtManhattan.Text = "0";

}

}

private void txtManhattan\_Click(object sender, EventArgs e)

{

txtManhattan.Text = "";

txtManhattan.Focus();

}

private void chkOldFash\_CheckedChanged(object sender, EventArgs e)

{

if (chkOldFash.Checked == true)

{

txtOldFash.Enabled = true;

}

else

{

txtOldFash.Enabled = false;

txtOldFash.Text = "0";

}

}

private void txtOldFash\_Click(object sender, EventArgs e)

{

txtOldFash.Text = "";

txtOldFash.Focus();

}

private void chkCoffe\_CheckedChanged(object sender, EventArgs e)

{

if (chkCoffe.Checked == true)

{

txtCoffeCake.Enabled = true;

}

else

{

txtCoffeCake.Enabled = false;

txtCoffeCake.Text = "0";

}

}

private void txtCoffeCake\_Click(object sender, EventArgs e)

{

txtCoffeCake.Text = "";

txtCoffeCake.Focus();

}

private void chkRedValvet\_CheckedChanged(object sender, EventArgs e)

{

if (chkRedValvet.Checked == true)

{

txtRedValvetCake.Enabled = true;

}

else

{

txtRedValvetCake.Enabled = false;

txtRedValvetCake.Text = "0";

}

}

private void txtRedValvetCake\_Click(object sender, EventArgs e)

{

txtRedValvetCake.Text = "";

txtRedValvetCake.Focus();

}

private void chkCheeseCake\_CheckedChanged(object sender, EventArgs e)

{

if (chkCheeseCake.Checked == true)

{

txtCheeseCake.Enabled = true;

}

else

{

txtCheeseCake.Enabled = false;

txtCheeseCake.Text = "0";

}

}

private void txtCheeseCake\_Click(object sender, EventArgs e)

{

txtCheeseCake.Text = "";

txtCheeseCake.Focus();

}

private void chkLatte\_CheckedChanged(object sender, EventArgs e)

{

if (chkLatte.Checked == true)

{

txtLatte.Enabled = true;

}

else

{

txtLatte.Enabled = false;

txtLatte.Text = "0";

}

}

private void txtLatte\_Click(object sender, EventArgs e)

{

txtLatte.Text = "";

txtLatte.Focus();

}

private void chkGreenTea\_CheckedChanged(object sender, EventArgs e)

{

if (chkGreenTea.Checked == true)

{

txtGreenTea.Enabled = true;

}

else

{

txtGreenTea.Enabled = false;

txtGreenTea.Text = "0";

}

}

private void txtKGreenTea\_Click(object sender, EventArgs e)

{

txtGreenTea.Text = "";

txtGreenTea.Focus();

}

private void chkCappucino\_CheckedChanged(object sender, EventArgs e)

{

if (chkCappucino.Checked == true)

{

txtCappucino.Enabled = true;

}

else

{

txtCappucino.Enabled = false;

txtCappucino.Text = "0";

}

}

private void txtCappucino\_Click(object sender, EventArgs e)

{

txtCappucino.Text = "";

txtCappucino.Focus();

}

private void chkColdCoffe\_CheckedChanged(object sender, EventArgs e)

{

if (chkColdCoffee.Checked == true)

{

txtColdCofee.Enabled = true;

}

else

{

txtColdCofee.Enabled = false;

txtColdCofee.Text = "0";

}

}

private void txtColdCoffe\_Click(object sender, EventArgs e)

{

txtColdCofee.Text = "";

txtColdCofee.Focus();

}

#endregion

#region Discount Handling

private void lbDiscount\_Click(object sender, EventArgs e)

{

Check\_Discount();

}

/// <summary>

/// Timer Ticker

/// </summary>

private void timer1\_Tick(object sender, EventArgs e)

{

var now = DateTime.Now;

lblTimer.Text = DateTime.Now.ToLongTimeString();

CheckStatus(now);

}

/// <summary>

/// Get Discount Status & Vsability

/// </summary>

private void Check\_Discount()

{

if (clone\_disc\_status && disc\_status)

{

if (discount\_cake)

{

lbDiscount.Text = "Cake Hour!";

rtfReceipt.AppendText("Happy Hour: - 15% on cake!" + Environment.NewLine);

rtfReceipt.AppendText("The discount is avilable within 3 hour from 9 am" + Environment.NewLine);

clone\_disc\_status = false;

}

else

{

lbDiscount.Text = "Drink Hours!";

rtfReceipt.AppendText("Drink Hours!!: - 15% all drinks. We suppose you are like it!" + Environment.NewLine);

rtfReceipt.AppendText("The discount is avilable within 2 hours from 8 pm" + Environment.NewLine);

clone\_disc\_status = false;

}

}

else

{

lbDiscount.Text = "No discount";

}

clone\_disc\_status = true;

}

/// <summary>

/// Set CheckStatus

/// </summary>

private bool CheckStatus(DateTime now)

{

if (now.Hour > 9 && now.Hour < 13)

{

disc\_status = true;

discount\_cake = true;

}

else if (now.Hour > 19 && now.Hour < 24)

{

disc\_status = true;

discount\_alc = true;

}

else

{

discount\_alc = false;

discount\_cake = false;

disc\_status = false;

}

return disc\_status;

}

#endregion

#region Controls: Printer and ToolStrip Buttons

private void printDocument1\_PrintPage(object sender, System.Drawing.Printing.PrintPageEventArgs e)

{

e.Graphics.DrawString(rtfReceipt.Text, new Font("Arial", 23, FontStyle.Regular), Brushes.Black, 70, 70);

}

private void printToolStripButton\_Click(object sender, EventArgs e)

{

printPreviewDialog1.Document = printDocument1;

printPreviewDialog1.ShowDialog();

}

private void newToolStripButton\_Click(object sender, EventArgs e)

{

rtfReceipt.Clear();

}

private void cutToolStripButton\_Click(object sender, EventArgs e)

{

rtfReceipt.Cut();

}

private void copyToolStripButton\_Click(object sender, EventArgs e)

{

rtfReceipt.Copy();

}

private void pasteToolStripButton\_Click(object sender, EventArgs e)

{

rtfReceipt.Paste();

}

private void saveToolStripButton\_Click(object sender, EventArgs e)

{

//this code will save text files

SaveFileDialog saveFile = new SaveFileDialog();

saveFile.FileName = "Check.txt";

saveFile.Filter = "Text Files (\*.txt)|\*.txt|All files (\*.\*)|\*.\*";

if (saveFile.ShowDialog() == System.Windows.Forms.DialogResult.OK)

{

using (System.IO.StreamWriter sw = new System.IO.StreamWriter(saveFile.FileName))

sw.WriteLine(rtfReceipt.Text);

}

}

#endregion

#region Buttons

/// <summary>

/// Total handling.

/// </summary>

private void button1\_Click(object sender, EventArgs e)

{

//btnTotal

toolStrip1.Enabled = true;

double tax;

tax = 0.89;

//Coffee

double Tequila = Convert.ToDouble(txtTequilaSun.Text);

double Margarita = Convert.ToDouble(txtMargarita.Text);

double LongCoctail = Convert.ToDouble(txtLong.Text);

double MojitoCoc = Convert.ToDouble(txtMojito.Text);

double DaiquiriCoc = Convert.ToDouble(txtDaiquiri.Text);

double GinTonCoc = Convert.ToDouble(txtGinTon.Text);

double ManHattanCoc = Convert.ToDouble(txtManhattan.Text);

double OldFashCoc = Convert.ToDouble(txtOldFash.Text);

//Cakes

double cofCake = Convert.ToDouble(txtCoffeCake.Text);

double redValvet = Convert.ToDouble(txtRedValvetCake.Text);

double chCheeseCk = Convert.ToDouble(txtCheeseCake.Text);

//Coffee

double LatteDrink = Convert.ToDouble(txtLatte.Text);

double GreenTeaDrink = Convert.ToDouble(txtGreenTea.Text);

double CappucinoDrink = Convert.ToDouble(txtCappucino.Text);

double ColdCoffeeDrink = Convert.ToDouble(txtColdCofee.Text);

if (IsAllEmpty())

{

MessageBox.Show("The list is Empty" + Environment.NewLine + "Fill the bar blank");

}

else

{

if (log)

{

Waiter.Text = "Change Person";

double drinkCosts = (Tequila \* Teq) + (Margarita \* Long) + (LongCoctail \* Margar) + (MojitoCoc \* Mojito) + (DaiquiriCoc \* Daiquiri) + (GinTonCoc \* GinTon) + (ManHattanCoc \* Manhatt) + (OldFashCoc \* OldFash) + +(LatteDrink \* Latte) + (GreenTeaDrink \* GreenTea) + (CappucinoDrink \* Cappucino) + (ColdCoffeeDrink \* ColdCoffee);

double cakeCosts = (cofCake \* cCake) + (redValvet \* rValvet) + (chCheeseCk \* cheeseCake);

if (disc\_status)

{

if (discount\_cake)

{

cakeCosts \*= 1 - percentoff;

}

else

{

drinkCosts \*= 1 - percentoff;

}

}

lblDrinkCost.Text = Convert.ToString(drinkCosts);

lblCakeCost.Text = Convert.ToString(cakeCosts);

double svcCharge = Convert.ToDouble(lblSvcCharge.Text);

lblSubTotal.Text = Convert.ToString(cakeCosts + drinkCosts + svcCharge);

lblTax.Text = Convert.ToString(((cakeCosts + drinkCosts + svcCharge) \* tax) / 100);

double totalAftTax = Convert.ToDouble(lblTax.Text);

lblTotal.Text = Convert.ToString(cakeCosts + drinkCosts + svcCharge + totalAftTax);

//element:char

lblDrinkCost.Text = String.Format("{0:C}", drinkCosts);

lblCakeCost.Text = String.Format("{0:C}", cakeCosts);

lblSvcCharge.Text = Convert.ToString(svcCharge);

lblSubTotal.Text = String.Format("{0:C}", (cakeCosts + drinkCosts + svcCharge));

lblTax.Text = String.Format("{0:C}", totalAftTax);

lblTotal.Text = String.Format("{0:C}", (cakeCosts + drinkCosts + svcCharge + totalAftTax));

Reciept();

button3.Focus();

}

else

{

Waiter waiter = new Waiter();

waiter.Show();

log = true;

}

}

}

/// <summary>

/// Button Reset. Refill the Check Area , check the discount, resets the product.text status,resets the checkboxes to defalut statement

/// </summary>

private void button3\_Click(object sender, EventArgs e)

{

rtfReceipt.Clear();

Check\_Discount();

txtTequilaSun.Text = "0";

txtLong.Text = "0";

txtMargarita.Text = "0";

txtMojito.Text = "0";

txtDaiquiri.Text = "0";

txtGinTon.Text = "0";

txtManhattan.Text = "0";

txtOldFash.Text = "0";

//Cake

txtCoffeCake.Text = "0";

txtRedValvetCake.Text = "0";

txtCheeseCake.Text = "0";

txtCappucino.Text = "0";

//Coffe

txtLatte.Text = "0";

txtGreenTea.Text = "0";

txtCappucino.Text = "0";

txtColdCofee.Text = "0";

lblCakeCost.Text = "0";

lblDrinkCost.Text = "0";

lblSvcCharge.Text = "3,21";

//CheckBoxes

chkTequilaSun.Checked = false;

chkLong.Checked = false;

chkMargarita.Checked = false;

chkMojito.Checked = false;

chkDaiquiri.Checked = false;

chkGinTon.Checked = false;

chkManhattan.Checked = false;

chkOldFash.Checked = false;

//

chkRedValvet.Checked = false;

chkCoffe.Checked = false;

chkCheeseCake.Checked = false;

//

chkLatte.Checked = false;

chkGreenTea.Checked = false;

chkCappucino.Checked = false;

chkColdCoffee.Checked = false;

//Expenses

lblTax.Text = "0";

lblSubTotal.Text = "0";

lblTotal.Text = "0";

}

/// <summary>

/// Exit

/// </summary>

private void btnExit\_Click(object sender, EventArgs e)

{

Application.Exit();

double latteeCof = Convert.ToDouble(txtTequilaSun.Text);

}

/// <summary>

/// Price List Printing

/// </summary>

private void button4\_Click(object sender, EventArgs e)

{

toolStrip1.Enabled = false;

PrintPrice\_List();

}

/// <summary>

/// Log In button

/// </summary>

private void Waiter\_Click(object sender, EventArgs e)

{

Waiter waiter = new Waiter();

waiter.Show();

log = true;

}

#endregion

#region Functions

/// <summary>

/// Output of orders

/// </summary>

public void Reciept()

{

toolStrip1.Enabled = true;

//btnReceipt

rtfReceipt.Clear();

PrintReceipt();

}

/// <summary>

/// Printin the Total

/// </summary>

private void PrintReceipt()

{

// rtfReceipt.AppendText(Environment.NewLine);

rtfReceipt.AppendText("-------------------------------------------------------------" + Environment.NewLine);

rtfReceipt.AppendText("\t" + " AROMAT Restaurant" + Environment.NewLine);

rtfReceipt.AppendText("-------------------------------------------------------------" + Environment.NewLine);

rtfReceipt.AppendText("Tequila Sunrise \t\t" + txtTequilaSun.Text + Environment.NewLine);

rtfReceipt.AppendText("Long Island Ice Tea \t\t" + txtLong.Text + Environment.NewLine);

rtfReceipt.AppendText("Margarita \t\t\t" + txtMargarita.Text + Environment.NewLine);

rtfReceipt.AppendText("Mojito \t\t\t\t" + txtMojito.Text + Environment.NewLine);

rtfReceipt.AppendText("Daiquiri \t\t\t" + txtDaiquiri.Text + Environment.NewLine);

rtfReceipt.AppendText("Gin and Tonic\t\t\t" + txtGinTon.Text + Environment.NewLine);

rtfReceipt.AppendText("Manhatan \t\t\t" + txtManhattan.Text + Environment.NewLine);

rtfReceipt.AppendText("Old Fashioned Cocktail\t\t" + txtOldFash.Text + Environment.NewLine);

rtfReceipt.AppendText("Coffee Cake \t\t\t" + txtCoffeCake.Text + Environment.NewLine);

rtfReceipt.AppendText("Red Valvet Cake \t\t" + txtRedValvetCake.Text + Environment.NewLine);

rtfReceipt.AppendText("Cheese Cake \t\t\t" + txtCheeseCake.Text + Environment.NewLine);

rtfReceipt.AppendText("Latte \t\t\t\t" + txtLatte.Text + Environment.NewLine);

rtfReceipt.AppendText("Grean Tea\t\t\t" + txtGreenTea.Text + Environment.NewLine);

rtfReceipt.AppendText("Cappucino\t\t\t" + txtCappucino.Text + Environment.NewLine);

rtfReceipt.AppendText("Cold Coffee\t\t\t" + txtColdCofee.Text + Environment.NewLine);

rtfReceipt.AppendText("-------------------------------------------------------------" + Environment.NewLine);

rtfReceipt.AppendText("Service Charge \t\t" + lblSvcCharge.Text + Environment.NewLine);

rtfReceipt.AppendText("-------------------------------------------------------------" + Environment.NewLine);

rtfReceipt.AppendText("Tax \t\t\t\t" + lblTax.Text + Environment.NewLine);

rtfReceipt.AppendText("Sub Total \t\t\t" + lblSubTotal.Text + Environment.NewLine);

rtfReceipt.AppendText("Total Cost \t\t\t" + lblTotal.Text + Environment.NewLine);

rtfReceipt.AppendText("-------------------------------------------------------------" + Environment.NewLine);

rtfReceipt.AppendText(lblTimer.Text + "\t" + lblDate.Text + Environment.NewLine);

rtfReceipt.AppendText("Served by " + persons);

}

/// <summary>

/// Logical expression

/// </summary>

private bool IsAllEmpty()

{

if (!chkDaiquiri.Checked & !chkTequilaSun.Checked & !chkLong.Checked & !chkOldFash.Checked & !chkMargarita.Checked & !chkManhattan.Checked & !chkCoffe.Checked & !chkMojito.Checked & !chkGinTon.Checked & !chkCoffe.Checked & !chkRedValvet.Checked & !chkCheeseCake.Checked & !chkLatte.Checked & !chkGreenTea.Checked & !chkCappucino.Checked & !chkColdCoffee.Checked)

{

return true;

}

else

{

return false;

}

}

/// <summary>

/// Set the default price

/// </summary>

private void SetVar()

{

//Alcohol

txtTequilaSun.Text = "0";

txtLong.Text = "0";

txtMargarita.Text = "0";

txtMojito.Text = "0";

txtDaiquiri.Text = "0";

txtGinTon.Text = "0";

txtManhattan.Text = "0";

txtOldFash.Text = "0";

txtCoffeCake.Text = "0";

txtRedValvetCake.Text = "0";

txtCheeseCake.Text = "0";

//Drinks - Alcohol prises

Teq = 22.20; Long = 22.29; Margar = 17.29; Mojito = 13.45;

Daiquiri = 13.70; GinTon = 18.50; Manhatt = 22.13; OldFash = 22.20;

//Cofee

txtLatte.Text = "0";

txtGreenTea.Text = "0";

txtCappucino.Text = "0";

txtColdCofee.Text = "0";

//Drinkgs - Coffee prises

Latte = 11.50; GreenTea = 8.40; Cappucino = 11.30; ColdCoffee = 12.10;

//Cakes

lblCakeCost.Text = "0";

lblDrinkCost.Text = "0";

lblSvcCharge.Text = "3,21";

//Cakes - prises

cCake = 10.99; rValvet = 12.99; cheeseCake = 11.99;

lblSubTotal.Text = "0";

lblTax.Text = "0";

lblTotal.Text = "0";

}

/// <summary>

/// Controls setter

/// </summary>

private void Enabling()

{

txtTequilaSun.Enabled = false;

txtLong.Enabled = false; ;

txtMargarita.Enabled = false;

txtMojito.Enabled = false;

txtDaiquiri.Enabled = false;

txtGinTon.Enabled = false;

txtManhattan.Enabled = false;

txtOldFash.Enabled = false;

txtCoffeCake.Enabled = false;

txtRedValvetCake.Enabled = false;

txtCheeseCake.Enabled = false;

txtLatte.Enabled = false;

txtGreenTea.Enabled = false;

txtCappucino.Enabled = false;

txtColdCofee.Enabled = false;

chkTequilaSun.Checked = false;

chkLong.Checked = false;

chkMargarita.Checked = false;

chkMojito.Checked = false;

chkDaiquiri.Checked = false;

chkGinTon.Checked = false;

chkManhattan.Checked = false;

chkOldFash.Checked = false;

chkRedValvet.Checked = false;

chkCoffe.Checked = false;

chkCheeseCake.Checked = false;

chkLatte.Checked = false;

chkGreenTea.Checked = false;

chkCappucino.Checked = false;

chkColdCoffee.Checked = false;

}

/// <summary>

/// PrintPrice in rtfReceipt control

/// </summary>

private void PrintPrice\_List()

{

rtfReceipt.Clear();

rtfReceipt.AppendText("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" + Environment.NewLine);

rtfReceipt.AppendText("\t" + " AROMAT Restaurant" + Environment.NewLine);

rtfReceipt.AppendText("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" + Environment.NewLine);

rtfReceipt.AppendText("Tequila Sunrise \t\t" + Teq + Environment.NewLine);

rtfReceipt.AppendText("Long Island Ice Tea \t\t" + Long + Environment.NewLine);

rtfReceipt.AppendText("Margarita \t\t\t" + Margar + Environment.NewLine);

rtfReceipt.AppendText("Mojito \t\t\t\t" + Mojito + Environment.NewLine);

rtfReceipt.AppendText("Daiquiri \t\t\t" + Daiquiri + Environment.NewLine);

rtfReceipt.AppendText("Gin and Tonic\t\t\t" + GinTon + Environment.NewLine);

rtfReceipt.AppendText("Manhatan \t\t\t" + Manhatt + Environment.NewLine);

rtfReceipt.AppendText("Old Fashioned Cocktail\t\t" + OldFash + Environment.NewLine);

rtfReceipt.AppendText("Coffee Cake \t\t\t" + cCake + Environment.NewLine);

rtfReceipt.AppendText("Red Valvet Cake \t\t" + rValvet + Environment.NewLine);

rtfReceipt.AppendText("Cheese Cake \t\t\t" + cheeseCake + Environment.NewLine);

rtfReceipt.AppendText("Latte \t\t\t\t" + Latte + Environment.NewLine);

rtfReceipt.AppendText("Grean Tea\t\t\t" + GreenTea + Environment.NewLine);

rtfReceipt.AppendText("Cappucino\t\t\t" + Cappucino + Environment.NewLine);

rtfReceipt.AppendText("Cold Coffee\t\t\t" + ColdCoffee + Environment.NewLine);

rtfReceipt.AppendText("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" + Environment.NewLine);

}

/// <summary>

/// Focus the button after total click

/// </summary>

private void txtEspresso\_TextChanged(object sender, EventArgs e)

{

button1.Focus();

}

#endregion

**Waiter Form**

public partial class Waiter : Form

{

public string name;

Form1 form = new Form1();

public Waiter()

{

InitializeComponent();

BtnClose.Focus();

}

private void timer1\_Tick(object sender, EventArgs e)

{

lblTimer.Text = DateTime.Now.ToLongTimeString();

}

private void Waiter\_Load(object sender, EventArgs e)

{

lbDate.Text = DateTime.Now.ToShortDateString();

timer1.Start();

}

private void BtnClose\_Click(object sender, EventArgs e)

{

if (lbWaiterName.Text == "Your Name")

{

MessageBox.Show("You should fill the blank");

}

else

{

Form1.persons = lbWaiterName.Text;

this.Close();

}

}

private void lbWaiterName\_KeyDown(object sender, KeyEventArgs e)

{

if (e.KeyCode == Keys.Enter)

{

BtnClose.PerformClick();

}

}

}