

**MS806 – Business Application Programming**

**Student Name: Varun Dua**

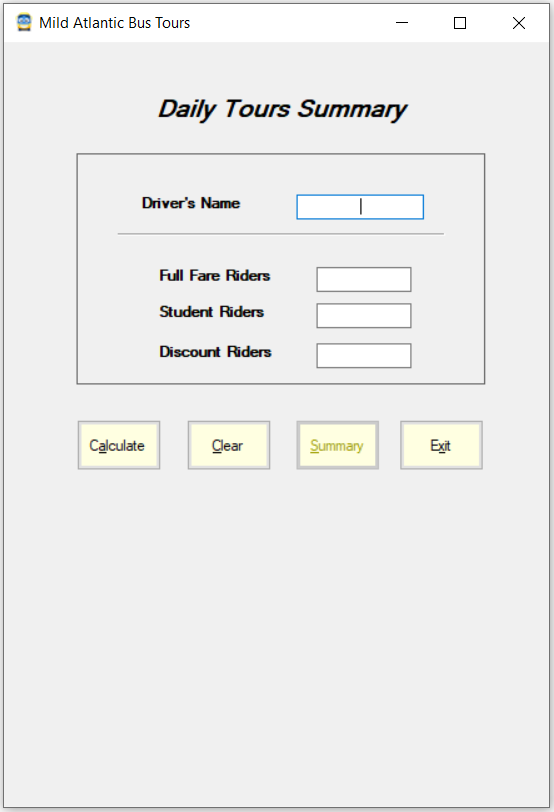
**NUI Galway ID Number: 19230587**

**Assignment:** 2

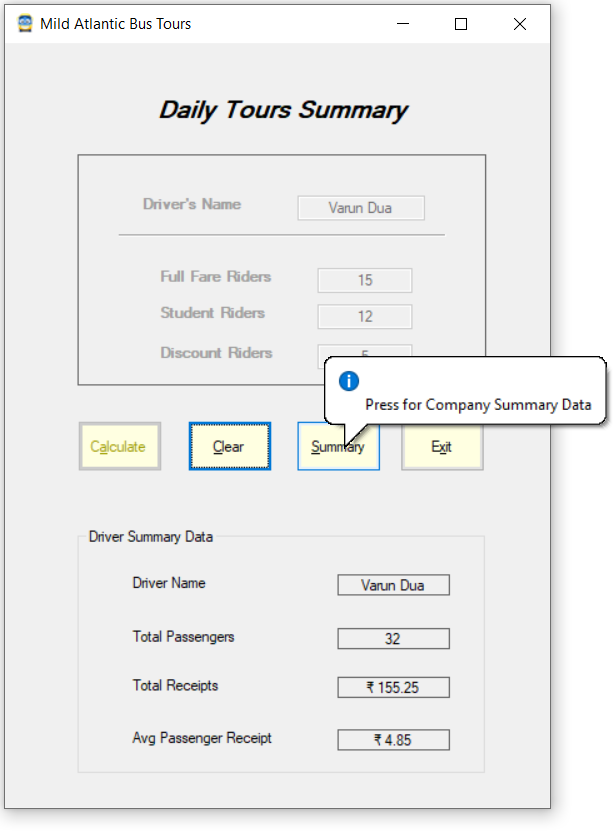
**Date of Submission:** 07/10/2019

**Assignment 2: Mild Atlantic Bus Tours**

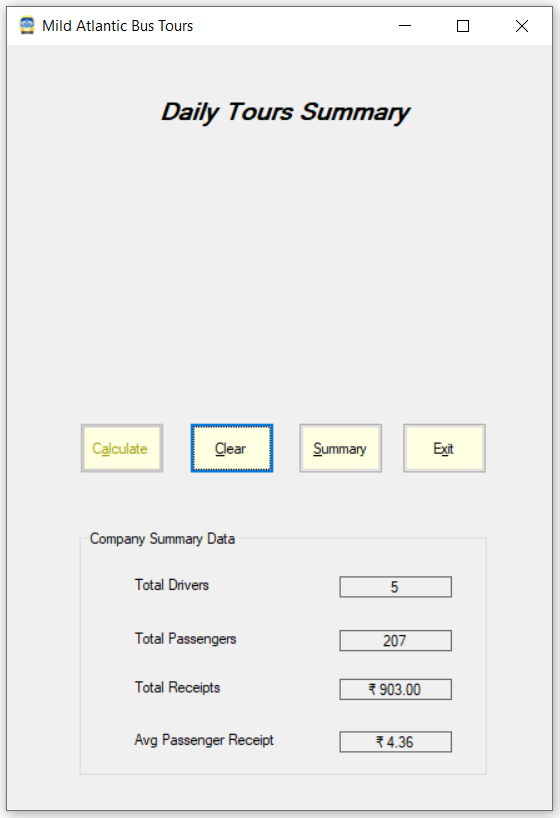
**Screen shots of Application at Runtime:**



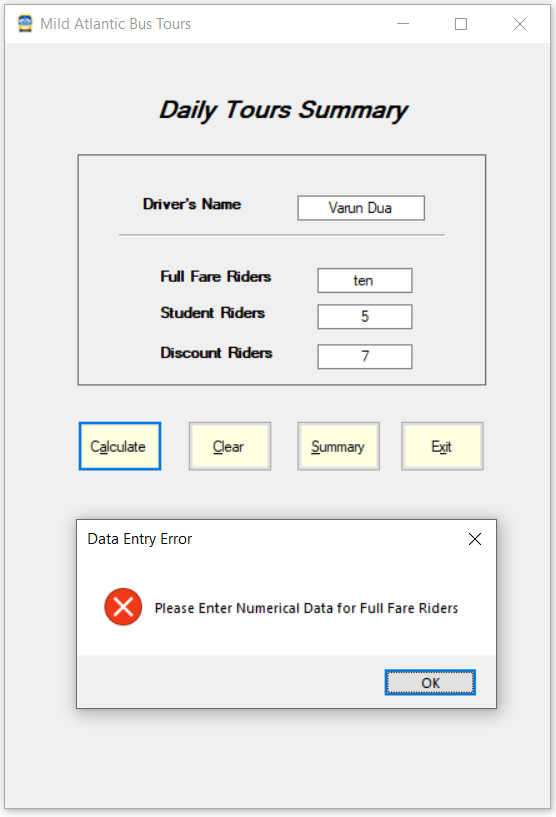
**Fig 1: On Form Load**



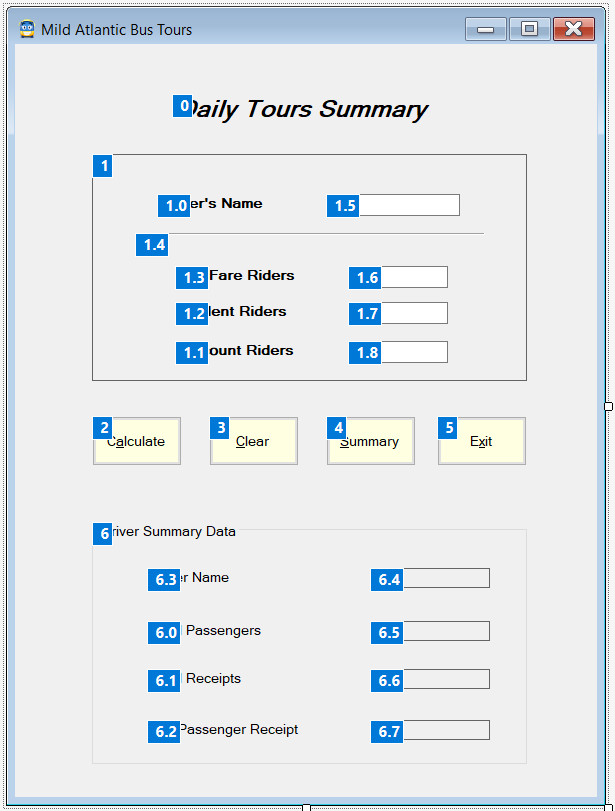
**Fig 2: After Calculate button is clicked**



**Fig 3: After Company Summary Button is clicked**



**Fig 4: Error Handling**



**Fig 4: Application Tab Order**

**Source Code:**

/\* Student Name: Varun Dua

\* Student ID: 19230587

\* Date: 07/10/2019

\* Assignment: 2

\* Assignment: Create a well-designed application that allows bus drivers for Mild Atlantic Bus Tours to enter their

\* name, the number of discounted, student and full-fare passengers that they have carried on their tours

\* that day. The fares for Mild Atlantic Bus Tours are €5.50 for a full-fare passenger, €4.50 for a student

\* and €3.75 for a discount fare passenger.

\* Calculate data for a driver including total riders, total receipts and average receipts per rider and

\* display this data, along with the driver’s name in a group box. Calculate summary data for the company

\* including total number of drivers, total number of riders,

\* total receipts and average receipt per rider and display this data in a group box.

\*/

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;

namespace Assignment2

{

public partial class BusForm : Form

{

/\*Initialize Form\*/

public BusForm()

{

InitializeComponent();

}

const decimal FULLFARE = 5.5m; //Full Fare Charges

const decimal STUDENTFARE = 4.5m; //Student Charges

const decimal DISCOUNTFARE = 3.75m; //Discounted Charges

int NumberOfDriver = 0; //To keep tabs on number of drivers

decimal TotalFare = 0.0m;

int TotalPassengers = 0;

int NegativeExceptionFlag = 0; //To catch Negative values in Text boxes

/\*Handle Form load event. Summary button is kept disabled.\*/

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

{

SummaryButton.Enabled = false;

DataGroup.Visible = false;

}

/\*Event handler for clicking on Calculate button. Driver summary is calculated, and company summary elements updated\*/

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

{

int FullFarePassengers = 0;

int StudentPassengers = 0;

int DiscountPassengers = 0;

try //STEP 1 Parse Full Fare Passengers

{

FullFarePassengers = int.Parse(FullFareBox.Text);

if (FullFarePassengers < 0)

{

NegativeExceptionFlag = 1;

throw new Exception();

}

try //STEP 2 Parse Students Passengers

{

StudentPassengers = int.Parse(StudentBox.Text);

if (StudentPassengers < 0)

{

NegativeExceptionFlag = 1;

throw new Exception();

}

try //STEP 3 Parse Discounted Passengers

{

DiscountPassengers = int.Parse(DiscountBox.Text);

if (DiscountPassengers < 0)

{

NegativeExceptionFlag = 1;

throw new Exception();

}

DriverData.Text = "Driver Name";

DriverDataLabel.Text = DriverBox.Text.ToString(); ;

TotalPassengerLabel.Text = (FullFarePassengers + StudentPassengers + DiscountPassengers).ToString();

TotalReceiptLabel.Text = ((FullFarePassengers \* FULLFARE) + (StudentPassengers \* STUDENTFARE) + (DiscountPassengers \* DISCOUNTFARE)).ToString("C");

/\* Handling situation where number of riders in each category is entered 0, and avoid divide by zero exception \*/

if((FullFarePassengers + StudentPassengers + DiscountPassengers) !=0)

{

AverageLabel.Text = (((FullFarePassengers \* FULLFARE) + (StudentPassengers \* STUDENTFARE) + (DiscountPassengers \* DISCOUNTFARE)) / (FullFarePassengers + StudentPassengers + DiscountPassengers)).ToString("C");

}

else

{

AverageLabel.Text = "0";

}

DataGroup.Text = "Driver Summary Data"; //Name of the group box updated for Driver Summary

DataGroup.Visible = true;

CalculateButton.Enabled = false;

ClearButton.Focus();

SummaryButton.Enabled = true;

InfoPanel.Enabled = false;

/\*Updating data for Company Summary\*/

TotalPassengers += FullFarePassengers + StudentPassengers + DiscountPassengers;

TotalFare += ((FullFarePassengers \* FULLFARE) + (StudentPassengers \* STUDENTFARE) + (DiscountPassengers \* DISCOUNTFARE));

NumberOfDriver++;

}

catch

{

if (NegativeExceptionFlag == 0)

{

MessageBox.Show("Please Enter Numerical Data for Discount Fare Riders", "Data Entry Error", MessageBoxButtons.OK, MessageBoxIcon.Stop);

DiscountBox.Focus();

DiscountBox.SelectAll();

}

else

{

MessageBox.Show("Please Enter Non Negative Data for Discount Fare Riders", "Data Entry Error", MessageBoxButtons.OK, MessageBoxIcon.Stop);

DiscountBox.Focus();

DiscountBox.SelectAll();

NegativeExceptionFlag = 0;

}

}

}

catch

{

if (NegativeExceptionFlag == 0)

{

MessageBox.Show("Please Enter Numerical Data for Student Fare Riders", "Data Entry Error", MessageBoxButtons.OK, MessageBoxIcon.Stop);

StudentBox.Focus();

StudentBox.SelectAll();

}

else

{

MessageBox.Show("Please Enter Non Negative Data for Student Fare Riders", "Data Entry Error", MessageBoxButtons.OK, MessageBoxIcon.Stop);

StudentBox.Focus();

StudentBox.SelectAll();

NegativeExceptionFlag = 0;

}

}

}

catch

{

if(NegativeExceptionFlag == 0)

{

MessageBox.Show("Please Enter Numerical Data for Full Fare Riders", "Data Entry Error", MessageBoxButtons.OK, MessageBoxIcon.Stop);

FullFareBox.Focus();

FullFareBox.SelectAll();

}

else

{

MessageBox.Show("Please Enter Non Negative Data for Full Fare Riders", "Data Entry Error", MessageBoxButtons.OK, MessageBoxIcon.Stop);

FullFareBox.Focus();

FullFareBox.SelectAll();

NegativeExceptionFlag = 0;

}

}

}

/\*Event handler on clicking Clear button.\*/

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

{

DataGroup.Visible = false;

InfoPanel.Visible = true;

InfoPanel.Enabled = true;

CalculateButton.Enabled = true;

DriverBox.Text = "";

FullFareBox.Text = "";

StudentBox.Text = "";

DiscountBox.Text = "";

DriverBox.Focus();

}

/\*Event handler for Summary button Click. Company Summary is displayed and Calculate button is disabled\*/

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

{

DataGroup.Text = "Company Summary Data"; //Name of the Group box is updated for Company Summary

DriverData.Text = "Total Drivers";

DriverDataLabel.Text = NumberOfDriver.ToString();

TotalPassengerLabel.Text = TotalPassengers.ToString();

TotalReceiptLabel.Text = TotalFare.ToString("C");

/\* Handling situation where number of riders in each category is entered 0, and avoid divide by zero exception \*/

if (TotalPassengers !=0)

{

AverageLabel.Text = (TotalFare/TotalPassengers).ToString("C");

}

else

{

AverageLabel.Text = "0";

}

InfoPanel.Visible = false;

DataGroup.Visible = true;

CalculateButton.Enabled = false;

ClearButton.Focus();

}

/\*Event handler for Exit button Click.\*/

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

{

this.Close();

}

}

}