Informatics College Pokhara

Application Development (CS6004NP)

Title: Visitor Records Management System

Submitted By:

Name: Netra Bahadur Rana

London Met ID: 19031004

Group: C4

Submission Date: 03 Jan, 2022

Submitted To:

Mr. Sabin Shrestha

Application Development

Contents

List of Figures

1 Introduction

"Sam's place" is a arts centre providing musical instruments training and arts for a decade. Apart from that it also provides other recreational facilities. It attracts a lot of visitor's attention especially because all age groups are allowed and it opens on all days. But now, because of the large number of visitors, it has been difficult for them to keep records manually.

In this project, we are tasked to create a C# desktop application that helps to manage records of tickets and visitors. The system must be able to record data on all days of a week (from Sunday to Saturday). Recreation centre has different ticket options based on age (child and adult), based on days (holidays and weekdays), and based on group (Group of 5, Group of 10) and time duration (1hr, 2hr, ….., unlimited). The application must be designed around these aspects.

In comparison, doing things like records manually is much easier in an automated environment than doing it manually. The application must be able to conduct all the things that required to be done manually, smoothly and provide a better efficiency for the recreation centre.

In this document, we will be going through all the process of the development of the system until its testing.

2 System Architecture Diagram

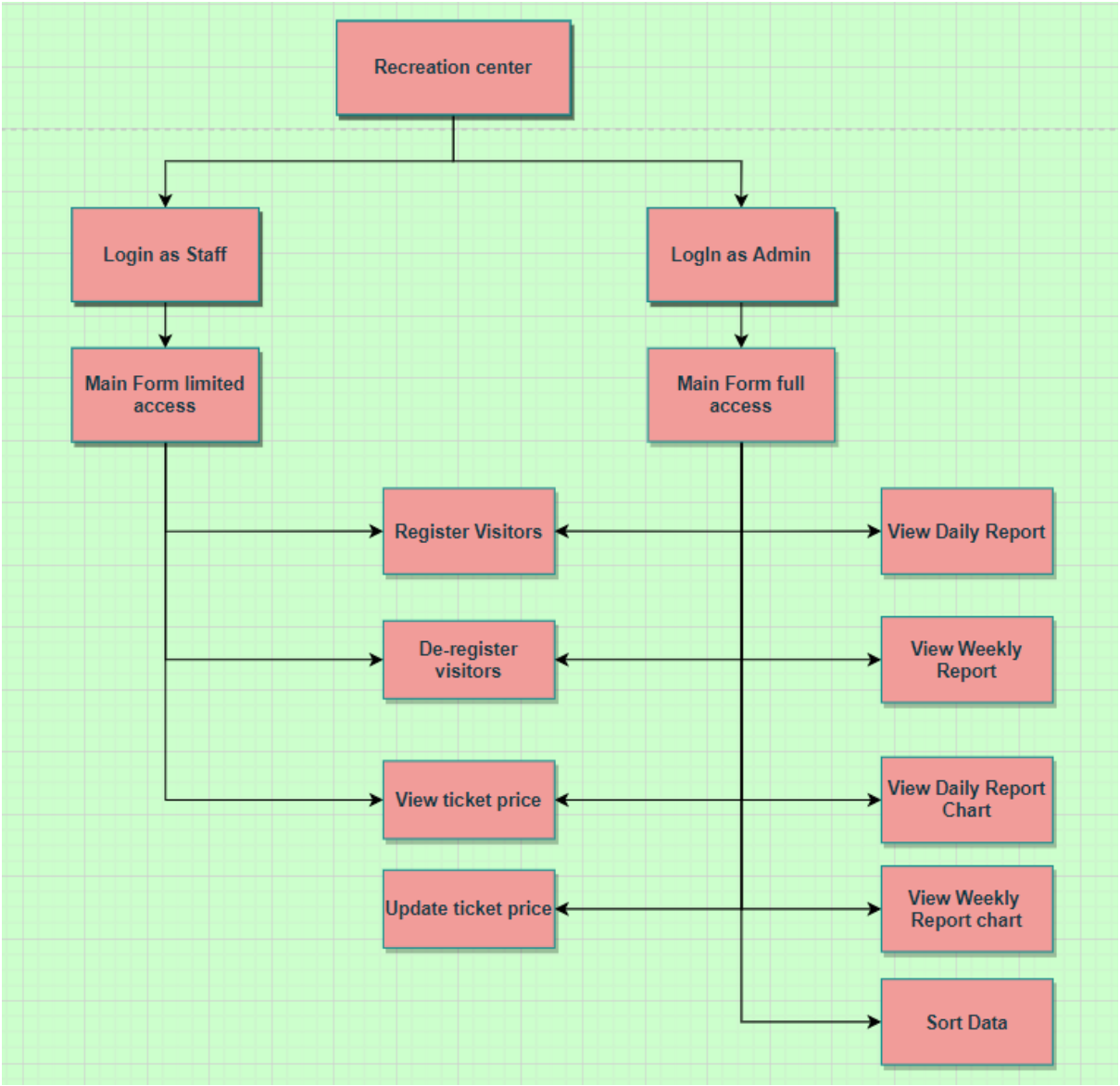


Figure 1: System Architecture diagram

3 Use case diagram

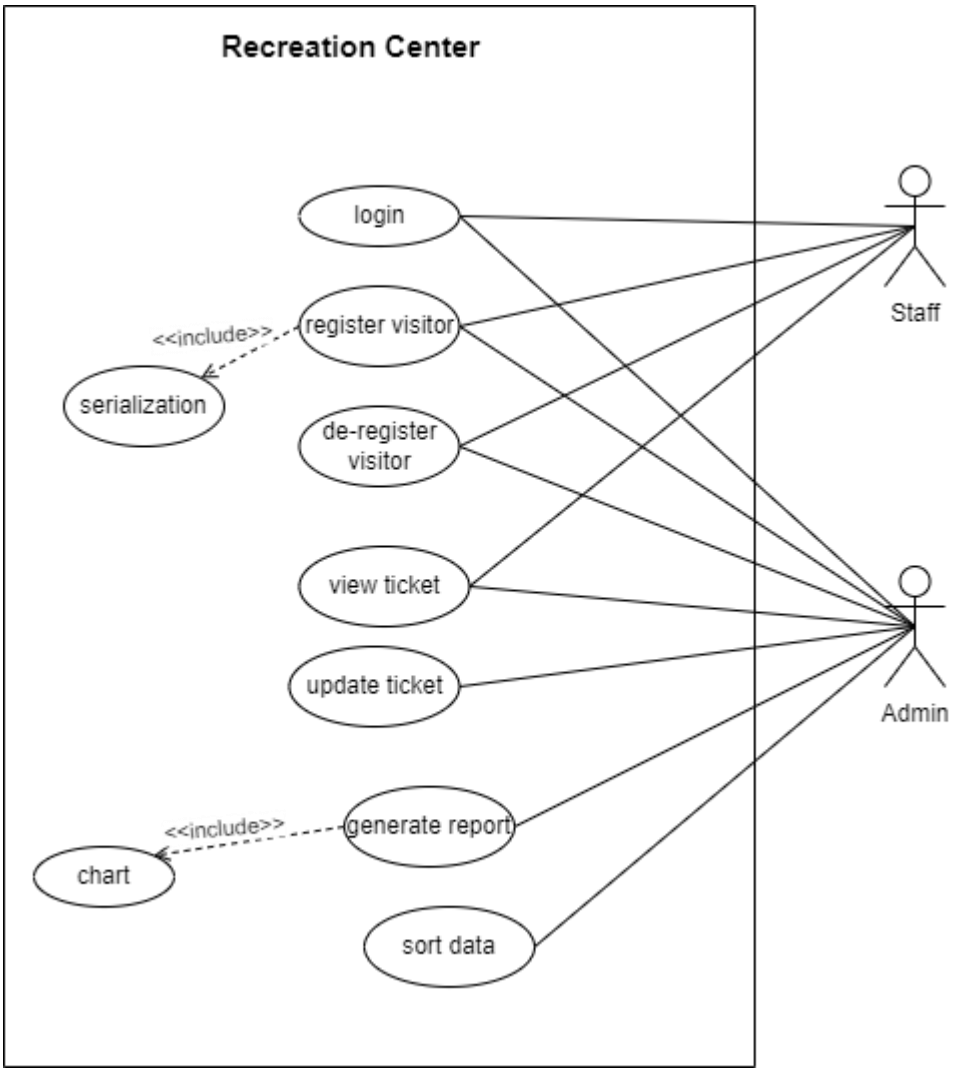


Figure 2: Use Case Diagram

4 User Manual

In this section, we are going to be looking at how to approach the system.

1. First shows up login form. We have two options: Login as **Admin** or **Staff**.

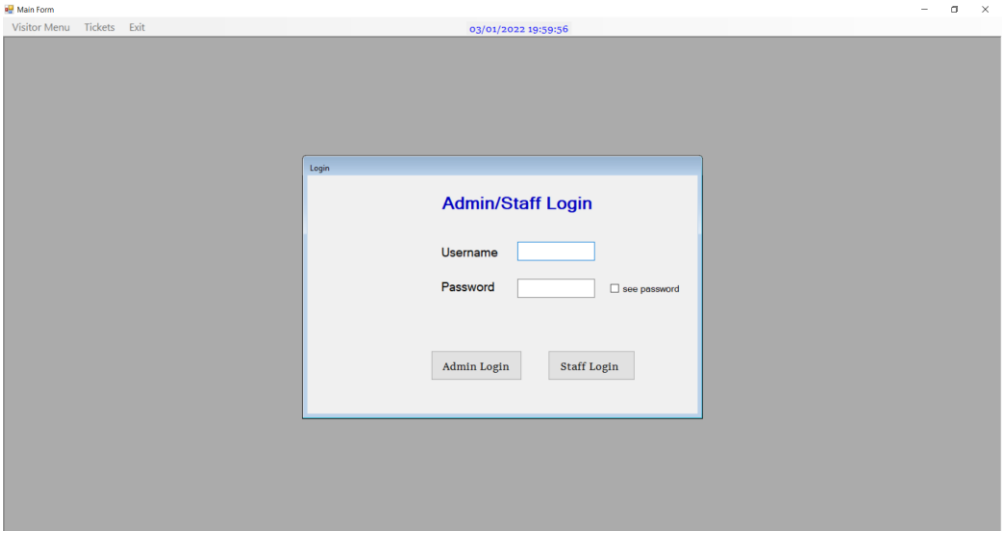


Figure 3: Login form

2. We cannot access the menu items without logging in.

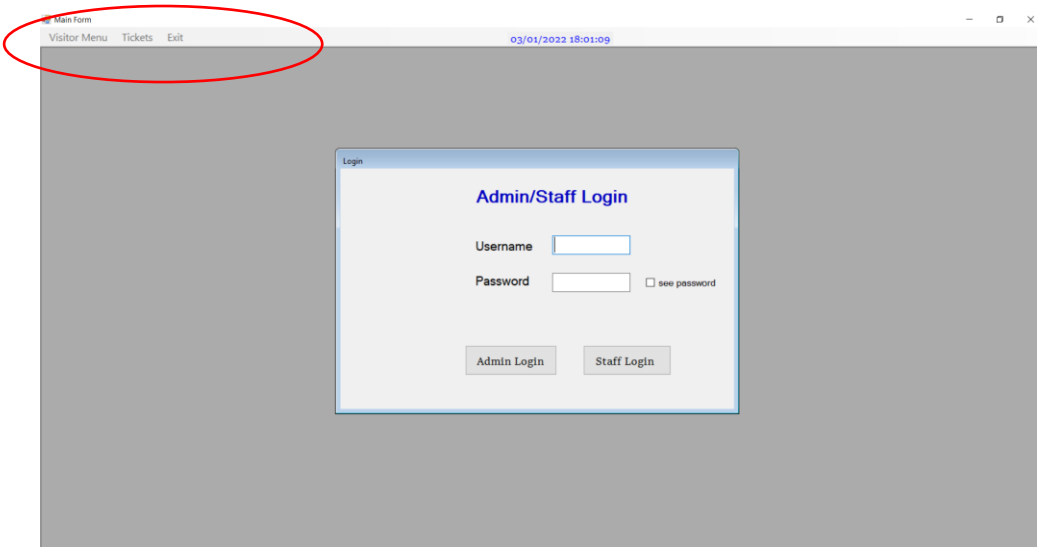


Figure 4: Inaccessible menu items

3. To login as Admin, use **Admin** as both **Username** and **Password**.

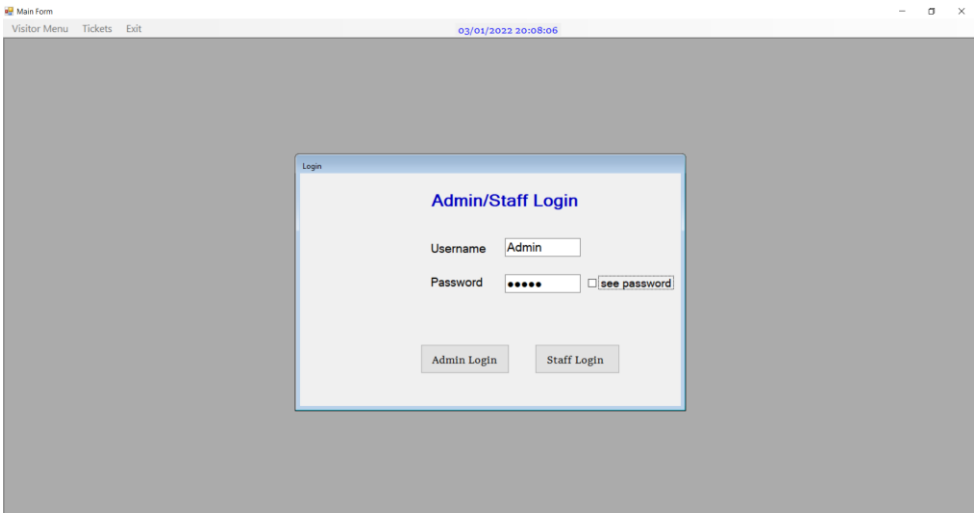


Figure 5: Admin login with hidden passwords

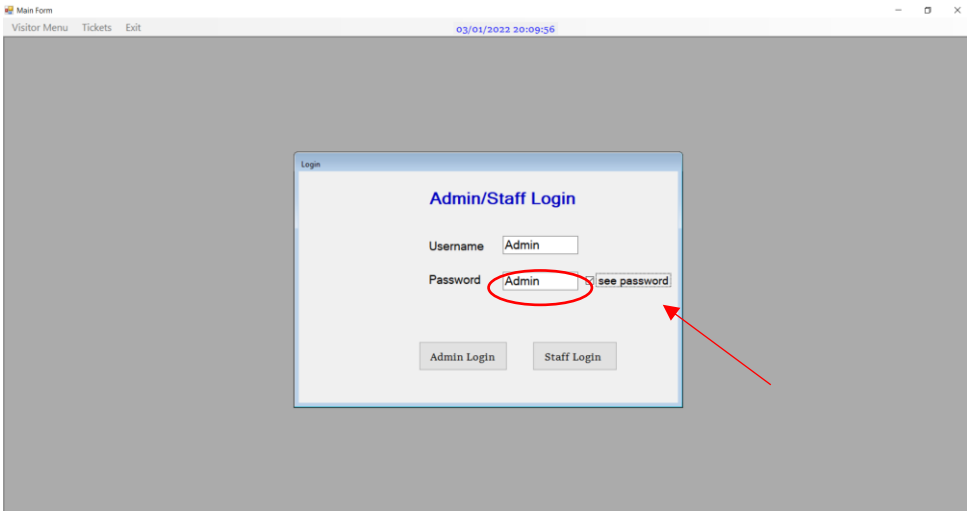


Figure 6: Admin login without hiding passwords

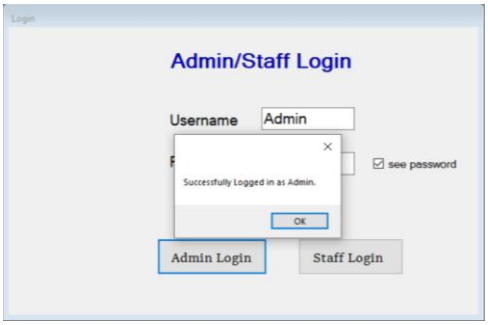


Figure 7: Notification on successful Admin Login

Similarly, to login as Staff, use **Staff** as both **Username** and **Password**.

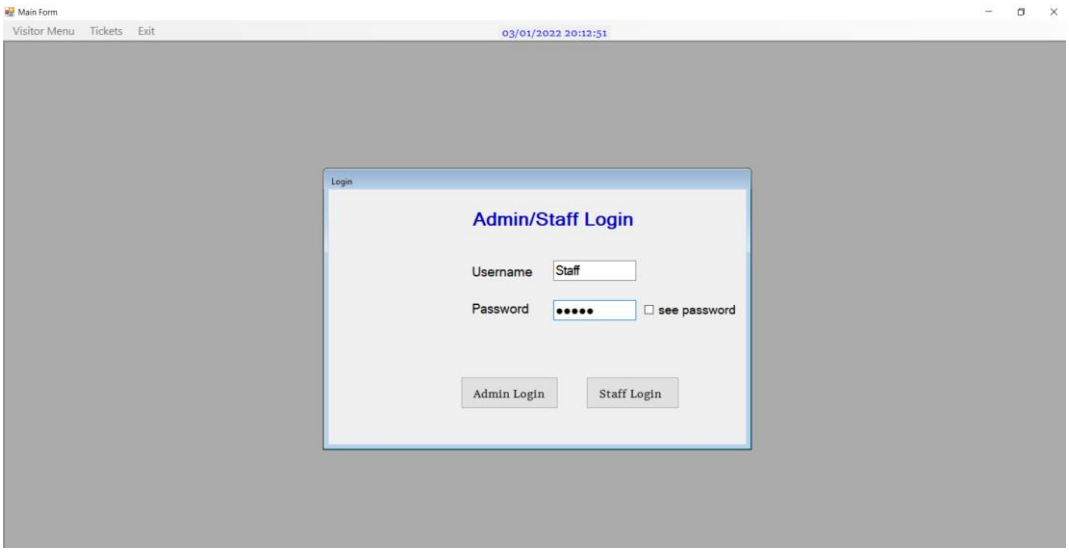


Figure 8: Staff login with hidden passwords

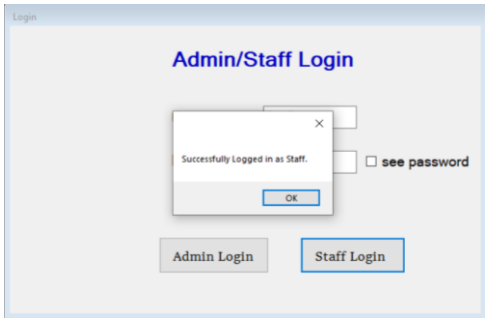
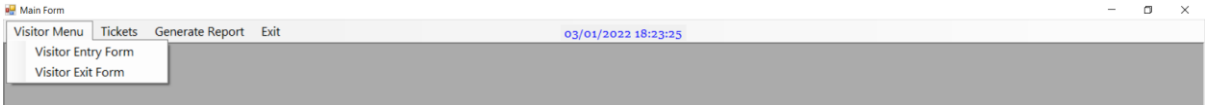


Figure 9: Notification on successful Staff login

4. Accessible menu items to Admin.

Accessible menu items to Staff.

5. Visitor menu sub-items.



6. Visitor Entry Form User Interface

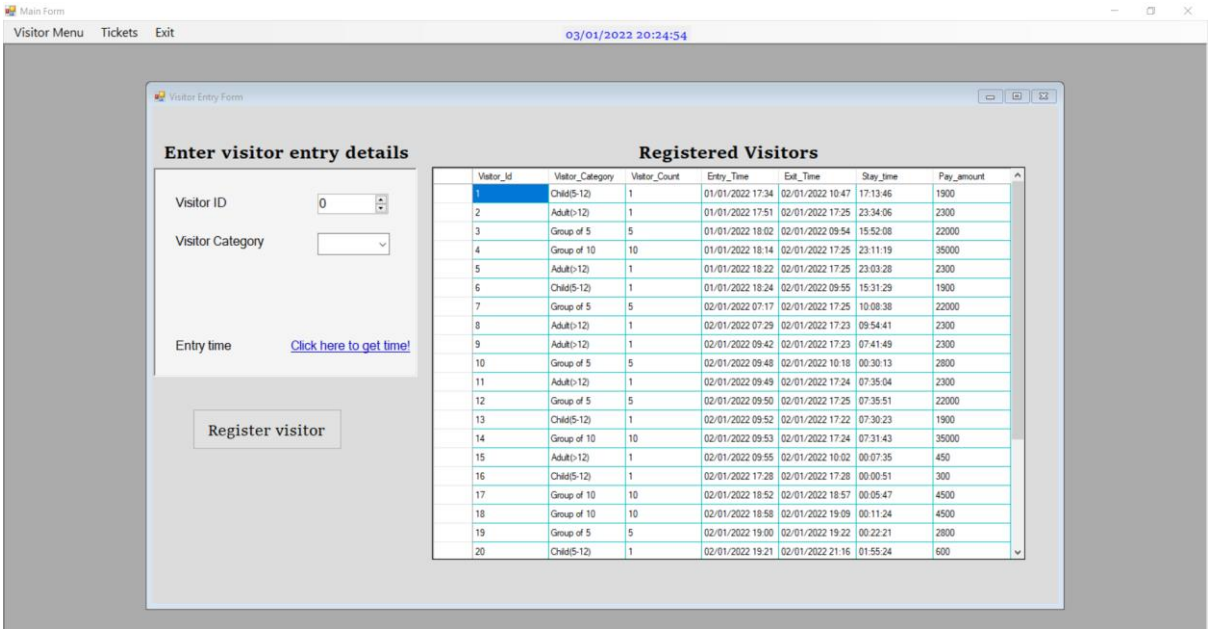


Figure 10: Visitor Entry Form UI

Visitor Exit Form User Interface

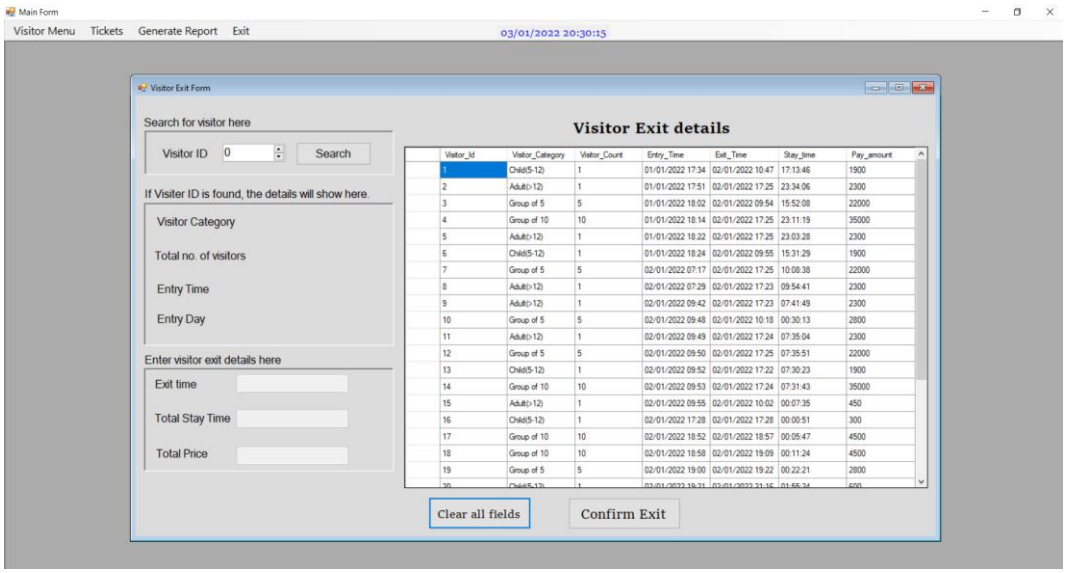


Figure 11: Visitor Exit Form UI

7. Tickets form. In the tickets form, only the admin can update and save prices. Staff won't be able to update prices.

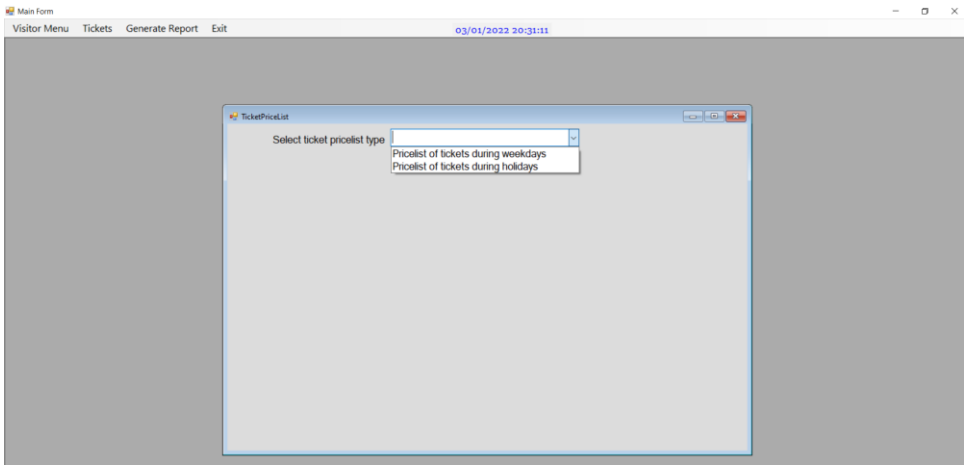


Figure 12: Tickets UI

8. Tickets form opened by Admin.

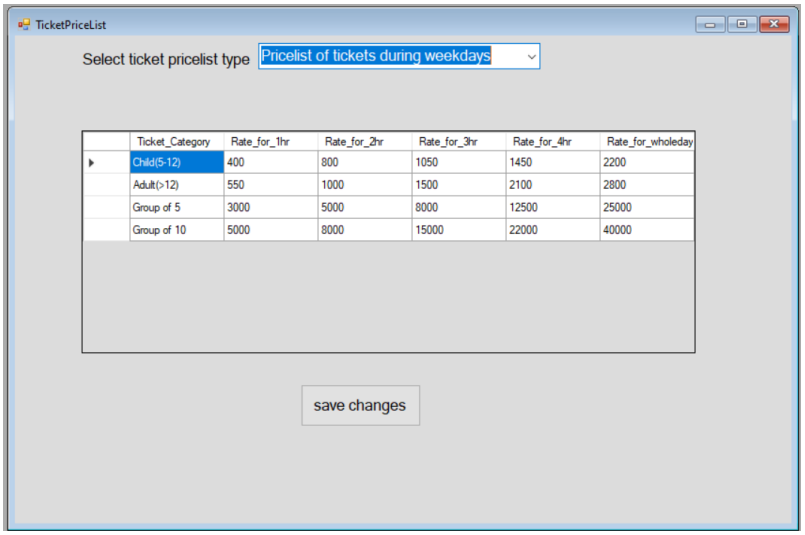


Figure 13: Tickets price list during weekdays

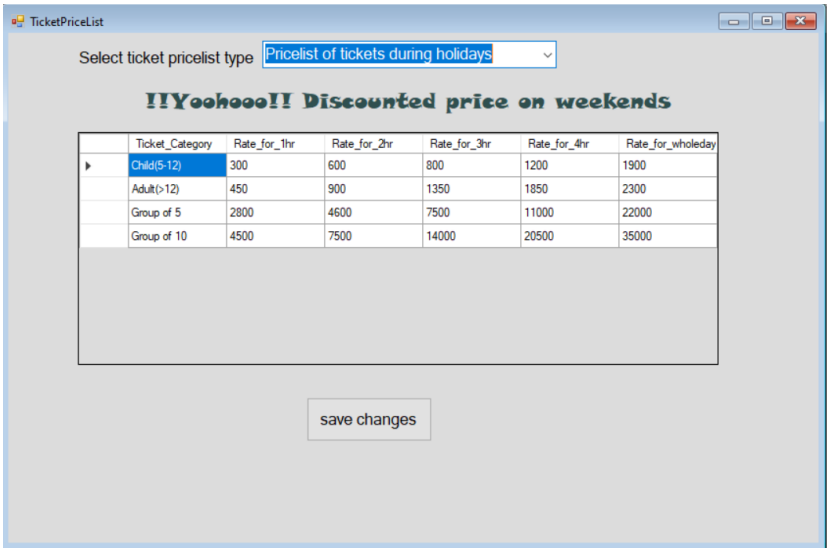


Figure 14: Tickets price list during weekends

9. Tickets form viewed by Staff (**No save button available**).

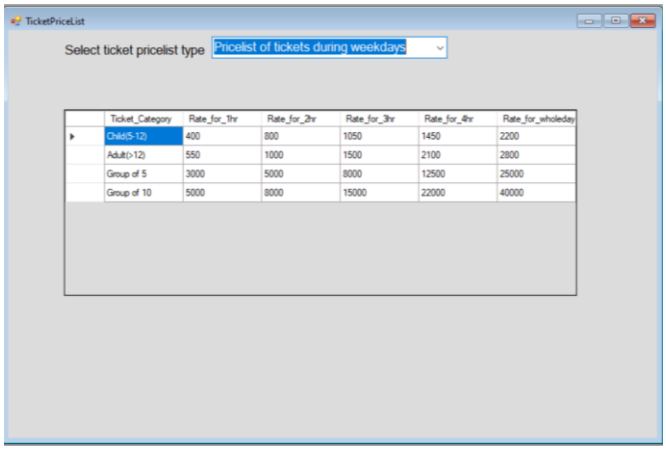


Figure 15: Tickets form viewed by Staff

10. Generate Report sub-menu. **Only Admin** can access it**.**

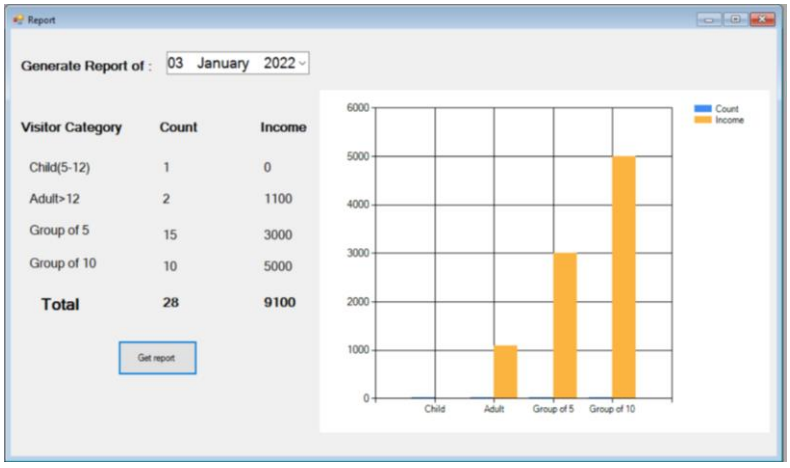


Figure 16: Daily Report UI

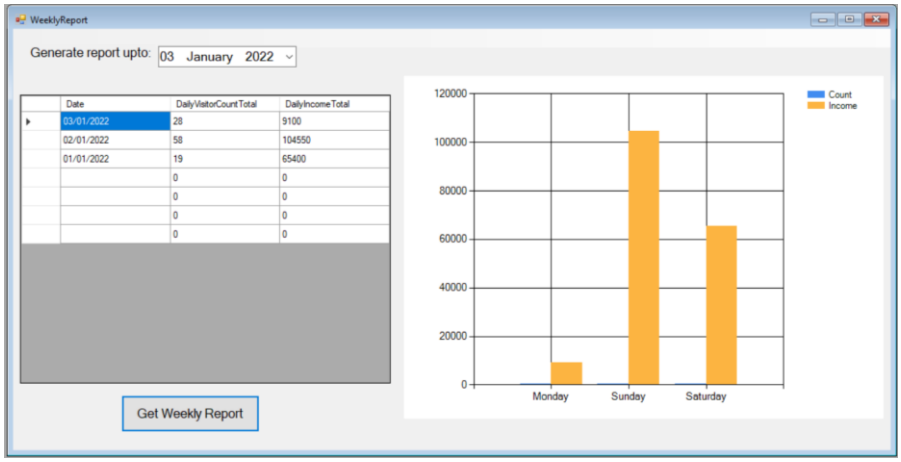


Figure 17: Weekly Report UI

In **fig: 17**, there are only 3 data even though it is a weekly report. The reason for it is we only have 3 dates in our system. That is why there are no other dates shown. If other dates were available, it would have appeared in the chart.

11. Exit sub-menu.

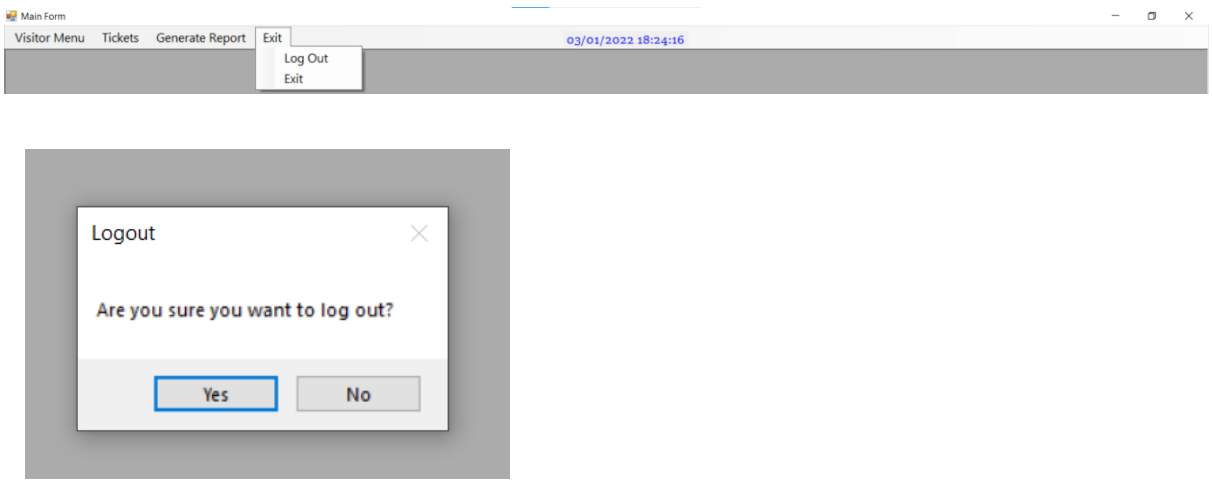


Figure 18: Logout dialog box

If "**Yes**" is clicked, login form will reopen. If "**No**" is selected, then the dialog box disappears.

Clicking exit will entirely close the system.

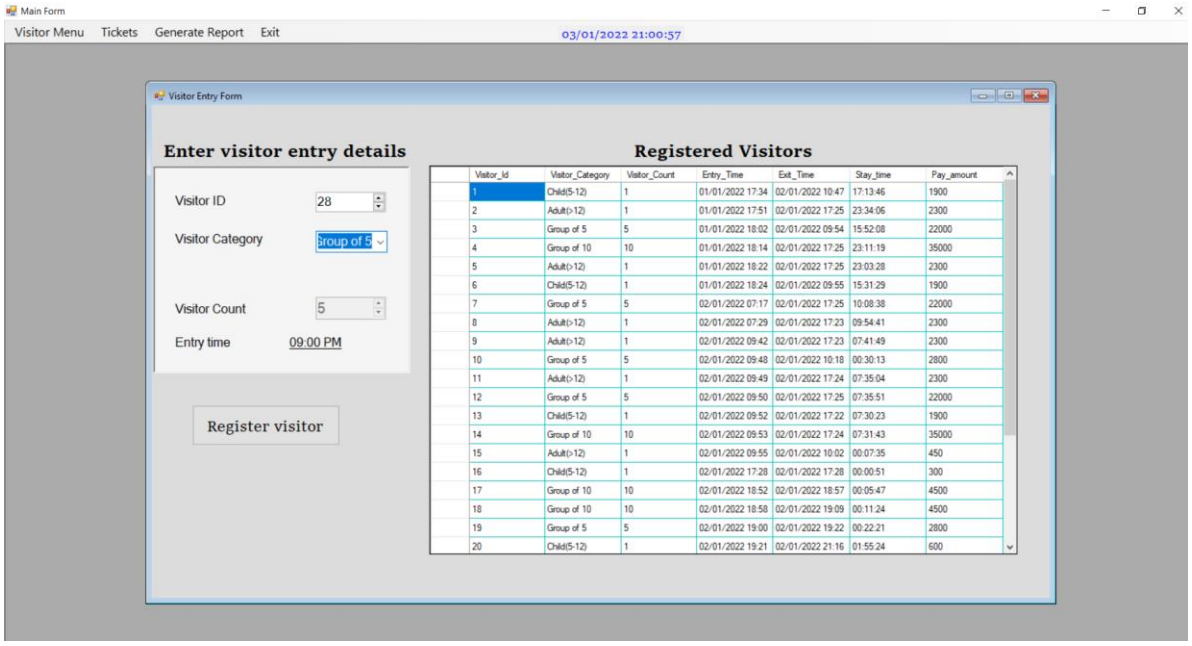


Figure 19: Fully functioning Entry Form

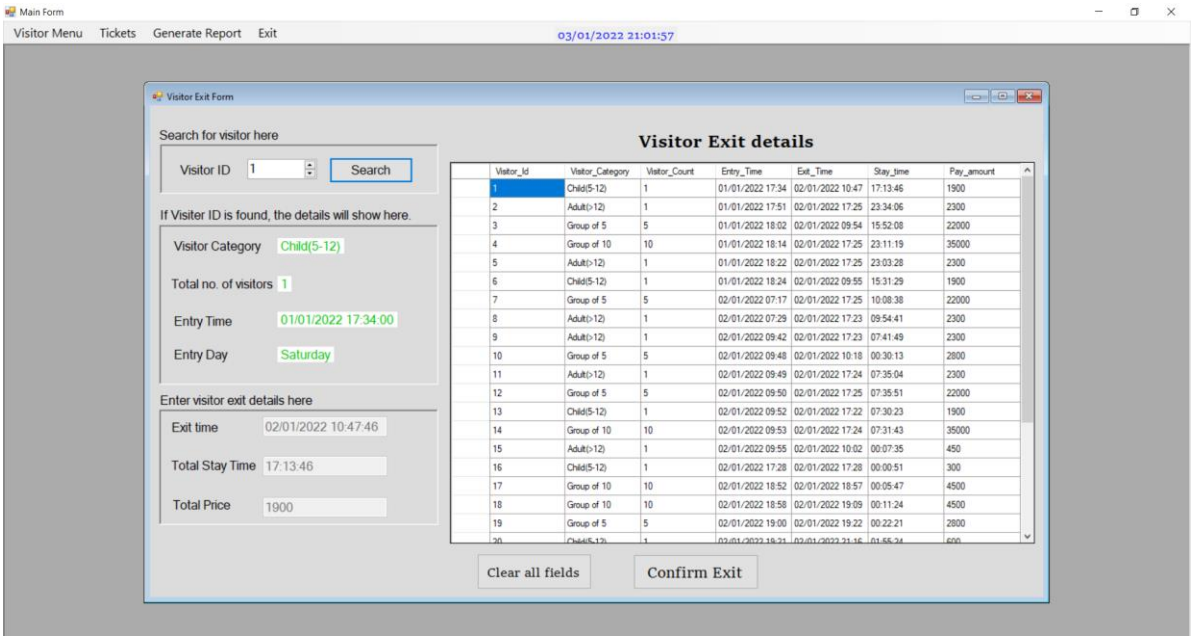


Figure 20: Fully functioning Exit form

5 System Validation

Validating is important in a system because it prevents wrong or incorrect data. Our system contains the following validations.

- Validating login form.

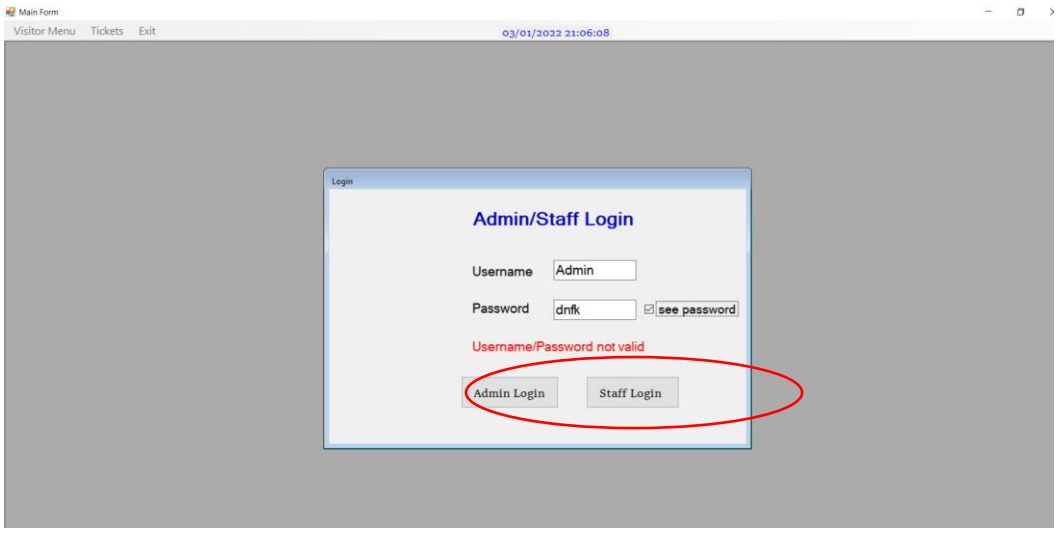


Figure 21: Validating login

- Validating entry details

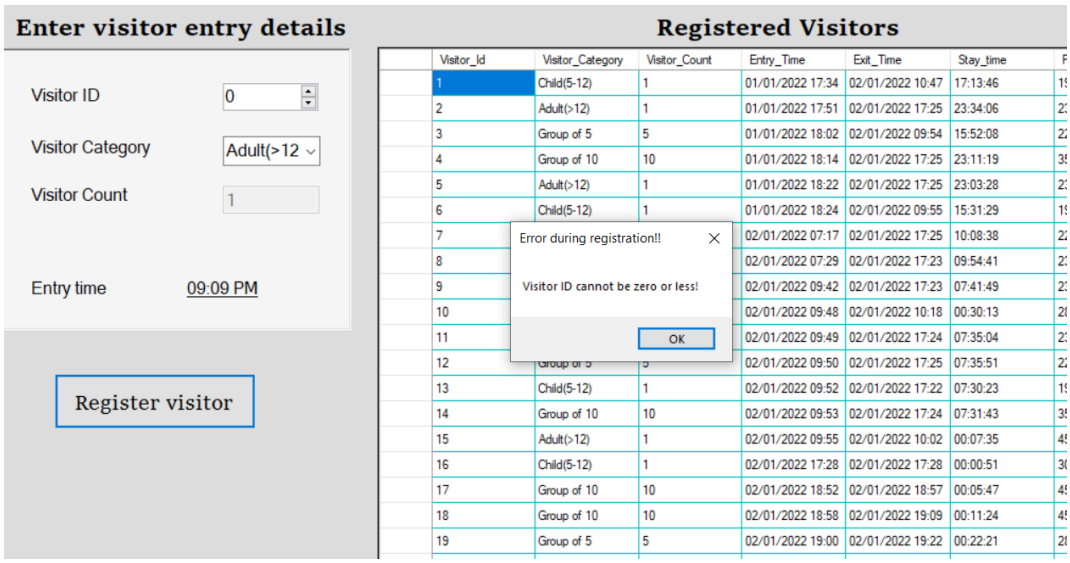


Figure 22: Trying to register as 0

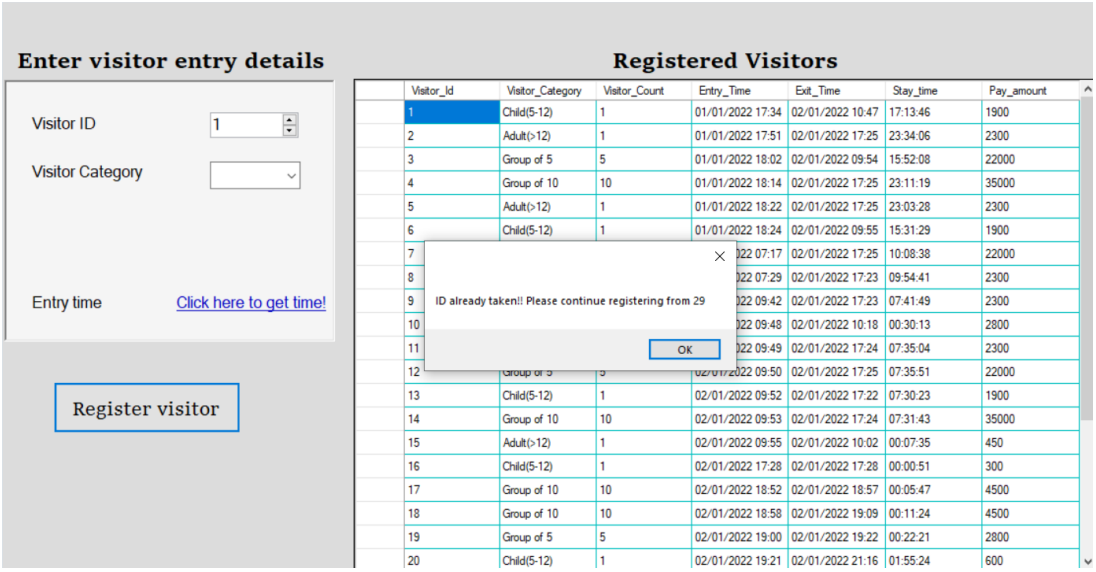


Figure 23: trying to register duplicate ID

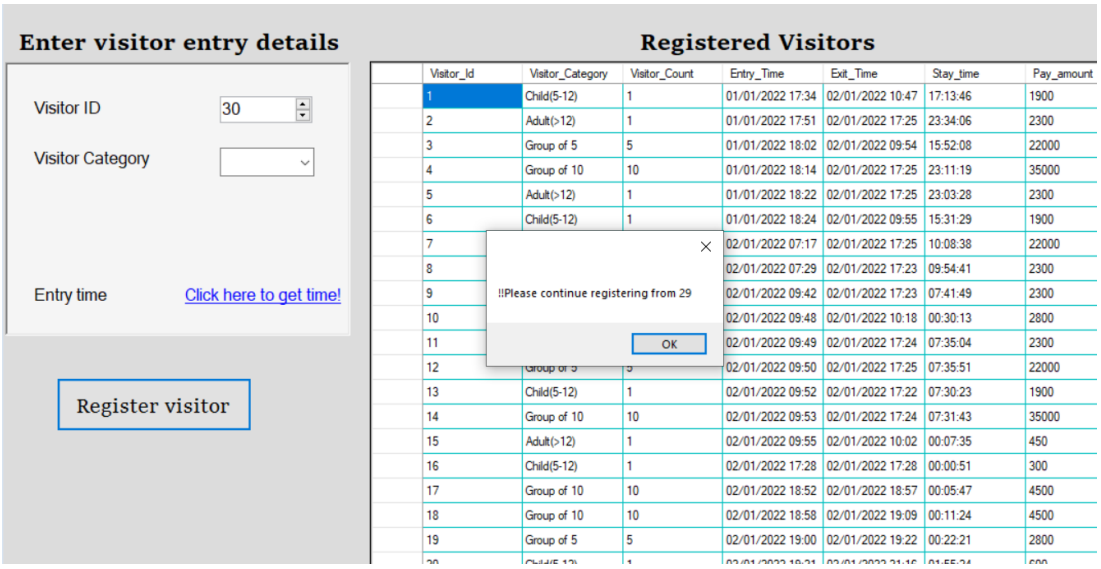


Figure 24: trying to register with random ID

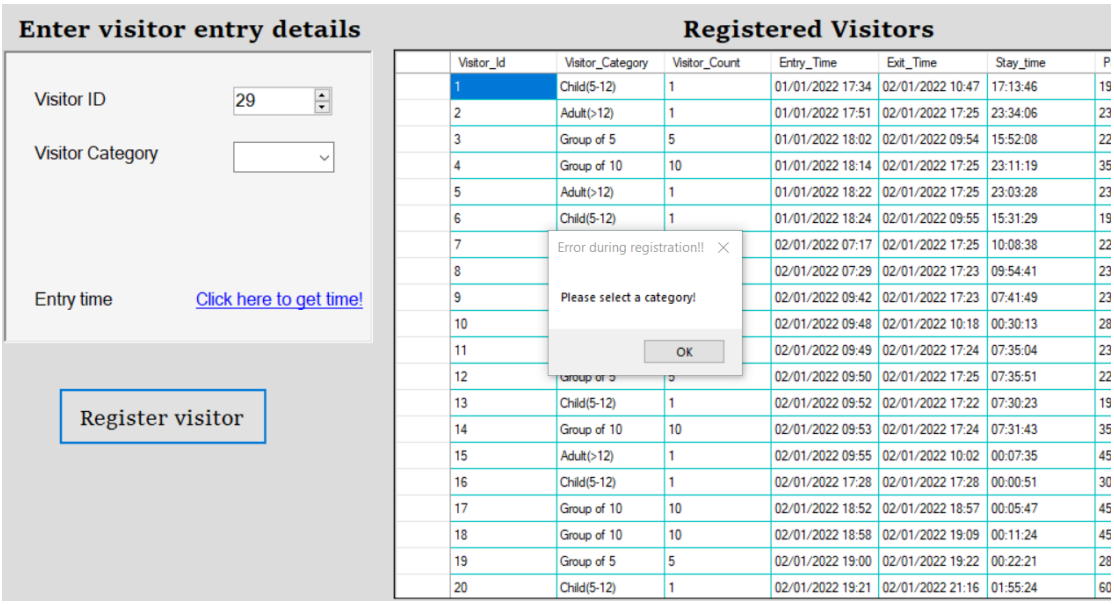


Figure 25: Trying to register without category

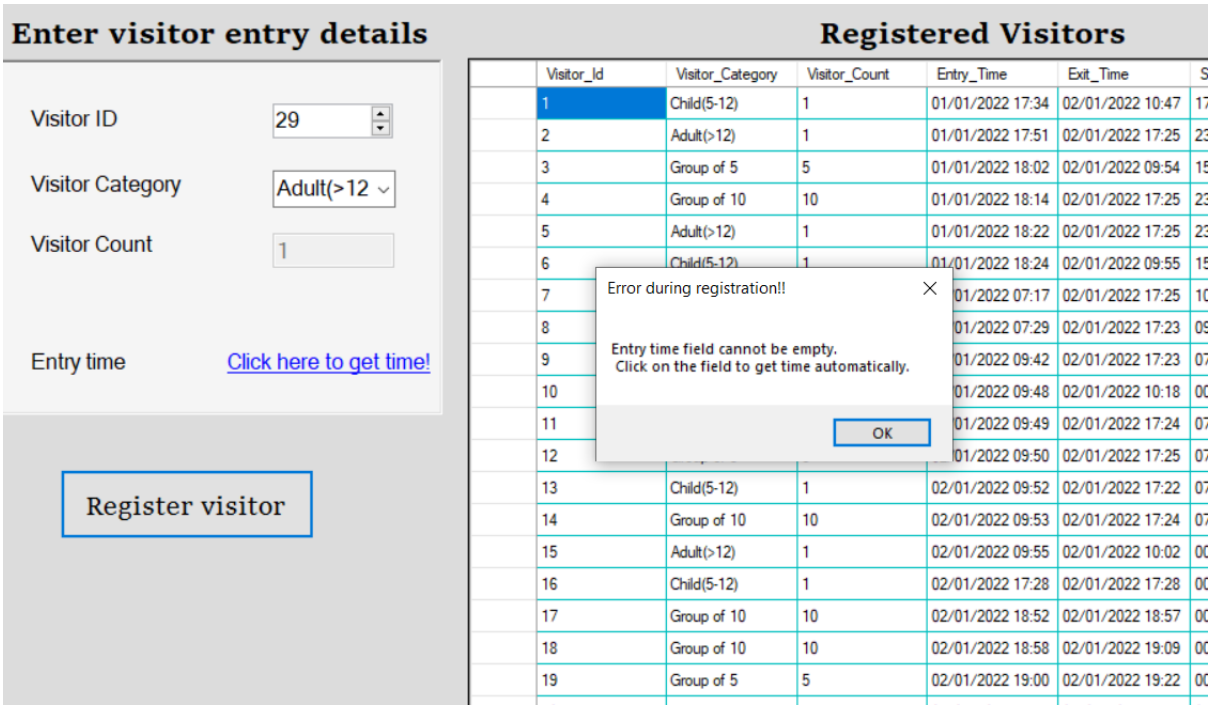


Figure 26: Trying to register without time

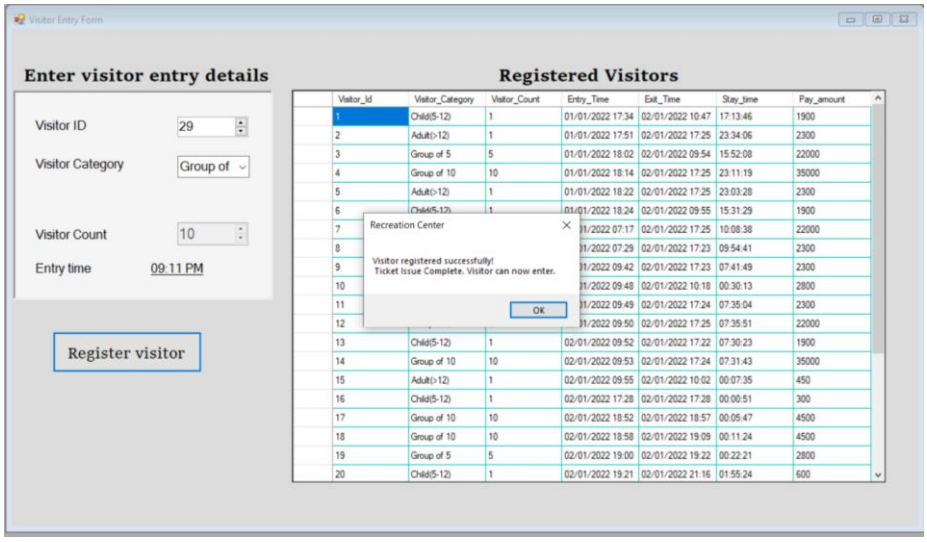


Figure 27: Registering with all valid data

- Validating Exit form

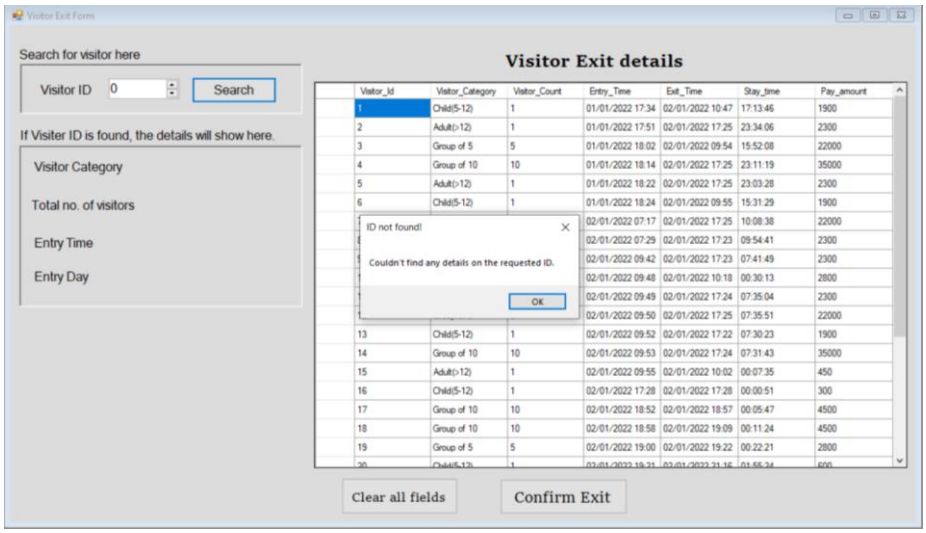


Figure 28: Searching for non-existent ID

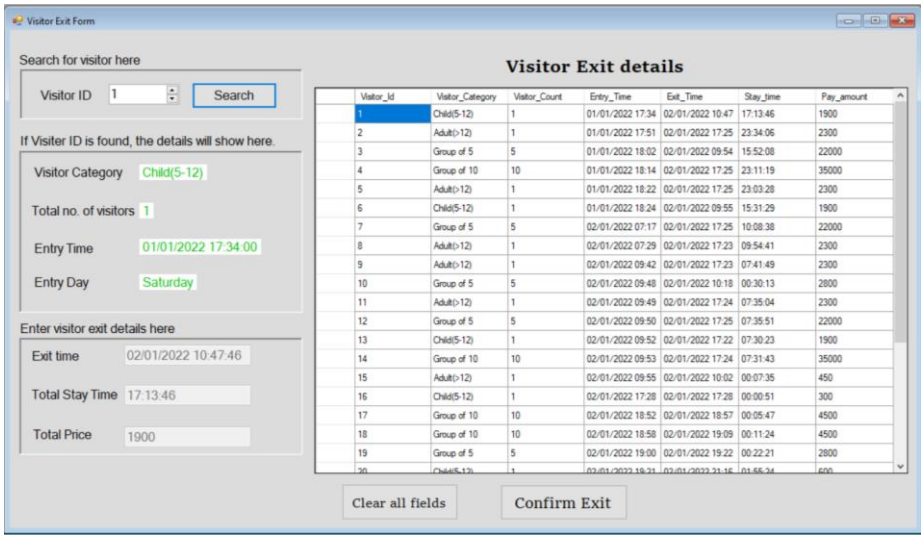


Figure 29: Search results for existing ID

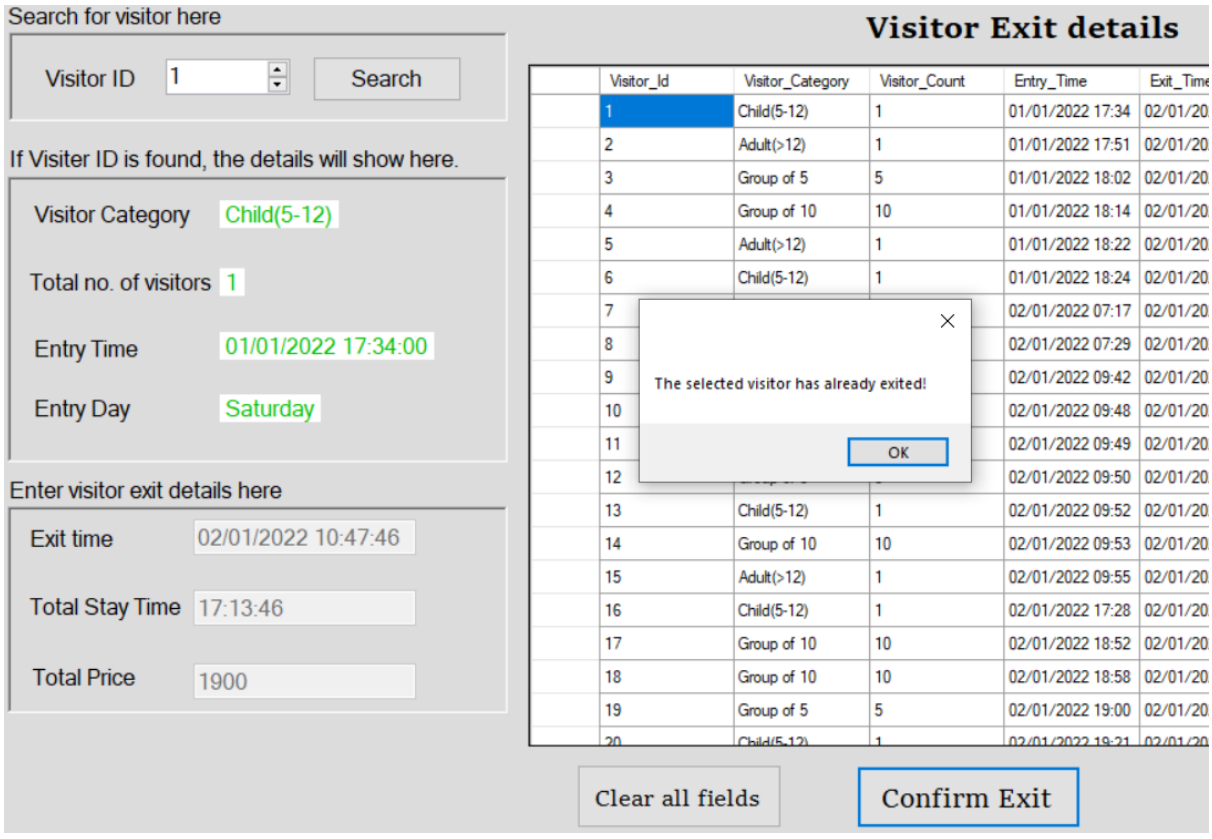


Figure 30: trying to exit same visitor twice

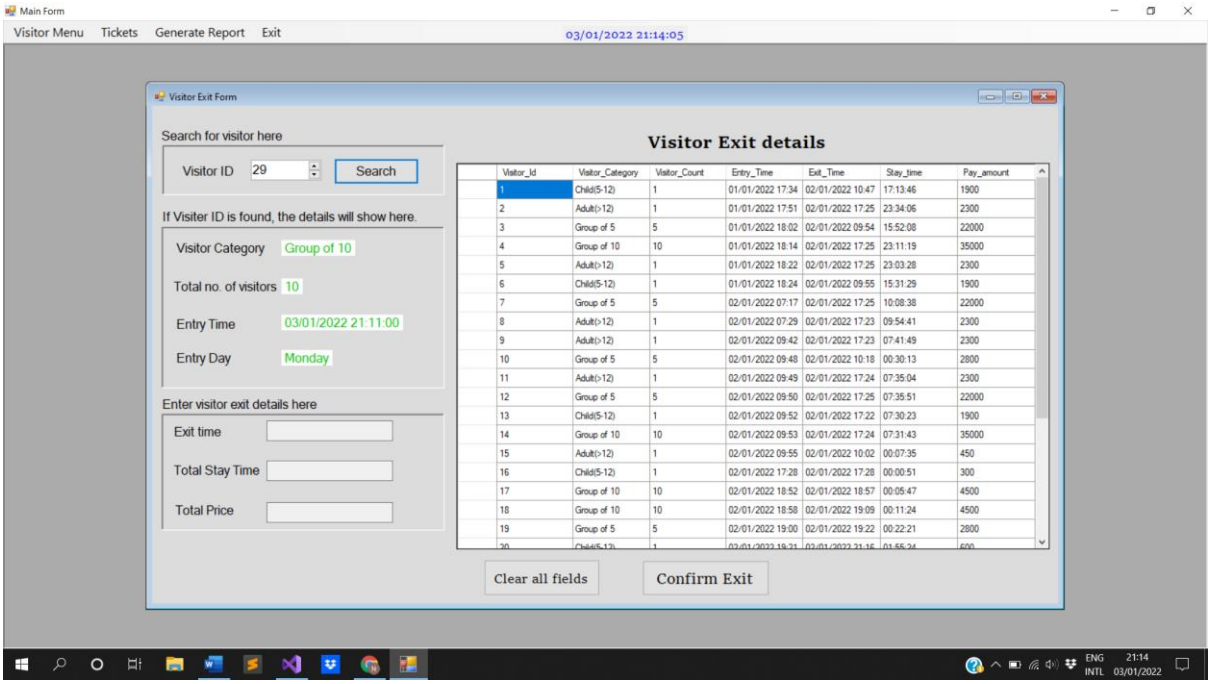


Figure 31: search results for visitor who hasn't left

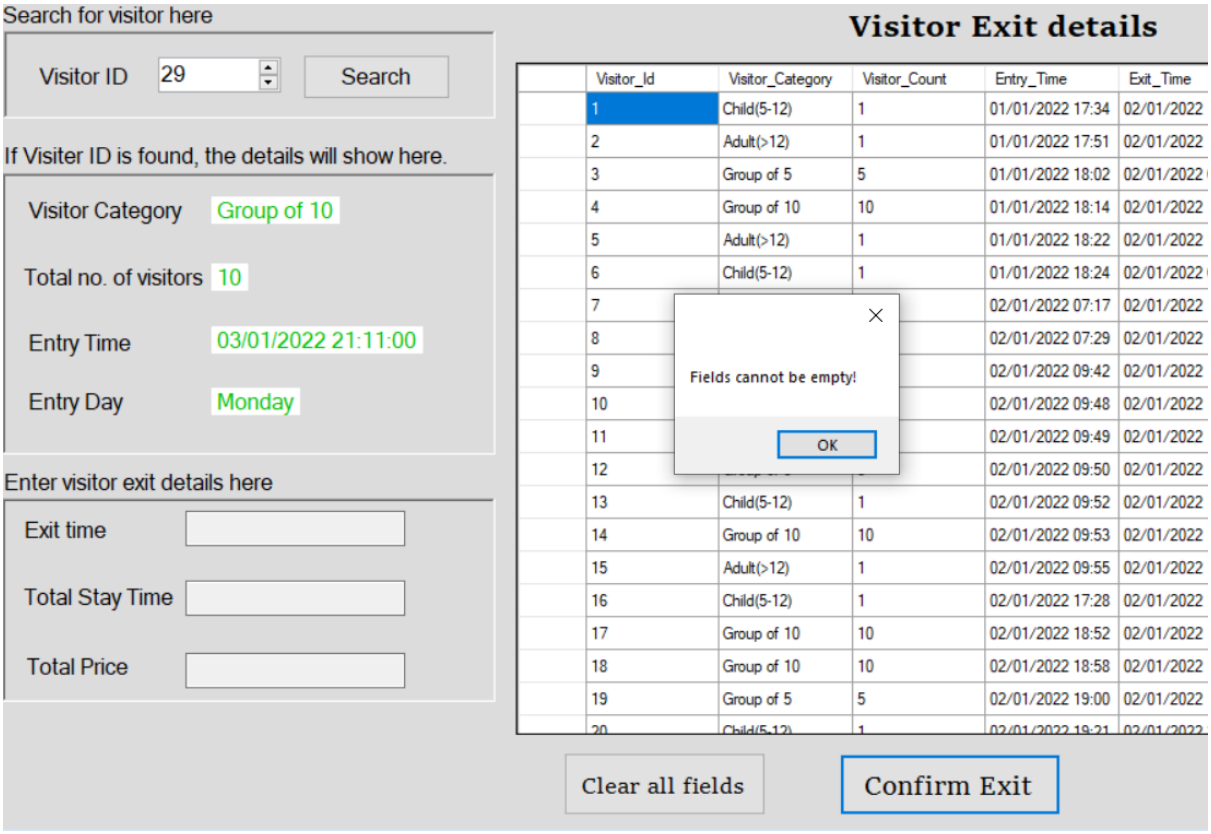


Figure 32: Trying to exit with fields empty

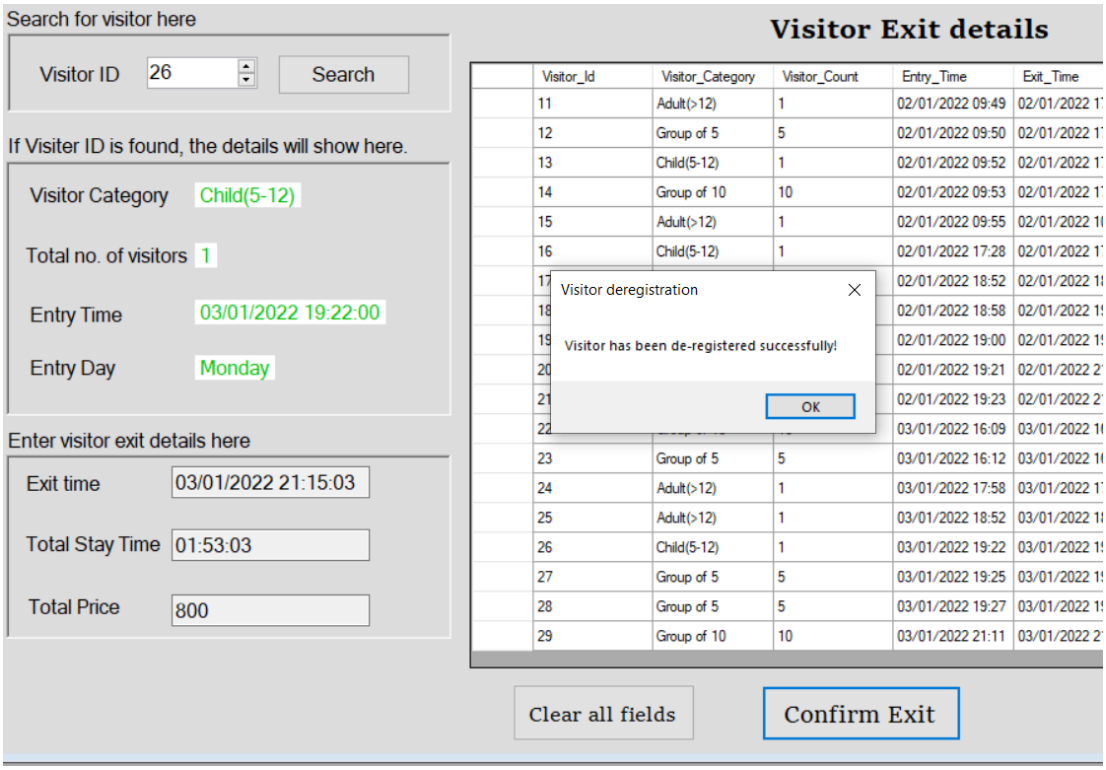


Figure 33: De-registering visitor successfully

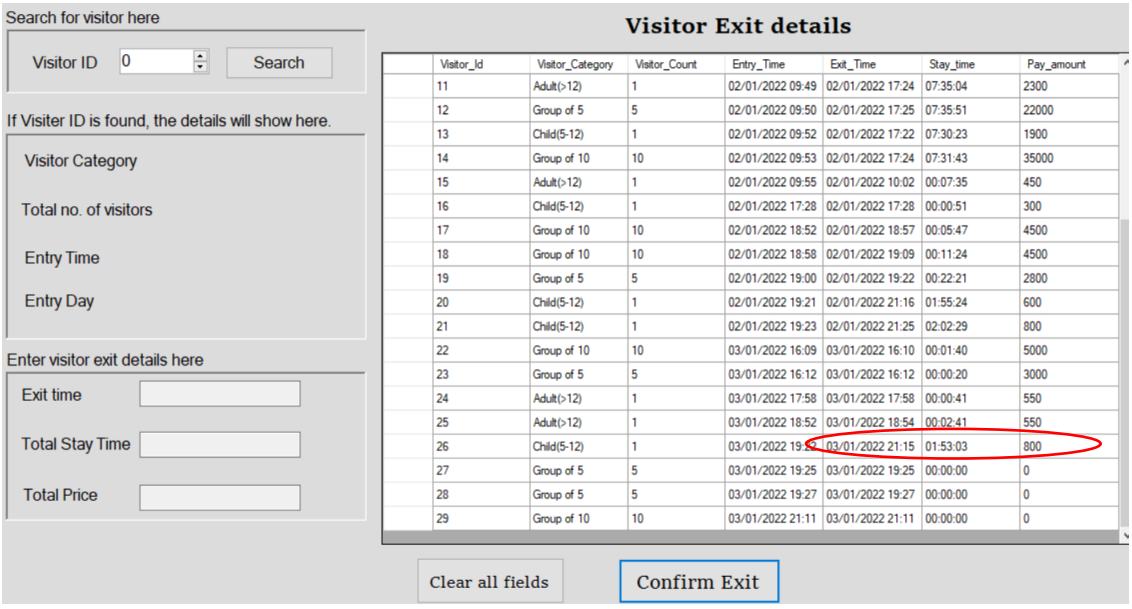


Figure 34: Successfully updated list after visitor exit

- Validating ticket price update

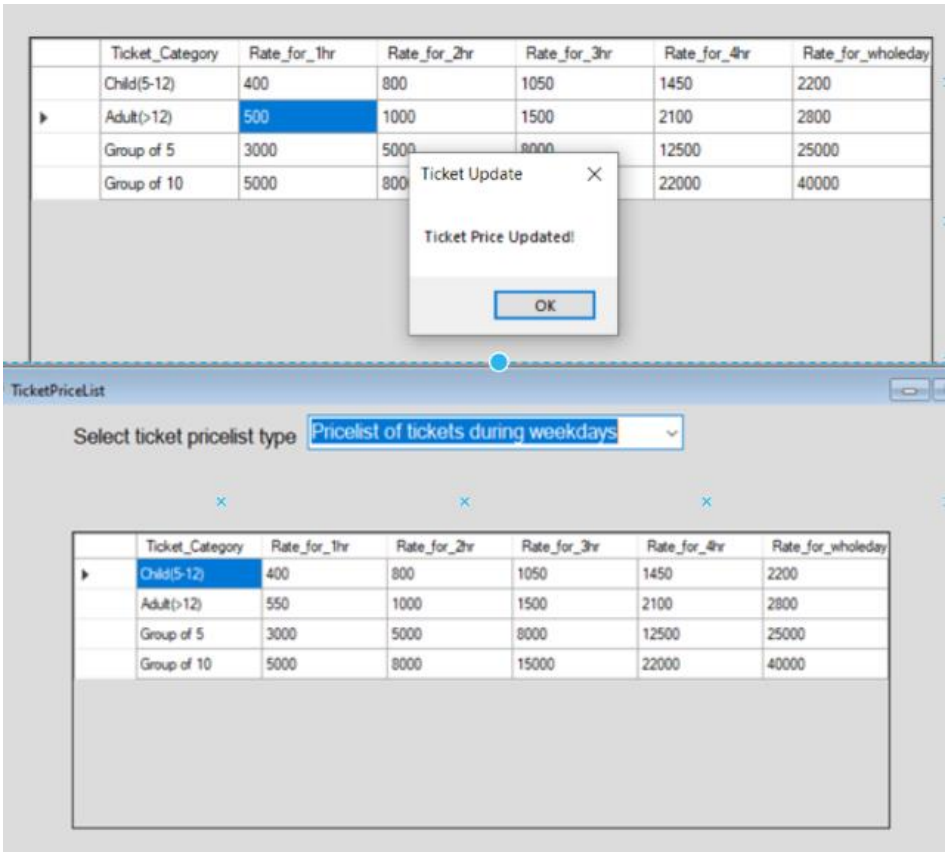


Figure 35: Successfully updating ticket price

- Validating Menu Items

We have already seen an example of menu items validation in User Manual Section **2.4.**

6 Methods used in forms

6.1 Login Form

- btnAdmin_Click(): Logs in as Admin.
- btnStaff_Click(): Logs in as Staff.
- passwordtxt_TextChanged(): secures password.
- Pw_see_hide_CheckedChanged(): hides or shows password.

6.2 Main form

- f1_FormClosed(): close event for f1.
- ShowLogin(): displays login form.
- VisitorEntryForm_Click(): Opens visitor entry form.
- VisitorExitForm_Click(): Opens visitor exit form.
- TicketsList_Click(): Opens ticketlist form.
- getdailyreport_Click(): Opens daily report form.
- getweeklyreport_Click(): Opens weekly report form.
- LogOut_Click(): Opens Dialog and gives logout options.
- Exit_Click(): closes the system;

6.3 Visitor Entry Form

- visitorCategory_SelectedIndexChanged(): displays components based on the selected value.
- VisitorEntryForm_Load(): activates event inside when the form is opened.
- ValidateVisitorDetails(): checks if the data is correct or not as programmed.
- ValidateVisitorCategory(): checks if the visitorcategory input is correct or not.
- ValidateVisitorEntryTime(): checks the entry time.
- AddVisitor_Click(): Adds visitor data in list and serialize the data.
- VisitorEntryTime_Click(): gets time when clicked.
- AddVisitorToCsv(): Adds data from list in csv file.
- ClearInputs(): clears input fields when called.

6.4 Visitor Exit Form

- Searchvisitor_Click(): Deserializes the data and search for visitor.
- RetriveVisitor(): retrives all data of visitor and display.
- VisitorExitTime_Click(): gets current time.
- Visitorstaytime_Click(): gets visitor stay time.
- Pay_amount_Click(): gets amount to be paid.
- ClearInputs(): clears input fields when called.
- VisitorExitForm_Load(): activates event inside when the form is opened
- VisitorConfirmExit_Click (): de-registers visitor from the list and csv file.

6.5 Ticket Price List Form

- Ticket_day_type_SelectedIndexChanged(): shows corresponding ticketlist csv file.
- UpdateTicketPrice_Click (): overwrites old data in csv file with new data.

6.6 Daily Report Form

- see_report_Click (): shows daily data and chart.
- generateChart (): assigning value into the chart.

6.7 Weekly Report Form

- GetWeeklyReport_Click (): shows total daily data of a week and in chart.

7 Conclusion

In this project, we are assigned to develop an online based visitor management system by “Sam’s place” recreation centre to record their business details. We have used **C # desktop app** in **Visual Studio** to create this system.

Although, Visual Studio made this project very easy, this project isn’t as easy as I described. Many errors and many confusions arose during the making of this project. The given project has come to an end. To complete this coursework, we expended a lot of time on studies and research in the given field. I really enjoyed doing it even though it was frustrating when a small error leads you to delete a whole file. After visiting so many websites and tutorials and videos, it certainly sharpened my research habits. I would like to thank and congratulate the individuals that helped me in this task.

APPENDIX

Login form

```
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 Coursework1

{

   public partial class Login : Form

   {

      public Login()

      {

         InitializeComponent();

         LoginMessage.Visible = false;
```

```
        }

        private void label3_Click(object sender, EventArgs e)
        {

        }

        private void btnAdmin_Click(object sender, EventArgs e)
        {

            //check username and password
            if (usernametxt.Text == "Admin" && passwordtxt.Text == "Admin")
            {
                //set the user as admin
                GlobalValues.IsAdmin = true;
                MessageBox.Show("Successfully Logged in as Admin.");
                //Close the login form
                Close();
            }
            else
```

```
        {
            //if not valid -> show error message
            LoginMessage.Visible = true;
        }
    }

    private void label4_Click(object sender, EventArgs e)
    {

    }

    private void btnStaff_Click(object sender, EventArgs e)
    {
        if (usernametxt.Text == "Staff" && passwordtxt.Text == "Staff")
        {
            //set the user as staff
            GlobalValues.IsAdmin = false;
            MessageBox.Show("Successfully Logged in as Staff.");
            //Close the login form
            Close();
```

```
        }
        else
        {
            //if not valid -> show error message
            LoginMessage.Visible = true;
        }
    }

    private void button1_Click(object sender, EventArgs e)
    {

    }

    private void Login_Load(object sender, EventArgs e)
    {

    }

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

```
            passwordtxt.UseSystemPasswordChar = true;
        }

        private void Pw_see_hide_CheckedChanged(object sender, EventArgs e)
        {
            if (Pw_see_hide.Checked)
            {
                passwordtxt.UseSystemPasswordChar = false;
            }
            else
            {
                passwordtxt.UseSystemPasswordChar = true;
            }
        }
    }
}
```

Main form

```
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 Coursework1
{
    public partial class MainForm : Form
    {
        public MainForm()
        {
            InitializeComponent();
            ShowLogin();
        }
```

```
private void timer1_Tick(object sender, EventArgs e)
{
    Date_display.Text = DateTime.Now.ToString();
}

Login f1;
private void f1_FormClosed(object sender, FormClosedEventArgs e)
{
    f1 = null;
    //enable menu

    menuStrip.Enabled = true;
    Exit.Enabled = true;
    if (GlobalValues.IsAdmin == true)
    {
        ViewReport.Visible = true;
    }

}
```

```
private void ShowLogin()
{
    if (f1 == null)
    {
        f1 = new Login();
        f1.FormClosed += f1_FormClosed;
        f1.MdiParent = this;
        f1.StartPosition = FormStartPosition.CenterScreen;
        f1.Show();
        menuStrip.Enabled = false;
    }
    else
    {
        f1.Activate();
    }
}
```

```
private void MainForm_Load_1(object sender, EventArgs e)
```

```
{
    Date_display.Text = DateTime.Now.ToString();
}

private void VisitorMenu_Click(object sender, EventArgs e)
{
}

VisitorEntryForm visitform;

private void Visitform_FormClosed(object sender, FormClosedEventArgs e)
{
    visitform = null;
}
private void VisitorEntryForm_Click(object sender, EventArgs e)
{
    if (visitform == null)
    {
        visitform = new VisitorEntryForm();
        visitform.FormClosed += Visitform_FormClosed;
```

```
            visitform.MdiParent = this;
            visitform.StartPosition = FormStartPosition.CenterScreen;
            visitform.Show();
        }
        else
        {
            visitform.Activate();
        }
    }

    VisitorExitForm exitform;

    private void Exitform_FormClosed(object sender, FormClosedEventArgs e)
    {
        exitform = null;
    }

    private void VisitorExitForm_Click(object sender, EventArgs e)
    {
        if (exitform == null)
```

```
        {
            exitform = new VisitorExitForm();
            exitform.FormClosed += Exitform_FormClosed;
            exitform.MdiParent = this;
            exitform.StartPosition = FormStartPosition.CenterScreen;
            exitform.Show();
        }
        else
        {
            exitform.Activate();
        }
    }

    TicketPriceList ticket;
    private void TicketsList_Click(object sender, EventArgs e)
    {
        if (ticket == null)
        {
            ticket = new TicketPriceList();
            ticket.FormClosed += TicketList_FormClosed;
```

```
            ticket.MdiParent = this;
            ticket.StartPosition = FormStartPosition.CenterScreen;
            ticket.Show();
        }
        else
        {
            ticket.Activate();
        }
    }

    private void TicketList_FormClosed(object sender, FormClosedEventArgs e)
    {
        ticket = null;
    }

   /*  DailyReport rep;

    private void ViewReport_Click(object sender, EventArgs e)
    {
        if (rep == null)
```

```
        {
            rep = new DailyReport();
            rep.FormClosed += Report_FormClosed;
            rep.MdiParent = this;
            rep.StartPosition = FormStartPosition.CenterScreen;
            rep.Show();
        }
        else
        {
            rep.Activate();
        }
    }*/

    private void DailyReport_FormClosed(object sender, FormClosedEventArgs e)
    {
        rep = null;
    }

    DailyReport rep;
    private void getdailyreport_Click(object sender, EventArgs e)
```

```
        {
            if (rep == null)
            {
                rep = new DailyReport();
                rep.FormClosed += DailyReport_FormClosed;
                rep.MdiParent = this;
                rep.StartPosition = FormStartPosition.CenterScreen;
                rep.Show();
            }
            else
            {
                rep.Activate();
            }
        }

        private void WeeklyReport_FormClosed(object sender, FormClosedEventArgs e)
        {
            rep = null;
        }
```

```
WeeklyReport wr;
private void getweeklyreport_Click(object sender, EventArgs e)
{
   if (wr == null)
   {
      wr = new WeeklyReport();
      wr.FormClosed += WeeklyReport_FormClosed;
      wr.MdiParent = this;
      wr.StartPosition = FormStartPosition.CenterScreen;
      wr.Show();
   }
   else
   {
      wr.Activate();
   }
}

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

```
        DialogResult dialogResult = MessageBox.Show("Are you sure you want to log out?", "Logout", MessageBoxButtons.YesNo);
        if (dialogResult == DialogResult.Yes)
        {
            ViewReport.Visible = false;


            ShowLogin();
            if (GlobalValues.IsAdmin == false)
            {
                ViewReport.Visible = false;
            }
        }
        else
        {


        }
    }

    private void Exit_Click(object sender, EventArgs e)
```

```
            {
                this.Close();
            }
        }
    }
```

Entry Form

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.IO;
using System.Linq;
using System.Runtime.Serialization.Formatters.Binary;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Coursework1
{
    public partial class VisitorEntryForm : Form
    {

        string filePath = "registered_visitors.csv";
```

```
public VisitorEntryForm()
{
    InitializeComponent();
}
private void visitorCategory_SelectedIndexChanged(object sender, EventArgs e)
{
    if ((string)visitorCategory.SelectedItem == "Child(5-12)" || (string)visitorCategory.SelectedItem == "Adult(>12)")
    {
        countlabel.Visible = true;
        visitorCount.Visible = true;
        Groupcountlabel.Visible = false;
        visitorCountGroup.Visible = false;
        visitorCount.Text = "1";
    }

    else if ((string)visitorCategory.SelectedItem == "Group of 5")
    {
        Groupcountlabel.Visible = true;
```

```
            visitorCountGroup.Visible = true;
            countlabel.Visible = false;
            visitorCount.Visible = false;
            visitorCountGroup.Value = 5;
        }

        else if ((string)visitorCategory.SelectedItem == "Group of 10")
        {
            Groupcountlabel.Visible = true;
            visitorCountGroup.Visible = true;
            countlabel.Visible = false;
            visitorCount.Visible = false;
            visitorCountGroup.Value = 10;
        }
    }

    private void VisitorEntryForm_Load(object sender, EventArgs e)
    {
        GlobalValues.RegisteredVisitorList = new List<Visitor>();
        string[] lines = File.ReadAllLines(filePath);
```

```
            foreach (string line in lines)
            {
                string[] st = line.Split(',');


                if (st.Length == 7)
                {
                    Visitor vis = new Visitor();
                    vis.Visitor_Id = Convert.ToInt32(st[0]);
                    vis.Visitor_Category = st[1];
                    vis.Visitor_Count = Convert.ToInt32(st[2]);
                    vis.Entry_Time = Convert.ToDateTime(st[3]);
                    vis.Exit_Time = Convert.ToDateTime(st[4]);
                    vis.Stay_time = TimeSpan.Parse(st[5]);
                    vis.Pay_amount = float.Parse(st[6]);


                    GlobalValues.RegisteredVisitorList.Add(vis);
                }
                VisitorView.DataSource = GlobalValues.RegisteredVisitorList;
            }
        }
```

```
        private void ValidateVisitorDetails()
        {
            Visitor v = new Visitor();
            bool exists = false;
            foreach (Visitor vis in GlobalValues.RegisteredVisitorList)
            {
                if ((int)visitorID.Value == vis.Visitor_Id)
                {
                    MessageBox.Show("ID already taken!! Please continue registering from " +
(GlobalValues.RegisteredVisitorList.Count + (int)1));
                    exists = true;
                    break;
                }
                else if (visitorID.Value < 1)
                {
                    MessageBox.Show("Visitor ID cannot be zero or less!", "Error during
registration!!");
                    exists = true;
                    break;
                }
```

```
            else if (visitorID.Value != (GlobalValues.RegisteredVisitorList.Count + (int)1))
            {
                MessageBox.Show("!!Please continue registering from " + (GlobalValues.RegisteredVisitorList.Count + (int)1));
                exists = true;
                break;
            }
        }

        if(!exists)
        {
            v.Visitor_Id = (int)visitorID.Value;
            ValidateVisitorCategory(v);
        }

    }
    private void ValidateVisitorCategory(Visitor v)
    {
        if ((string)visitorCategory.SelectedItem == "Child(5-12)" || (string)visitorCategory.SelectedItem == "Adult(>12)")
```

```
            {
                v.Visitor_Category = (String)visitorCategory.SelectedItem;
                v.Visitor_Count = Convert.ToInt32(visitorCount.Text);
                ValidateVisitorEntryTime(v);
            }

            else if ((string)visitorCategory.SelectedItem == "Group of 5" || (string)visitorCategory.SelectedItem == "Group of 10")
            {
                v.Visitor_Category = (String)visitorCategory.SelectedItem;
                v.Visitor_Count = (int)visitorCountGroup.Value;
                ValidateVisitorEntryTime(v);

            }
            else
            {
                MessageBox.Show("Please select a category!", "Error during registration!!");
            }
        }
        private void ValidateVisitorEntryTime(Visitor v)
```

```
        {
            if (VisitorEntryTime.Text.Equals("Click here to get time!"))
            {
                MessageBox.Show("Entry time field cannot be empty. " +
                    "\n Click on the field to get time automatically.", "Error during registration!!");
            }
            else
            {
                v.Entry_Time = DateTime.Parse(VisitorEntryTime.Text);
                v.Exit_Time = v.Entry_Time;
                //v.Pay_amount = 0;

                GlobalValues.RegisteredVisitorList.Add(v);
                AddVisitorToCsv(v.Visitor_Id, v.Visitor_Category, v.Visitor_Count, v.Entry_Time, v.Exit_Time, v.Stay_time, v.Pay_amount);
                VisitorView.DataSource = null;
                VisitorView.DataSource = GlobalValues.RegisteredVisitorList;
                MessageBox.Show("Visitor registered successfully!" +
                    "\n Ticket Issue Complete. Visitor can now enter.", "Recreation Center");
            }
```

```
        }

        private void AddVisitor_Click(object sender, EventArgs e)
        {
            try
            {
                ValidateVisitorDetails();
                ClearInputs();
                //prepare visitor list object for serialization
                GlobalValues.VisitorRecordList = new VisitorCollection();
                GlobalValues.VisitorRecordList.VisitorRecords = GlobalValues.RegisteredVisitorList;

                //serializing the visitors details
                //FileStream Stream = new FileStream("C:\\Project Files\\visitors"+"\\"+"visitor"+ s.Visitor_Id, FileMode.Create);
                FileStream Stream = new FileStream("visitors\\visitorCollection", FileMode.Create);
                BinaryFormatter formatter = new BinaryFormatter();
                formatter.Serialize(Stream, GlobalValues.VisitorRecordList);
                Stream.Close();
```

```
            //GlobalValues.VisitorExitList = GlobalValues.RegisteredVisitorList;

            /*List<string> lines = new List<string>();

            foreach (Visitor visit in GlobalValues.VisitorRecordList.VisitorRecords)

            //foreach (Visitor visit in GlobalValues.VisitorExitList)

            {

               string line = visit.Visitor_Id + "," + visit.Visitor_Category + "," + visit.Visitor_Count + "," + visit.Entry_Time + "," + visit.Exit_Time+ "," + visit.Stay_time+ "," + visit.Pay_amount + "\n";

               lines.Add(line);

            }

            File.WriteAllLines("C:\\Project Files\\exit_visitors.csv", lines);*/



        }

        catch (FormatException)

        {

            MessageBox.Show("Unknown error occured during registration. Please try again!!", "!!UNKNOWN ERROR!! ");

        }

    }

    private void VisitorEntryTime_Click(object sender, System.EventArgs e)
```

```
        {
            VisitorEntryTime.Text = DateTime.Now.ToString("hh:mm tt");
            VisitorEntryTime.ForeColor = Color.Black;
        }

        private void AddVisitorToCsv(int id, string category, int count, DateTime entry, DateTime exit, TimeSpan stay, float payment)
        {
            string newSt = "\n" + id + "," + category + "," + count + "," + entry + "," + exit + ","+ stay + "," + payment;
            File.AppendAllText(filePath, newSt);
        }

        private void ClearInputs()
        {
            visitorID.Value = 0;
            visitorCategory.SelectedIndex = -1;
            countlabel.Visible = false;
            Groupcountlabel.Visible = false;
            visitorCount.Visible = false;
            visitorCountGroup.Visible = false;
```

```
            visitorCountGroup.Value = 0;

            VisitorEntryTime.Text = "Click here to get time!";

            VisitorEntryTime.ForeColor = Color.Blue;

        }

    }

}
```

Exit form

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.IO;
using System.Linq;
using System.Runtime.Serialization.Formatters.Binary;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;


namespace Coursework1
{
    public partial class VisitorExitForm : Form
    {
        string FilePath;
        string filePath = "registered_visitors.csv";
        public VisitorExitForm()
```

```
        {
            InitializeComponent();
        }

        private void Searchvisitor_Click(object sender, EventArgs e)
        {
            FileStream Stream = new FileStream("visitors\\visitorCollection", FileMode.Open);
            BinaryFormatter formatter = new BinaryFormatter();
            GlobalValues.VisitorRecordList = (VisitorCollection)formatter.Deserialize(Stream);
            //s = (Visitor)formatter.Deserialize(Stream);
            Stream.Flush();
            RetriveVisitor();
            Stream.Close();
        }
        private void RetriveVisitor()
        {
            bool found = false;
            int id = (int)VisitorID.Value;
            foreach (Visitor v in GlobalValues.VisitorRecordList.VisitorRecords)
```

```
            {
                //MessageBox.Show("Count:"+
GlobalValues.VisitorRecordList.VisitorRecords.Count);
                if (id == v.Visitor_Id && v.Pay_amount == 0)
                {
                    VisitorCategory.Text = v.Visitor_Category;
                    VisitorCount.Text = v.Visitor_Count.ToString();
                    showvisitorentry.Text = v.Entry_Time.ToString();
                    visit_day.Text = v.Entry_Time.ToString("dddddddd");
                    found = true;
                    VisitorExitTime.Enabled = true;
                    visitorstaytime.Enabled = true;
                    pay_amount.Enabled = true;
                    VisitorExitTime.Text = "";
                    visitorstaytime.Text = "";
                    pay_amount.Text = "";
                    VisitorExitTime.ReadOnly = true;
                    visitorstaytime.ReadOnly = true;
                    pay_amount.ReadOnly = true;
                    VisitorCategory.Visible = true;
```

```
            VisitorCount.Visible = true;
            exitdetailslabel.Visible = true;
            ExitDetailsPanel.Visible = true;
            showvisitorentry.Visible = true;
            visit_day.Visible = true;
            break;
        }
        else if (id == v.Visitor_Id && v.Pay_amount != 0)
        {
            VisitorCategory.Text = v.Visitor_Category;
            VisitorCount.Text = v.Visitor_Count.ToString();
            showvisitorentry.Text = v.Entry_Time.ToString();
            visit_day.Text = v.Entry_Time.ToString("dddddddd");
            VisitorExitTime.Text = v.Exit_Time.ToString();
            visitorstaytime.Text = v.Stay_time.ToString();
            pay_amount.Text = v.Pay_amount.ToString();
            VisitorExitTime.Enabled = false;
            visitorstaytime.Enabled = false;
            pay_amount.Enabled = false;
            VisitorCategory.Visible = true;
```

```
                VisitorCount.Visible = true;
                exitdetailslabel.Visible = true;
                ExitDetailsPanel.Visible = true;
                showvisitorentry.Visible = true;
                visit_day.Visible = true;
                found = true;
                break;
            }

        }
        if (!found)
        {
            MessageBox.Show("Couldn't find any details on the requested ID.", "ID not found!");
        }
    }
    private void VisitorExitTime_Click(object sender, System.EventArgs e)
    {
        VisitorExitTime.Text = DateTime.Now.ToString();
    }
```

```
        private void Visitorstaytime_Click(object sender, System.EventArgs e)
        {
            if (VisitorExitTime.Text != "")
            {
                TimeSpan time = Convert.ToDateTime(VisitorExitTime.Text) -
Convert.ToDateTime(showvisitorentry.Text);
                visitorstaytime.Text = time.ToString();
            }
            else
            {
                MessageBox.Show("No exit time found! Please click on the exit field. ");
            }

        }

        private void Pay_amount_Click(object sender, System.EventArgs e)
        {
            TimeSpan time = Convert.ToDateTime(VisitorExitTime.Text) -
Convert.ToDateTime(showvisitorentry.Text);
            string temp_time = time.TotalMinutes.ToString();
```

```
        float timeofstay = float.Parse(temp_time);


        if (visit_day.Text == "Monday" || visit_day.Text == "Tuesday" || visit_day.Text
== "Wednesday" || visit_day.Text == "Thursday" || visit_day.Text == "Friday" )
        {
            FilePath = "Weekday_tickets.csv";
        }
        else if (visit_day.Text == "Saturday" || visit_day.Text == "Sunday")
        {
            FilePath = "Weekend_tickets.csv";
        }


        GlobalValues.TicketList = new List<Ticketpricerate>();
        string[] lines = File.ReadAllLines(FilePath);
        foreach (string line in lines)
        {
            string[] st = line.Split(',');


            if (st.Length == 6)
```

```
        {
            Ticketpricerate tpr = new Ticketpricerate();
            tpr.Ticket_Category = st[0];
            tpr.Rate_for_1hr = Convert.ToInt32(st[1]);
            tpr.Rate_for_2hr = Convert.ToInt32(st[2]);
            tpr.Rate_for_3hr = Convert.ToInt32(st[3]);
            tpr.Rate_for_4hr = Convert.ToInt32(st[4]);
            tpr.Rate_for_wholeday = Convert.ToInt32(st[5]);
            GlobalValues.TicketList.Add(tpr);
        }
    }

    foreach (Ticketpricerate rate in GlobalValues.TicketList)
    {
        if (VisitorCategory.Text == rate.Ticket_Category)
        {
            if (timeofstay <= 60)
            {
                pay_amount.Text = rate.Rate_for_1hr.ToString();
            }
```

```
            else if (timeofstay > 60 && timeofstay <= 120)
            {
                pay_amount.Text = rate.Rate_for_2hr.ToString();
            }
            else if (timeofstay > 120 && timeofstay <= 180)
            {
                pay_amount.Text = rate.Rate_for_3hr.ToString();
            }
            else if (timeofstay > 180 && timeofstay <= 240)
            {
                pay_amount.Text = rate.Rate_for_4hr.ToString();
            }
            else
            {
                pay_amount.Text = rate.Rate_for_wholeday.ToString();
            }
        }
    }
}
```

```
        private void clearInputs()
        {
            VisitorID.Value = 0;
            VisitorCategory.Visible = false;
            VisitorCount.Visible = false;
            showvisitorentry.Visible = false;
            visit_day.Visible = false;
            VisitorExitTime.Text = "";
            visitorstaytime.Text = "";
            pay_amount.Text = "";
        }

        private void ClearVisitordetails_Click(object sender, EventArgs e)
        {
            clearInputs();
        }

        private void VisitorExitForm_Load(object sender, EventArgs e)
        {
            GlobalValues.RegisteredVisitorList = new List<Visitor>();
```

```
string[] lines = File.ReadAllLines(filePath);
foreach (string line in lines)
{

   string[] st = line.Split(',');

   if (st.Length == 7)
   {
      Visitor v = new Visitor();
      v.Visitor_Id = Convert.ToInt32(st[0]);
      v.Visitor_Category = st[1];
      v.Visitor_Count = Convert.ToInt32(st[2]);
      v.Entry_Time = Convert.ToDateTime(st[3]);
      v.Exit_Time = Convert.ToDateTime(st[4]);
      v.Stay_time = TimeSpan.Parse(st[5]);
      v.Pay_amount = float.Parse(st[6]);

      GlobalValues.RegisteredVisitorList.Add(v);
   }
   VisitorExitView.DataSource = GlobalValues.RegisteredVisitorList;
```

```
            }
        }

        private void VisitorConfirmExit_Click(object sender, EventArgs e)
        {
            if (VisitorExitTime.Text == "" || visitorstaytime.Text == "" || pay_amount.Text
== "")
            {
                MessageBox.Show("Fields cannot be empty!");
            }
            else
            {
                foreach (Visitor v in GlobalValues.RegisteredVisitorList)
                {
                    if ((int)VisitorID.Value == v.Visitor_Id && v.Pay_amount != 0)
                    {
                        MessageBox.Show("The selected visitor has already exited!");
                        break;
                    }
                    else if ((int)VisitorID.Value == v.Visitor_Id)
```

```
                {
                    v.Exit_Time = Convert.ToDateTime(VisitorExitTime.Text);
                    v.Stay_time = TimeSpan.Parse(visitorstaytime.Text);
                    v.Pay_amount = float.Parse(pay_amount.Text);

                    List<string> lines = new List<string>();

                    foreach (Visitor renew in GlobalValues.RegisteredVisitorList)
                    {
                        string line = "\n" + renew.Visitor_Id + "," + renew.Visitor_Category + "," + renew.Visitor_Count + "," + renew.Entry_Time + "," + renew.Exit_Time + "," + renew.Stay_time + "," + renew.Pay_amount ;
                        lines.Add(line);
                    }
                    File.WriteAllLines(filePath, lines);
                    MessageBox.Show("Visitor has been de-registered successfully!", "Visitor deregistration");
                }
            }
            VisitorExitView.DataSource = null;
            VisitorExitView.DataSource = GlobalValues.RegisteredVisitorList;
```

```
            clearInputs();



            VisitorExitTime.Enabled = true;

            visitorstaytime.Enabled = true;

            pay_amount.Enabled = true;

        }

    }



    private void visit_day_Click(object sender, EventArgs e)

    {



    }

  }

}
```

Ticket List

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.IO;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;


namespace Coursework1
{
    public partial class TicketPriceList : Form
    {
        public TicketPriceList()
        {
            InitializeComponent();
        }
```

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

}

string FilePath;
private void Ticket_day_type_SelectedIndexChanged(object sender, EventArgs e)
{
   if (GlobalValues.IsAdmin == true)
   {
      UpdateTicketPrice.Visible = true;
   }
   if ((string)Ticket_day_type.SelectedItem == "Pricelist of tickets during weekdays")
   {
       FilePath = "Weekday_tickets.csv";
   }
   else if ((string)Ticket_day_type.SelectedItem == "Pricelist of tickets during holidays")
```

```
{
    weekenddiscountnotice.Visible = true;
    FilePath = "Weekend_tickets.csv";
}

GlobalValues.TicketList = new List<Ticketpricerate>();
string[] lines = File.ReadAllLines(FilePath);
foreach (string line in lines)
{
    string[] st = line.Split(',');

    if (st.Length == 6)
    {
        Ticketpricerate tpr = new Ticketpricerate();
        tpr.Ticket_Category = st[0];
        tpr.Rate_for_1hr = Convert.ToInt32(st[1]);
        tpr.Rate_for_2hr = Convert.ToInt32(st[2]);
        tpr.Rate_for_3hr = Convert.ToInt32(st[3]);
        tpr.Rate_for_4hr = Convert.ToInt32(st[4]);
        tpr.Rate_for_wholeday = Convert.ToInt32(st[5]);
```

```
                GlobalValues.TicketList.Add(tpr);
            }
        }
        TicketRateView.DataSource = GlobalValues.TicketList;
        TicketRateView.Visible = true;
        if (GlobalValues.IsAdmin == true)
        {
            UpdateTicketPrice.Visible = true;
        }

    }

    private void TicketRateView_CellContentClick(object sender,
DataGridViewCellEventArgs e)
    {

    }

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

```
        GlobalValues.UpdatedrateList = new List<Ticketpricerate>();



        for (int i=0; i<TicketRateView.Rows.Count; i++)

        {

            Ticketpricerate rate = new Ticketpricerate();

            rate.Ticket_Category = TicketRateView.Rows[i].Cells[0].Value.ToString();

            rate.Rate_for_1hr                                                    =
Convert.ToInt32(TicketRateView.Rows[i].Cells[1].Value);

            rate.Rate_for_2hr                                                    =
Convert.ToInt32(TicketRateView.Rows[i].Cells[2].Value);

            rate.Rate_for_3hr                                                    =
Convert.ToInt32(TicketRateView.Rows[i].Cells[3].Value);

            rate.Rate_for_4hr                                                    =
Convert.ToInt32(TicketRateView.Rows[i].Cells[4].Value);

            rate.Rate_for_wholeday                                               =
Convert.ToInt32(TicketRateView.Rows[i].Cells[5].Value);

            GlobalValues.UpdatedrateList.Add(rate);

        }

        //create collection of lines(string)

        List<string> lines = new List<string>();

        foreach (Ticketpricerate r in GlobalValues.UpdatedrateList)

        {
```

```
            string line = r.Ticket_Category + "," + r.Rate_for_1hr + "," + r.Rate_for_2hr +
"," + r.Rate_for_3hr + "," + r.Rate_for_4hr + "," + r.Rate_for_wholeday;

            lines.Add(line);

        }

        if ((string)Ticket_day_type.SelectedItem == "Pricelist of tickets during
weekdays")

        {

            FilePath = "Weekday_tickets.csv";

            File.WriteAllLines(FilePath, lines);

        }

        else if ((string)Ticket_day_type.SelectedItem == "Pricelist of tickets during
holidays")

        {

            FilePath = "Weekend_tickets.csv";

            File.WriteAllLines(FilePath, lines);

        }

        MessageBox.Show("Ticket Price Updated!","Ticket Update");

    }

  }

}
```

Daily Report

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.IO;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;


namespace Coursework1
{
    public partial class DailyReport : Form
    {
        int child = 0;
        float childamount = 0;


        int adult = 0;
```

```
        float adultamount = 0;


        int group0f5 = 0;
        float go5amount = 0;


        int group0f10 = 0;
        float go10amount = 0;


        public DailyReport()
        {
            InitializeComponent();
        }


        private void see_report_Click(object sender, EventArgs e)
        {
            GlobalValues.RegisteredVisitorList = new List<Visitor>();
            string[] lines = File.ReadAllLines("registered_visitors.csv");
            foreach (string line in lines)
            {
                string[] st = line.Split(',');
```

```
            if (st.Length == 7)
            {
                Visitor v = new Visitor();
                v.Visitor_Id = Convert.ToInt32(st[0]);
                v.Visitor_Category = st[1];
                v.Visitor_Count = Convert.ToInt32(st[2]);
                v.Entry_Time = Convert.ToDateTime(st[3]);
                v.Exit_Time = Convert.ToDateTime(st[4]);
                v.Stay_time = TimeSpan.Parse(st[5]);
                v.Pay_amount = float.Parse(st[6]);

                GlobalValues.RegisteredVisitorList.Add(v);
            }
        }

        child = 0;
        childamount = 0;

        adult = 0;
```

```
            adultamount = 0;


            group0f5 = 0;
            go5amount = 0;


            group0f10 = 0;
            go10amount = 0;


            foreach (Visitor v in GlobalValues.RegisteredVisitorList)
            {
               if (v.Entry_Time.DayOfYear != DateofReport.Value.DayOfYear) continue;
               if (v.Visitor_Category == "Child(5-12)")
               {
                  child += v.Visitor_Count;
                  ChildCount.Text = child.ToString();


                  childamount += v.Pay_amount;
                  ChildTotalIncome.Text = childamount.ToString();
               }
               else if (v.Visitor_Category == "Adult(>12)")
```

```
{
    adult += v.Visitor_Count;
    AdultCount.Text = adult.ToString();

    adultamount += v.Pay_amount;
    AdultTotalIncome.Text = adultamount.ToString();
}
else if (v.Visitor_Category == "Group of 5")
{
    group0f5 += v.Visitor_Count;
    Groupof5_Count.Text = group0f5.ToString();

    go5amount += v.Pay_amount;
    Go5TotalIncome.Text = go5amount.ToString();
}
else if (v.Visitor_Category == "Group of 10")
{
    group0f10 += v.Visitor_Count;
    Groupof10_Count.Text = group0f10.ToString();
```

```
            go10amount += v.Pay_amount;
            Go10TotalIncome.Text = go10amount.ToString();
        }

        int count = child + adult + group0f5 + group0f10;
        dailycount.Text = count.ToString();

        float income = childamount + adultamount + go5amount + go10amount;
        dailyincome.Text = income.ToString();


        if (child != 0)
        {
            ChildCount.Visible = true;
            ChildTotalIncome.Visible = true;
        }
        if (adult != 0)
        {
            AdultCount.Visible = true;
            AdultTotalIncome.Visible = true;
```

```
            }

            if (group0f5 != 0)
            {
                Groupof5_Count.Visible = true;
                Go5TotalIncome.Visible = true;
            }

            if (group0f10 != 0)
            {
                Groupof10_Count.Visible = true;
                Go10TotalIncome.Visible = true;
            }

            if (count != 0)
            {
                dailycount.Visible = true;
                dailyincome.Visible = true;
            }
        }
```

```
            generateChart();
        }

        private void generateChart()
        {

            DailyReportChart.Series["Count"].Points.AddXY("Child", child);
            DailyReportChart.Series["Count"].Points.AddXY("Adult", adult);
            DailyReportChart.Series["Count"].Points.AddXY("Group of 5", group0f5);
            DailyReportChart.Series["Count"].Points.AddXY("Group of 10", group0f10);

            DailyReportChart.Series["Income"].Points.AddXY("Child", childamount);
            DailyReportChart.Series["Income"].Points.AddXY("Adult", adultamount);
            DailyReportChart.Series["Income"].Points.AddXY("Group of 5", go5amount);
            DailyReportChart.Series["Income"].Points.AddXY("Group of 10",
go10amount);
        }

    }
}
```

Weekly report

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.IO;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;


namespace Coursework1
{
    public partial class WeeklyReport : Form
    {
        int count = 0;
        float amount = 0;
        readonly String filepath = "registered_visitors.csv";
        public WeeklyReport()
```

```
{
    InitializeComponent();
}

private void GetWeeklyReport_Click(object sender, EventArgs e)
{
    GlobalValues.RegisteredVisitorList = new List<Visitor>();
    string[] lines = File.ReadAllLines(filepath);
    foreach (string line in lines)
    {
        string[] st = line.Split(',');

        if (st.Length == 7)
        {
            Visitor v = new Visitor();
            v.Visitor_Id = Convert.ToInt32(st[0]);
            v.Visitor_Category = st[1];
            v.Visitor_Count = Convert.ToInt32(st[2]);
            v.Entry_Time = Convert.ToDateTime(st[3]);
            v.Exit_Time = Convert.ToDateTime(st[4]);
```

```
            v.Stay_time = TimeSpan.Parse(st[5]);
            v.Pay_amount = float.Parse(st[6]);

            GlobalValues.RegisteredVisitorList.Add(v);
        }
    }

    for (int i = 0; i < 7; i++)
    {
        count = 0;
        amount = 0;
        DateTime dateT = (Convert.ToDateTime(DateofReport.Text)).Date;
        dateT = dateT.AddDays(-i);
        VisitorReport vr = new VisitorReport();
        foreach (Visitor v in GlobalValues.RegisteredVisitorList)
        {
            if (v.Entry_Time.DayOfYear == dateT.DayOfYear)
            {
                count += v.Visitor_Count;
                vr.DailyVisitorCountTotal = count;
```

```
                    vr.Date = dateT;
                    amount += v.Pay_amount;
                    vr.DailyIncomeTotal = amount;
                }
            }
            GlobalValues.ReportList.Add(vr);

        }
        WeeklyReportView.DataSource = GlobalValues.ReportList;
        foreach (VisitorReport vr in GlobalValues.ReportList)
        {
            if (vr.DailyIncomeTotal != 0 && vr.DailyVisitorCountTotal != 0)
            {
                weeklychart.Series["Count"].Points.AddXY(vr.Date.ToString("dddd"),
vr.DailyVisitorCountTotal);
                weeklychart.Series["Income"].Points.AddXY(vr.Date.ToString("dddd"),
vr.DailyIncomeTotal);
            }
        }

    }
```

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



        }
    }
}
```