Module Code & Module Title

CS6004NT Application Development

Assessment Weightage & Type

30% Individual

Year and Semester

2021 Autumn

Student Name: Samuel Sherpa

London Met ID: 19031860

College ID: np05cp4a190148@iic.edu.np

Assignment Due Date: 3rd Jan 2021

Assignment Submission Date: 3rd Jan 2021

Word Count: 3500

I confirm that I understand my coursework needs to be submitted online via Google Classroom under the relevant module page before the deadline in order for my assignment to be accepted and marked. I am fully aware that late submissions will be treated as non-submission and a mark of zero will be awarded.

Acknowledgment

First of all, I would like to open my heart and life in a fresh way where I surrender to acknowledge God the almighty in all my ways that his hand is upon me because of which I am able to study in this prestigious college and acquire to get more knowledge and skills.

I would like to express my special thanks of gratitude to my module lecturer MR. Ravi Rouniyar for his guidance and support in completing my coursework. His calm nature and interesting subject-related conversations led me to ignite the curiosity of doing cool stuff with Visual Studio which helped me a lot to know more about the module and coursework.

I would like to acknowledge my parents and sister for continuous support in a home by providing me with a peaceful environment where I could study very well. Last but not least I would like to thank my classmates who have helped me a lot.

Thank you all from the bottom of my heart. God bless you all.

Abstract

Facebook wasn't the first thing Mark Zuckerberg created but he also build many chat systems, games, study tools, and music players before. J.K. Rowling was also rejected 12 times before publishing the legendary book 'Harry Potter'. Even Beyonce had to make hundreds of songs before creating 'Halo'. Similarly, this CW is a tiny step of my career to progress and grow in the IT field.

Building a desktop application using the C# .net framework was a challenging task though being a third-year student. This CW documentation is an individual first CW of Application development carrying 30% module marks. In this documentation, you would find a very well explained instruction to use the build system 'Damak Recreation Center System', class diagrams of build classes, explanation of methods present in respective classes, system architecture, and flow chart of the system, explanation of used algorithms and data structure, a reflection of the experience and heartfelt conclusion.

Table of Contents

Table of figure

Table of table

1. Introduction

The CourseWork(CW) deals with developing a desktop application using C# with an object-oriented approach and uses Visual Studio 2019 IDE. The build application," Damak Recreation Centre System" was a system that could be used by admin or staff. The admin has to enter a password, "admin" to use the system as admin whereas staff has to enter the password, "staff" to use the system as being staff.

In the system, the admin has the privilege of the editing ticket price rate which would be used by the system to calculate the total ticket fee of a visitor. On the other hand, staff has the privilege of adding visitors in the system with their details in the CSV file which would be shown in the datagridview table directly. Staff can also generate a daily report based on category and total visitors and a weekly report based on day, total visitors, and total earning in bar graph form.

This documentation file covers different topics to explain the coding portion of the CW simply to the reader. The topics discussed here are :

i) Detailed instruction to use the application
ii) System architecture
iii) Class Diagram
iv) Method Description
v) Flow Chart of the program
vi) Use Data Structure and algorithm
vii) Reflection of the learning experience
viii) Conclusion.

2. Detailed Instruction to run the program

Instruction to use the system is given below which would help users to use the system easily. Since the system can be used either being admin or staff, the instruction is given from two users' perspectives.

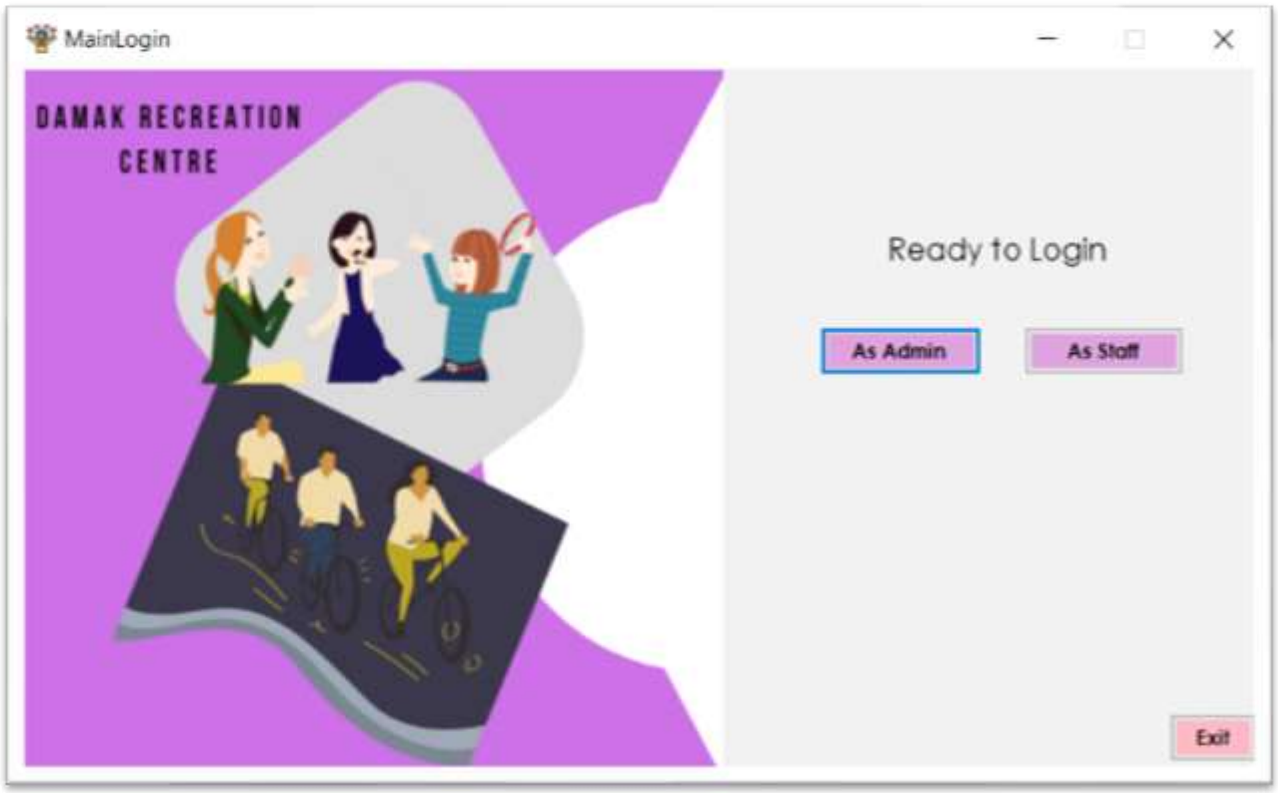


Figure 1 Opening page of ' Damak Recreation Centre'.

From the above diagram, we can see that users can use the system either being the admin or being a staff. Let us see, how we can use the system from the below guidelines.

2.1. Using 'Damak Recreation Center System' as Admin

2.1.1. Logging to the system as admin

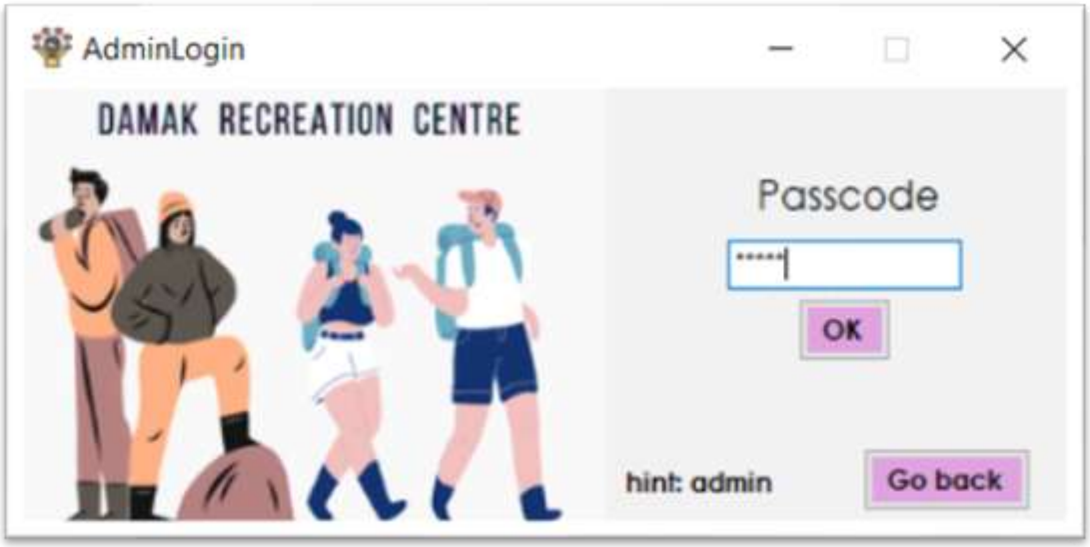
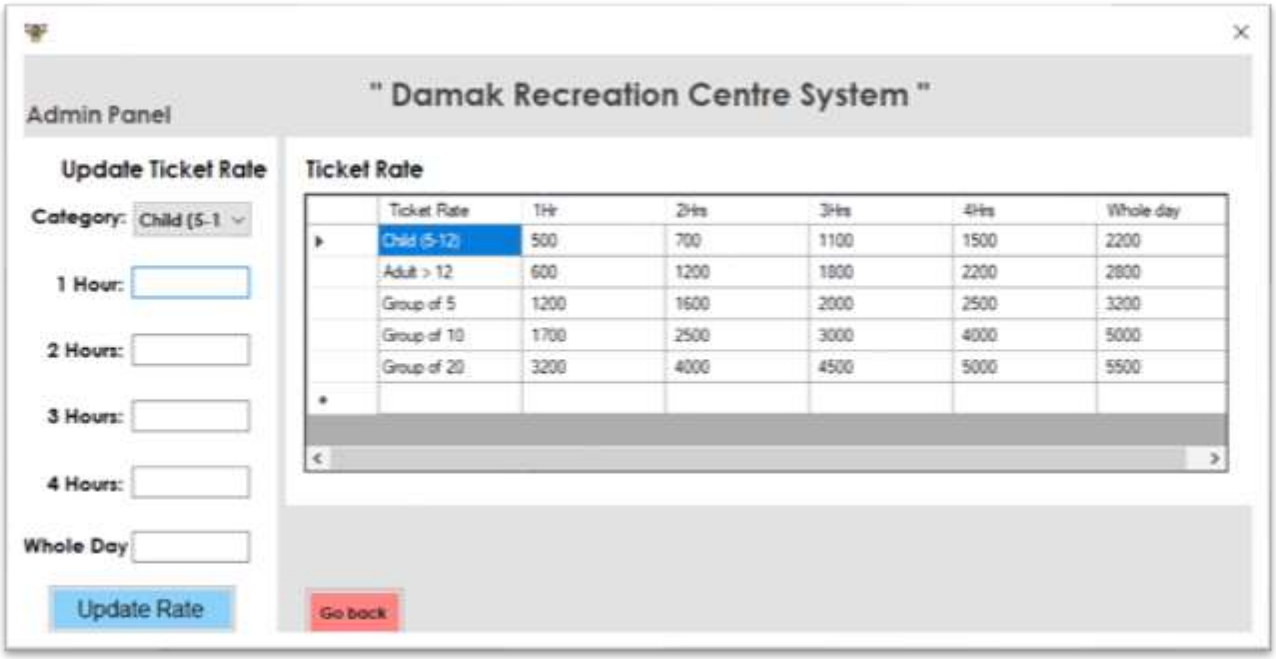


Figure 2 Entering a password to login as admin.

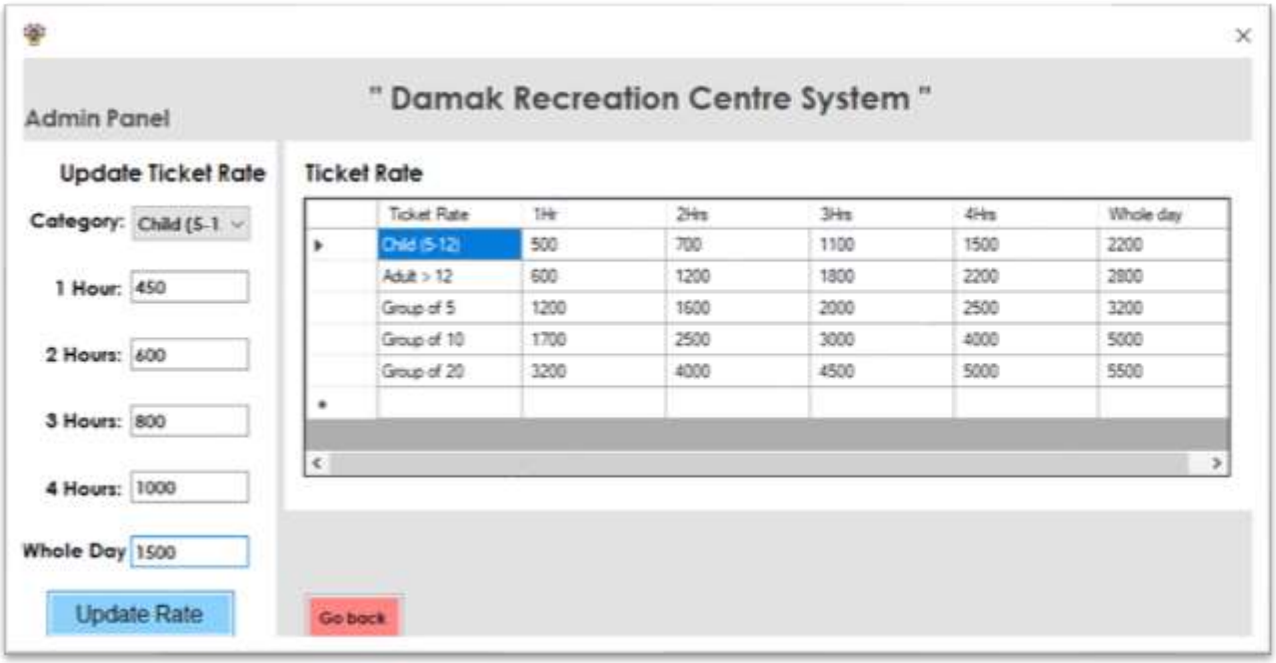
After clicking the admin button, the above panel is opened and if the correct password is entered below the panel is opened.



Ticket Rate	1Hr	2Hrs	3Hrs	4Hrs	Whole day
Child (5-12)	500	700	1100	1500	2200
Adult > 12	600	1200	1800	2200	2800
Group of 5	1200	1600	2000	2500	3200
Group of 10	1700	2500	3000	4000	5000
Group of 20	3200	4000	4500	5000	5500

Figure 3 Entering a correct password to get to Admin Panel aft

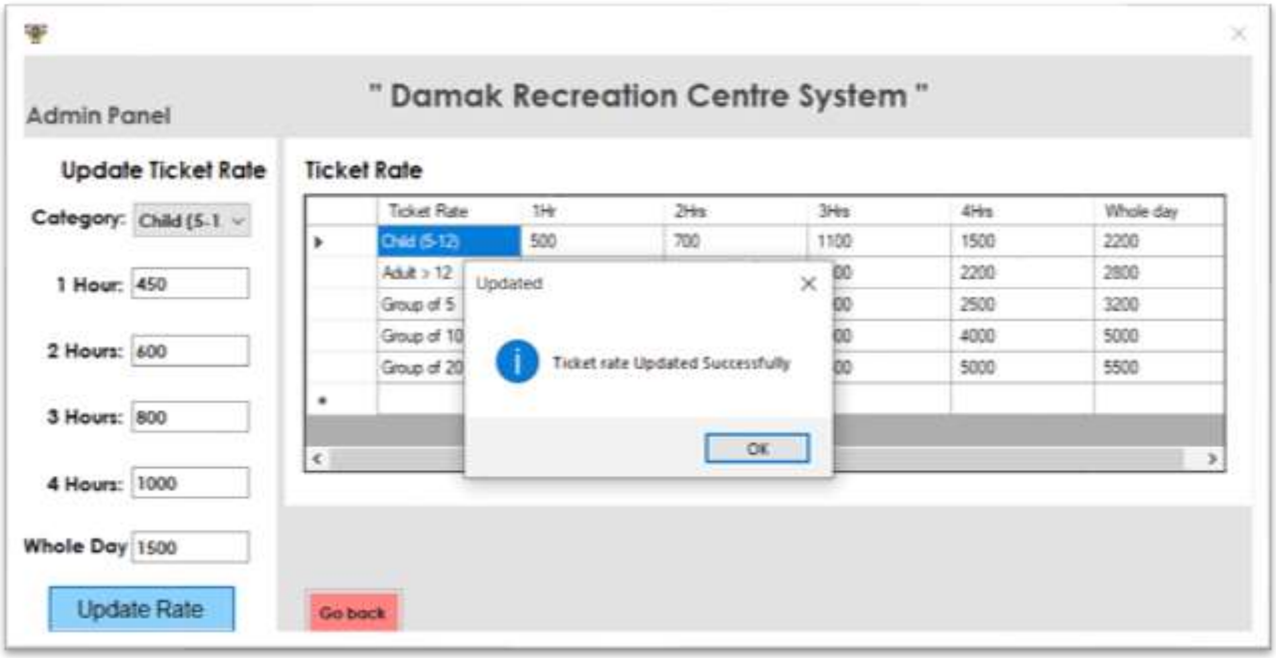
2.1.2. Editing & UpdateTicket Price Rate



Ticket Rate	1Hr	2Hrs	3Hrs	4Hrs	Whole day
Child (5-12)	500	700	1100	1500	2200
Adult > 12	600	1200	1800	2200	2800
Group of 5	1200	1600	2000	2500	3200
Group of 10	1700	2500	3000	4000	5000
Group of 20	3200	4000	4500	5000	5500

Figure 4 Insert new ticket price to change the ticket rate.

In the above figure, how to edit the selected row is shown.



Ticket rate Updated Successfully

Figure 5 Updating the ticket rate table after validating the entered price.

2.2. Logging in to the system as staff

2.2.1. Staff login

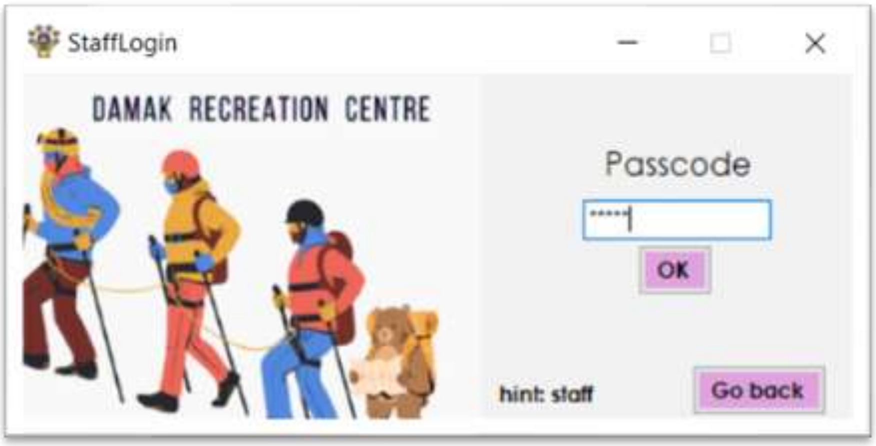
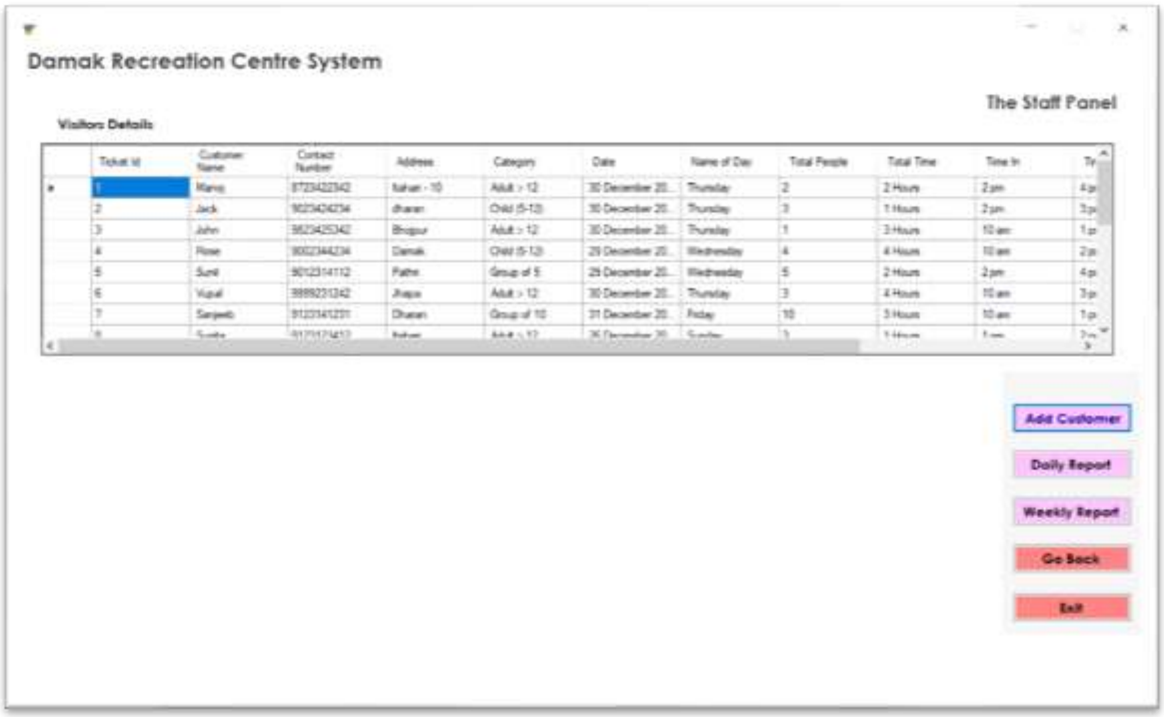


Figure 6 Login to the System as a staff

After clicking the staff button, the above panel is shown.



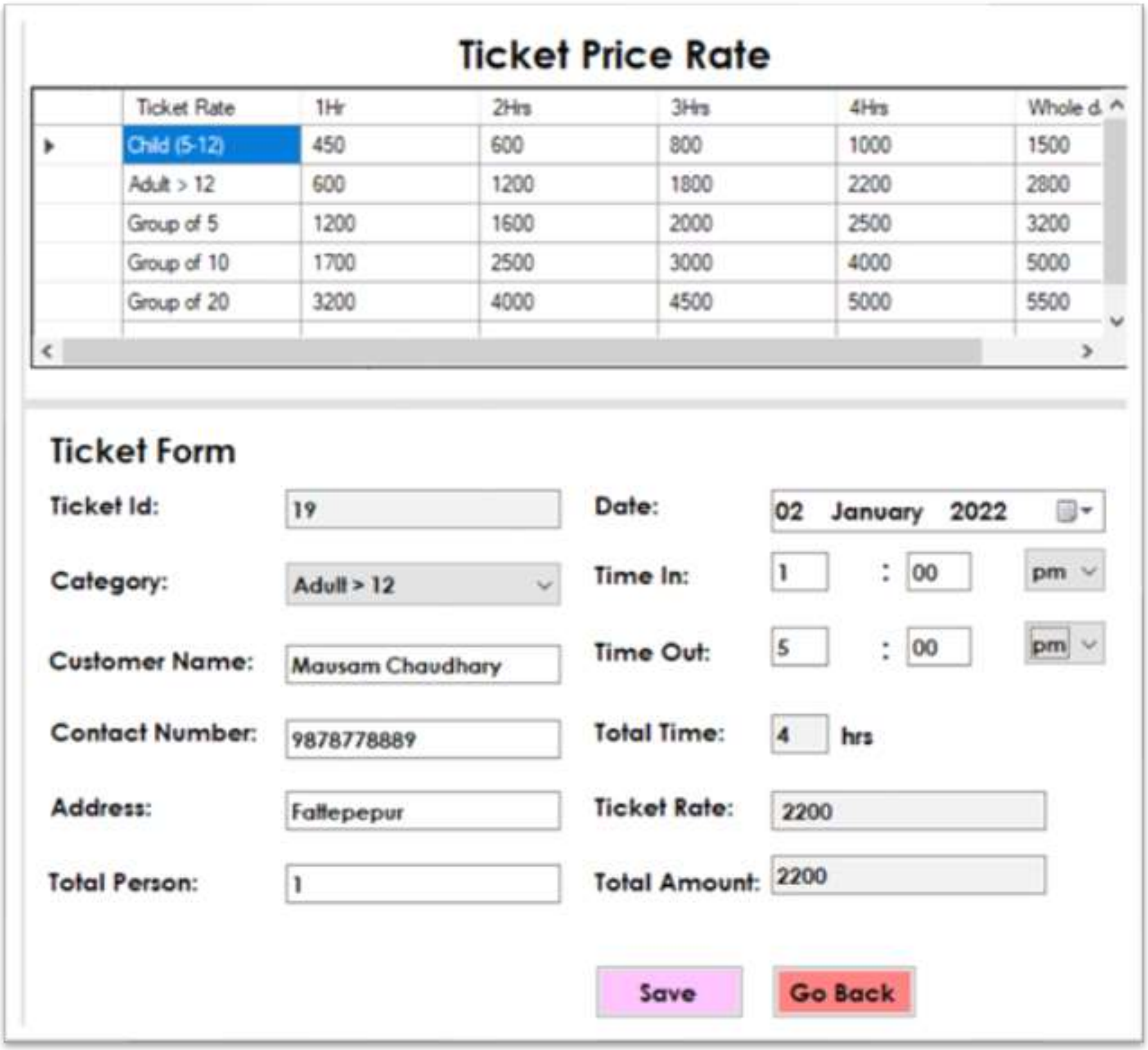
Visitors Details

Ticket Id	Customer Name	Contact Number	Address	Category	Date	Name of Day	Total People	Total Time	Time In	Ti
1	Manoj	9723422342	Itahari - 10	Adult > 12	30 December 20...	Thursday	2	2 Hours	2 pm	4 p
2	Jack	9023424234	dharan	Child (5-12)	30 December 20...	Thursday	3	1 Hours	2 pm	3 p
3	John	9823425342	Bhojpur	Adult > 12	30 December 20...	Thursday	1	3 Hours	10 am	1 p
4	Rose	9002344234	Damak	Child (5-12)	29 December 20...	Wednesday	4	4 Hours	10 am	2 p
5	Sunil	9012314112	Pathri	Group of 5	29 December 20...	Wednesday	5	2 Hours	2 pm	4 p
6	Vupal	9889231242	Jhapa	Adult > 12	30 December 20...	Thursday	3	4 Hours	10 am	3 p
7	Sanjeeb	9123141231	Dharan	Group of 10	31 December 20...	Friday	10	3 Hours	10 am	1 p
8	Sunita	9121173412	Itahari	Adult > 12	26 December 20...	Sunday	1	1 Hours	1 pm	2 p

Figure 7 After entering the correct password, the above dashboard is seen by staff.

After entering the correct password, the above panel is opened.

2.2.2. Add Customer



Ticket Price Rate

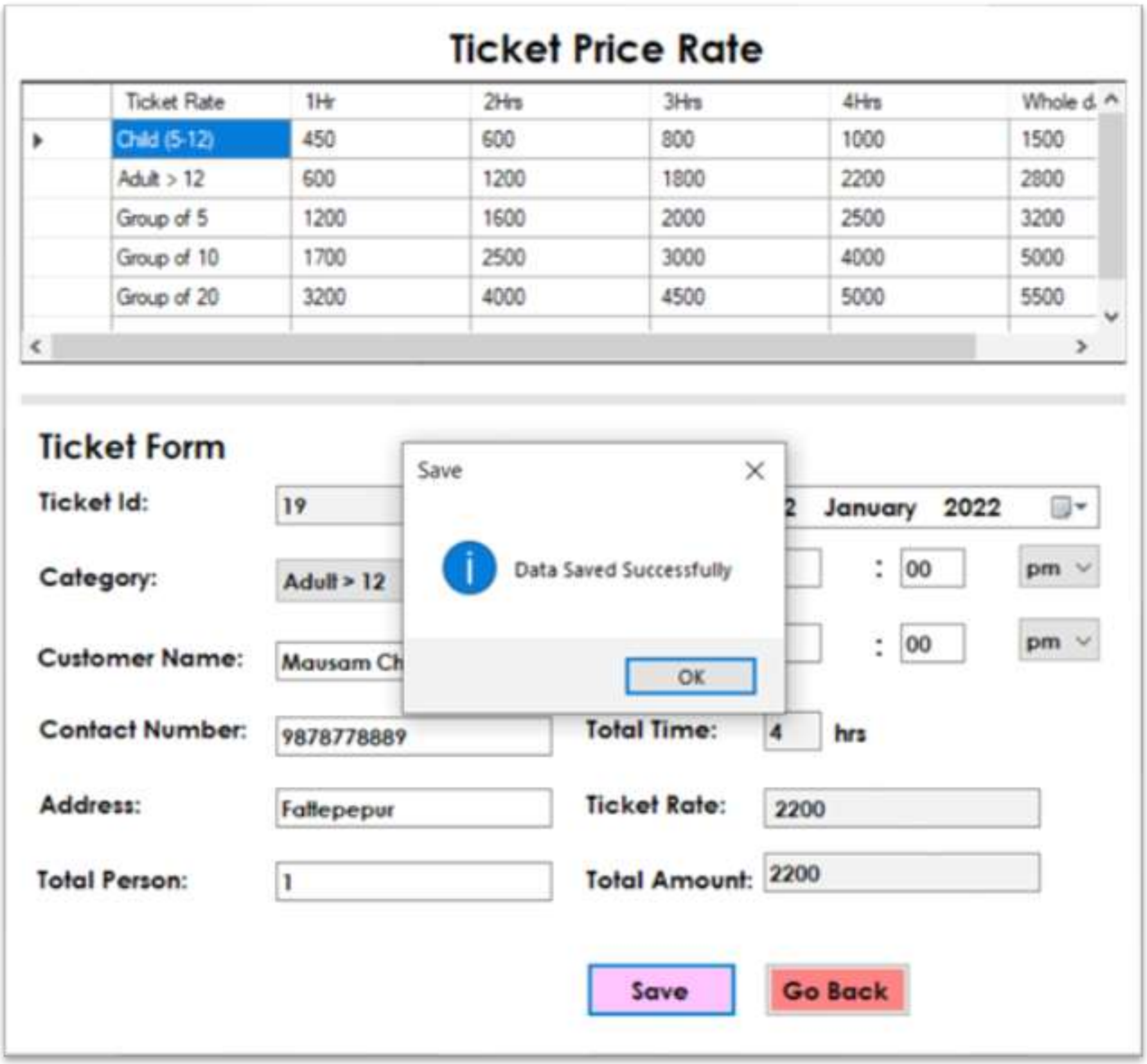
	Ticket Rate	1Hr	2Hrs	3Hrs	4Hrs	Whole d
▸	Child (5-12)	450	600	800	1000	1500
	Adult > 12	600	1200	1800	2200	2800
	Group of 5	1200	1600	2000	2500	3200
	Group of 10	1700	2500	3000	4000	5000
	Group of 20	3200	4000	4500	5000	5500

Ticket Form

Ticket Id:	19	Date:	02 January 2022
Category:	Adult > 12	Time In:	1 : 00 pm
Customer Name:	Mausam Chaudhary	Time Out:	5 : 00 pm
Contact Number:	9878778889	Total Time:	4 hrs
Address:	Faltepepur	Ticket Rate:	2200
Total Person:	1	Total Amount:	2200

Figure 8 Adding a new visitor to the system.

After clicking the 'Add Customer' button, the above panel is opened. In the above panel, we could add visitors' detail in the CSV file.



Ticket Price Rate

	Ticket Rate	1Hr	2Hrs	3Hrs	4Hrs	Whole d
▶	Child (5-12)	450	600	800	1000	1500
	Adult > 12	600	1200	1800	2200	2800
	Group of 5	1200	1600	2000	2500	3200
	Group of 10	1700	2500	3000	4000	5000
	Group of 20	3200	4000	4500	5000	5500

Ticket Form

Ticket Id:	19
Category:	Adult > 12
Customer Name:	Mausam Ch
Contact Number:	9878778889
Address:	Fattepepur
Total Person:	1
Total Time:	4 hrs
Ticket Rate:	2200
Total Amount:	2200

2 January 2022

: 00 pm

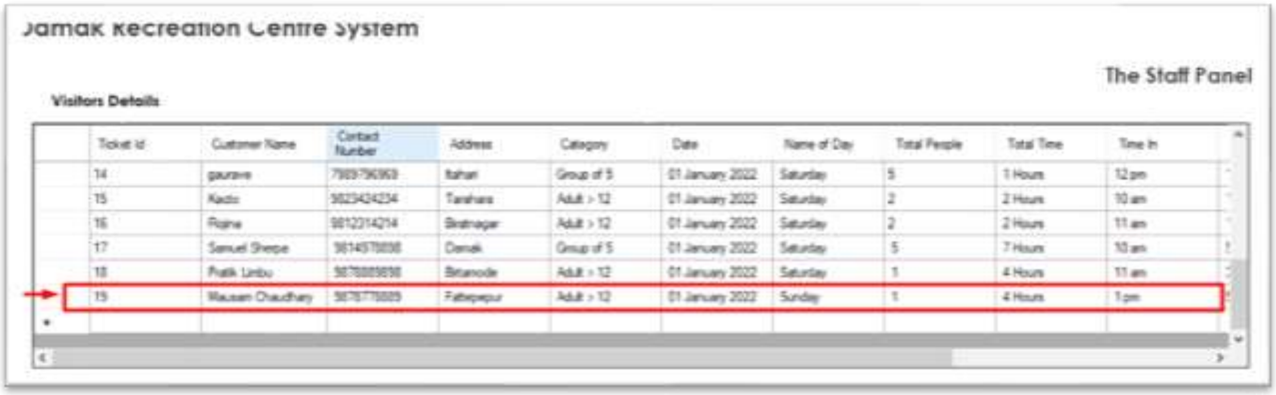
: 00 pm

Save

Data Saved Successfully

Figure 9 Saving the updated field to the system.

The detail of the visitor is saved successfully.


Jamak Recreation Centre System

The Staff Panel

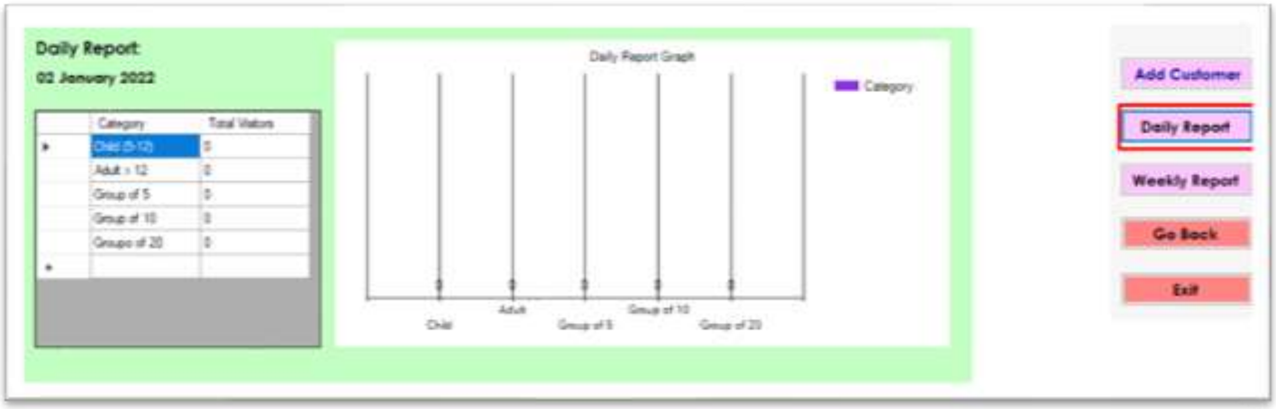
Visitors Details

Ticket Id	Customer Name	Contact Number	Address	Category	Date	Name of Day	Total People	Total Time	Time In
14	gaurave	7989796969	Itahari	Group of 5	01 January 2022	Saturday	5	1 Hours	12 pm
15	Kaclo	9823424234	Tarahara	Adult > 12	01 January 2022	Saturday	2	2 Hours	10 am
16	Rojina	9812314214	Biratnagar	Adult > 12	01 January 2022	Saturday	2	2 Hours	11 am
17	Samuel Sherpa	9814578898	Damak	Group of 5	01 January 2022	Saturday	5	7 Hours	10 am
18	Pratik Limbu	9878889898	Birtamode	Adult > 12	01 January 2022	Saturday	1	4 Hours	11 am
19	Mausam Chaudhary	9878778889	Fattepepur	Adult > 12	01 January 2022	Sunday	1	4 Hours	1 pm

Figure 10 The added visitor is shown in the visitor's detail table.

In the above figure, it's clear that the above-added visitor's details are saved in the CSV file successfully and who in the datagridview.

2.2.3. Generate Daily Report


Daily Report

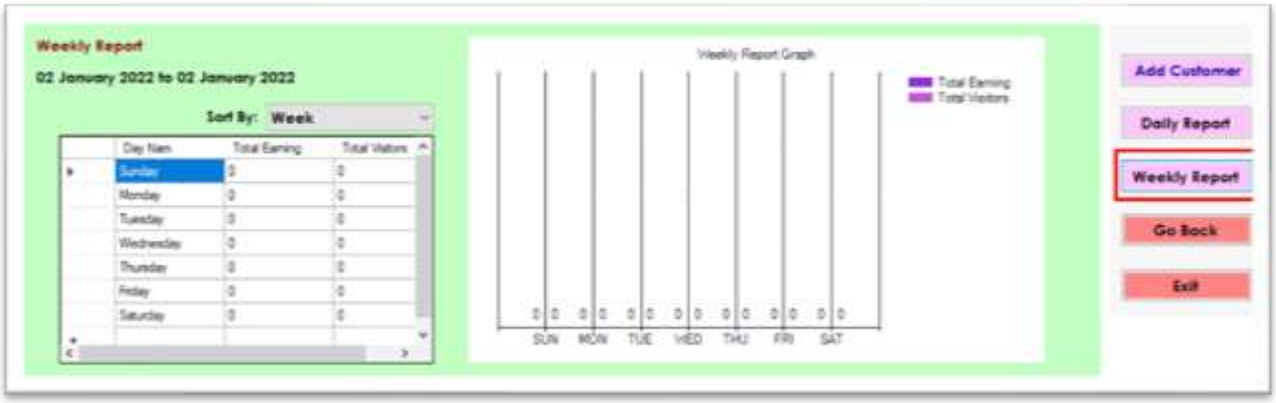
02 January 2022

Category	Total Visitors
Child (5-12)	0
Adult > 12	0
Group of 5	0
Group of 10	0
Groups of 20	0

Figure 11 Daily report has been generated

The daily report can be generated by clicking the 'Daily Report' button.

2.2.4. Generate Weekly Report



Weekly Report

02 January 2022 to 02 January 2022

Sort By: Week

Day Nam	Total Earning	Total Visitors
Sunday	0	0
Monday	0	0
Tuesday	0	0
Wednesday	0	0
Thursday	0	0
Friday	0	0
Saturday	0	0

Figure 12 Weekly report has been generated.

A weekly report can be generated by clicking the weekly report button.

Simple but effective information for the user

If you want to close the program then just click the exit button of the application page.

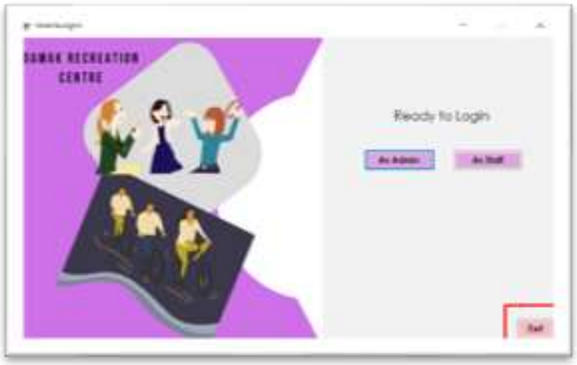


Figure 13 Exiting the system.

3. System Architecture

The architectural diagram of a system helps to abstract the software system's overall outline and build constraints, and relations with boundaries between components. It provides physical deployment of the evolution roadmap of any software system. is a conceptual model which defines the structure, behavior, and view of the system (wondershare_SystemArchitecture, 2021).

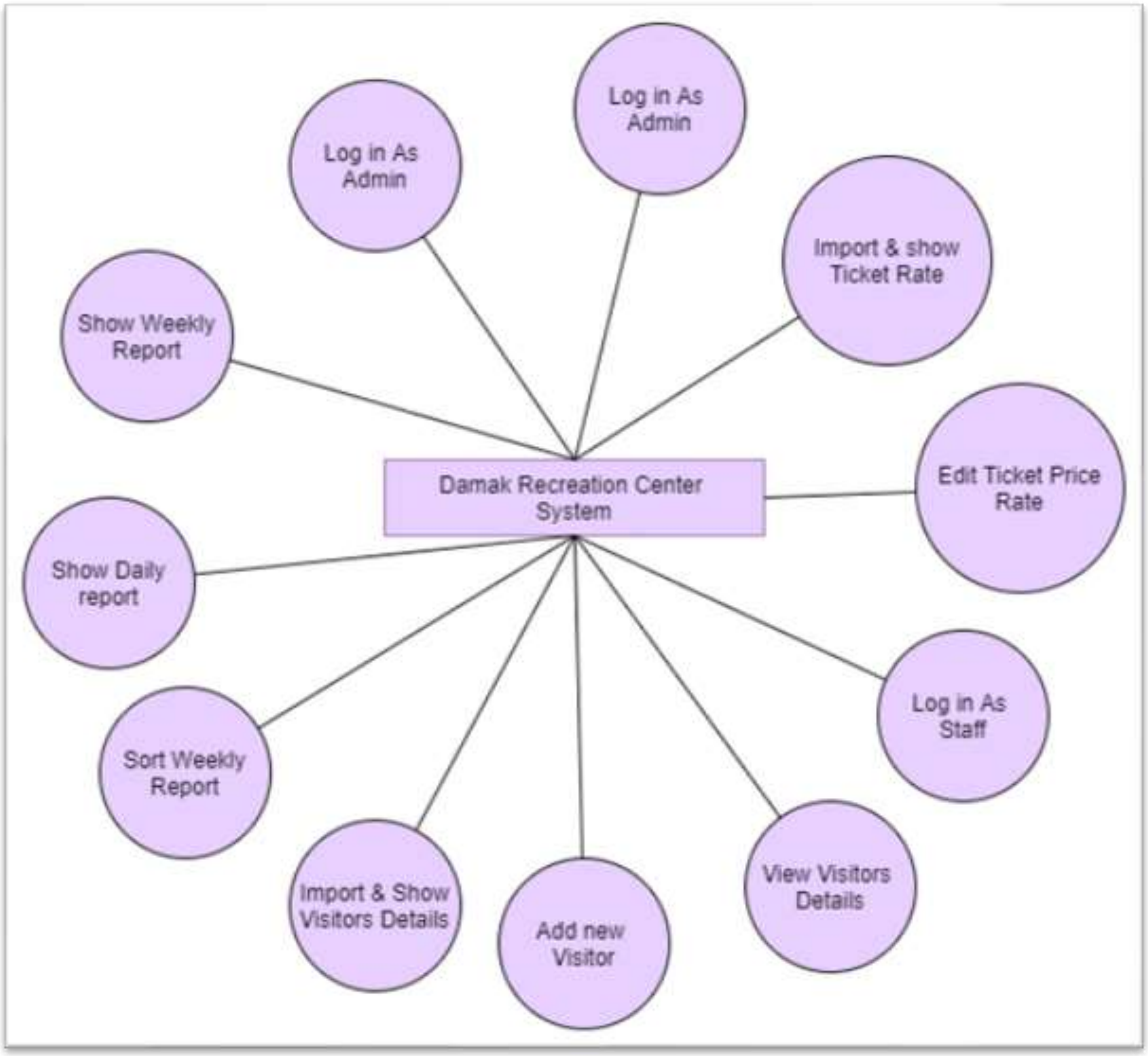


Figure 14 'Damak Recreation Center System' architecture

4. Class Diagram

A class diagram is a type of static structure diagram which describes the structure of a system by showing the system's classes, their attributes, operations, and relationships among different objects (VisualParadigm_ClassDiagram, 2021). The class diagrams of different classes of 'Damak Recreation Center System' are as follows:

4.1. MainLogin class diagram

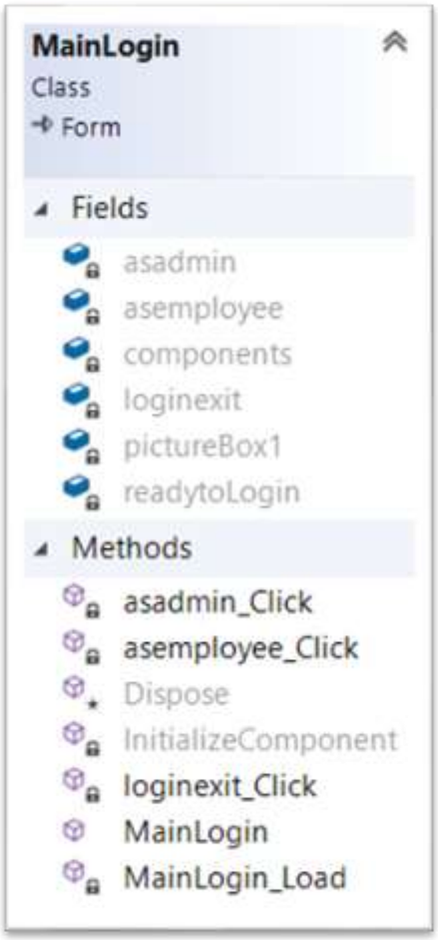

MainLogin
Class
→ Form
Fields
asadmin
asemployee
components
loginexit
pictureBox1
readytoLogin
Methods
asadmin_Click
asemployee_Click
Dispose
InitializeComponent
loginexit_Click
MainLogin
MainLogin_Load

Figure 15 MainLogin Classdiagram

4.2. AdminLogin Class Diagram

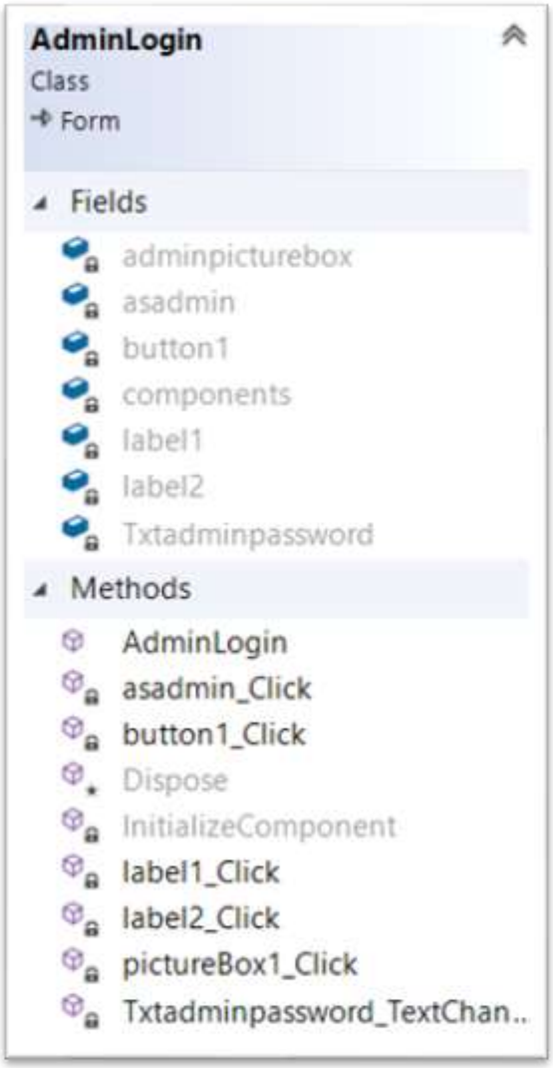
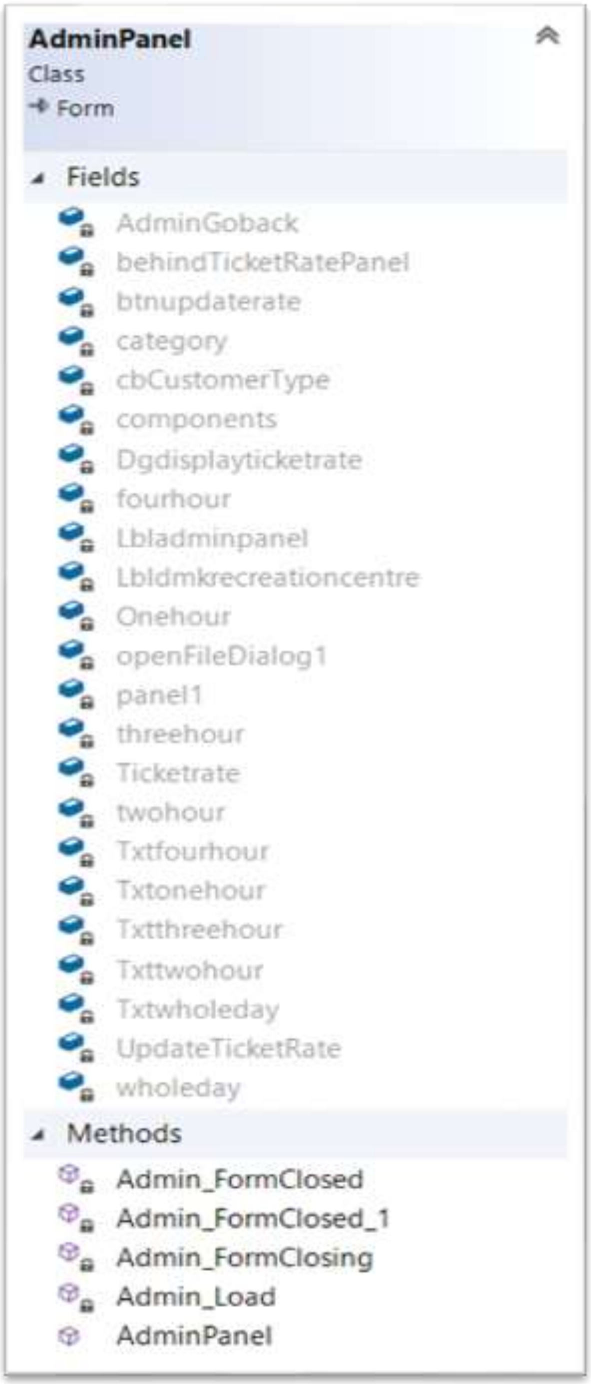


Figure 16 AdminLogin Classdiagram

4.3. AdminPanel Class Diagram



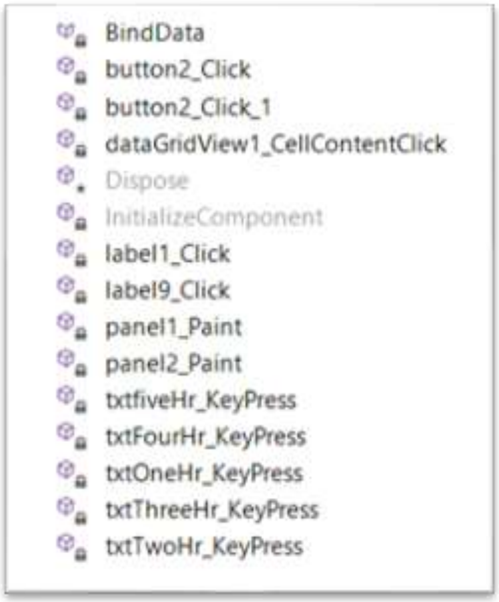
AdminPanel
Class
→ Form

Fields

AdminGoback
behindTicketRatePanel
btnupdaterate
category
cbCustomerType
components
Dgdisplayticketrate
fourhour
Lbladminpanel
Lbldmkrecreationcentre
Onehour
openFileDialog1
panel1
threehour
Ticketrate
twohour
Txtfourhour
Txtonehour
Txtthreehour
Txttwohour
Txtwholeday
UpdateTicketRate
wholeday

Methods

Admin_FormClosed
Admin_FormClosed_1
Admin_FormClosing
Admin_Load
AdminPanel



BindData
button2_Click
button2_Click_1
dataGridView1_CellContentClick
Dispose
InitializeComponent
label1_Click
label9_Click
panel1_Paint
panel2_Paint
txtfiveHr_KeyPress
txtFourHr_KeyPress
txtOneHr_KeyPress
txtThreeHr_KeyPress
txtTwoHr_KeyPress

Figure 17 Admin Panel Classdiagram

Figure 18 Admin Panel Classdiagram

4.4. StaffLogin Class Diagram

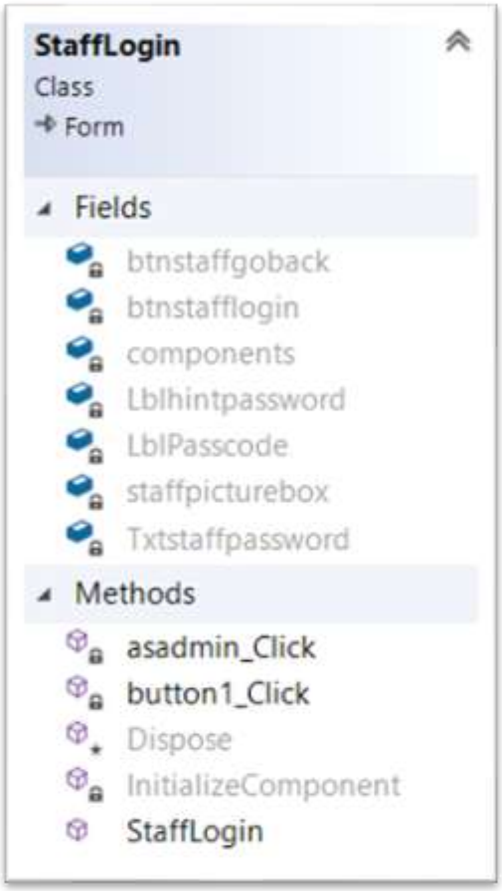
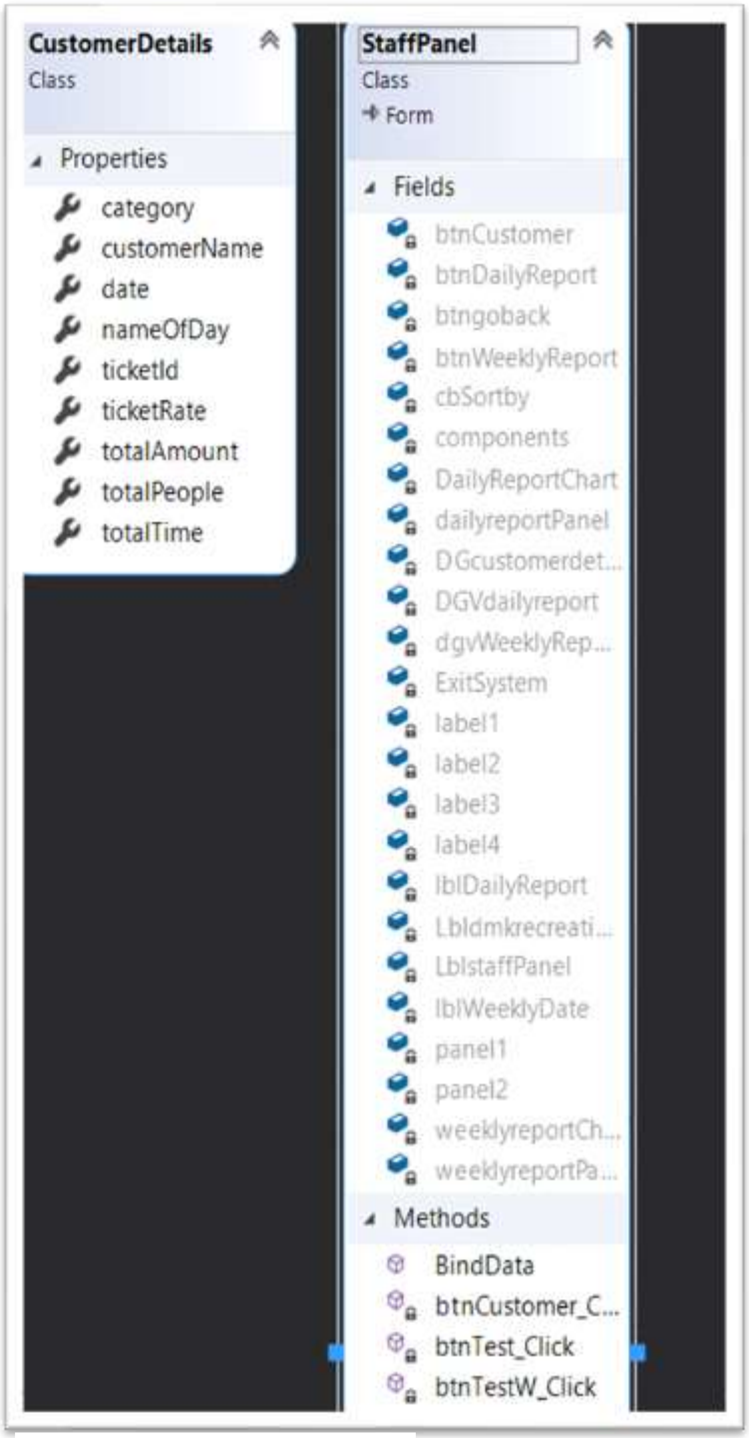


Figure 19 StaffLogin Classdiagram

4.5. StaffPanelClassdiagram



CustomerDetails
Class

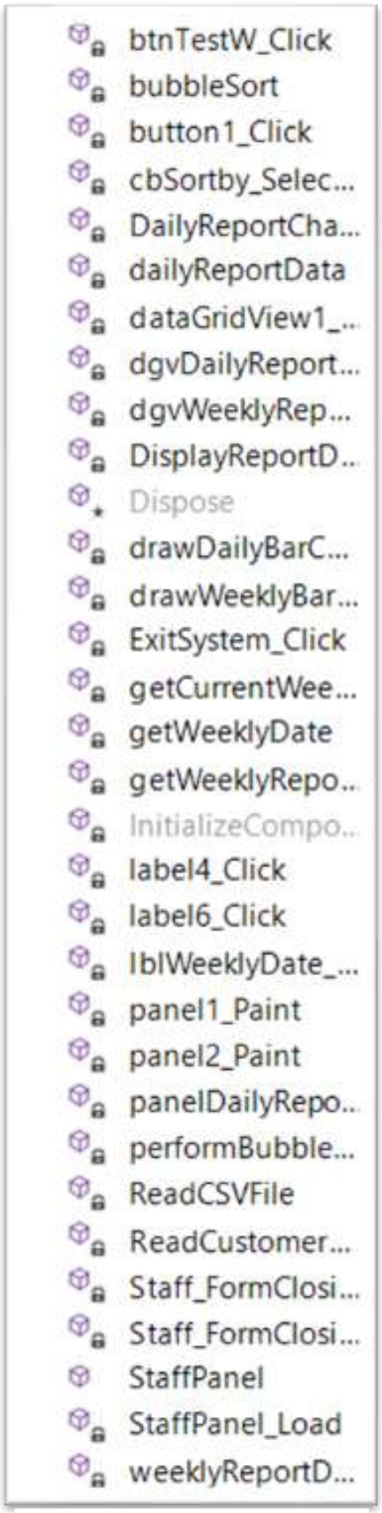
Properties
- category
- customerName
- date
- nameOfDay
- ticketId
- ticketRate
- totalAmount
- totalPeople
- totalTime

StaffPanel
Class
→ Form

Fields
- btnCustomer
- btnDailyReport
- btngoback
- btnWeeklyReport
- cbSortby
- components
- DailyReportChart
- dailyreportPanel
- DGcustomerdet...
- DGVdailyreport
- dgvWeeklyRep...
- ExitSystem
- label1
- label2
- label3
- label4
- lblDailyReport
- Lbldmkrecreati...
- LblstaffPanel
- lblWeeklyDate
- panel1
- panel2
- weeklyreportCh...
- weeklyreportPa...

Methods
- BindData
- btnCustomer_C...
- btnTest_Click
- btnTestW_Click

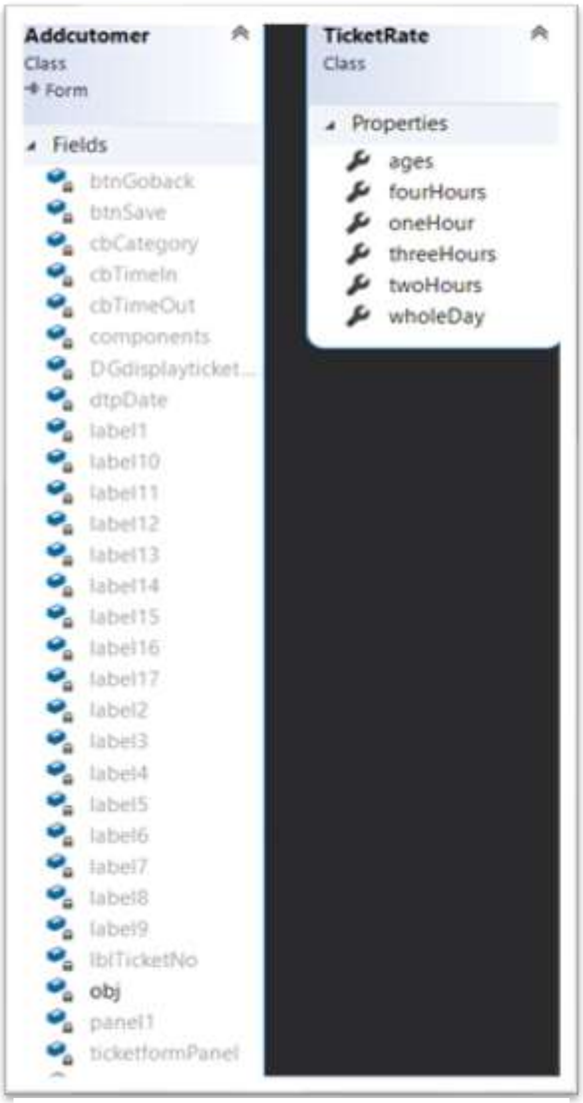
Figure 21 StaffPanel Class diagram



- btnTestW_Click
- bubbleSort
- button1_Click
- cbSortby_Selec...
- DailyReportCha...
- dailyReportData
- dataGridView1_...
- dgvDailyReport...
- dgvWeeklyRep...
- DisplayReportD...
- Dispose
- drawDailyBarC...
- drawWeeklyBar...
- ExitSystem_Click
- getCurrentWee...
- getWeeklyDate
- getWeeklyRepo...
- InitializeCompo...
- label4_Click
- label6_Click
- lblWeeklyDate_...
- panel1_Paint
- panel2_Paint
- panelDailyRepo...
- performBubble...
- ReadCSVFile
- ReadCustomer...
- Staff_FormClosi...
- Staff_FormClosi...
- StaffPanel
- StaffPanel_Load
- weeklyReportD...

Figure 20 StaffPanel Class Diagram continuation

4.6. Add visitors

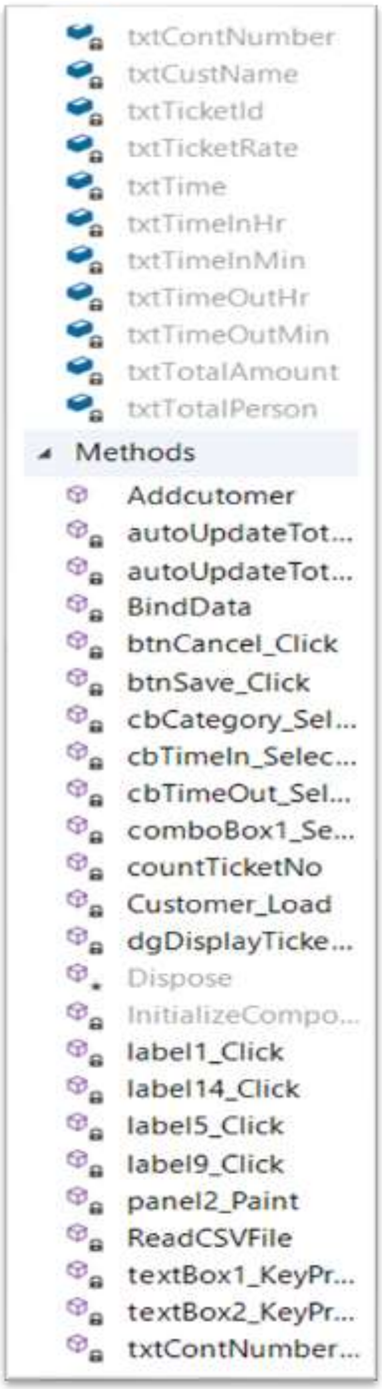

Addcutomer
Class
→ Form

Fields
btnGoback
btnSave
cbCategory
cbTimeIn
cbTimeOut
components
DGdisplayticket...
dtpDate
label1
label10
label11
label12
label13
label14
label15
label16
label17
label2
label3
label4
label5
label6
label7
label8
label9
lblTicketNo
obj
panel1
ticketformPanel

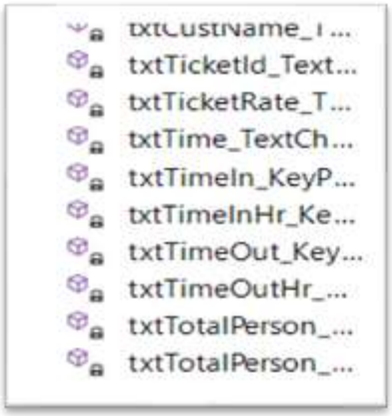
TicketRate
Class

Properties
ages
fourHours
oneHour
threeHours
twoHours
wholeDay

Figure 22 Add Visitors Class diagram


txtContNumber
txtCustName
txtTicketId
txtTicketRate
txtTime
txtTimeInHr
txtTimeInMin
txtTimeOutHr
txtTimeOutMin
txtTotalAmount
txtTotalPerson

Methods
Addcutomer
autoUpdateTot...
autoUpdateTot...
BindData
btnCancel_Click
btnSave_Click
cbCategory_Sel...
cbTimeIn_Selec...
cbTimeOut_Sel...
comboBox1_Se...
countTicketNo
Customer_Load
dgDisplayTicke...
Dispose
InitializeCompo...
label1_Click
label14_Click
label5_Click
label9_Click
panel2_Paint
ReadCSVFile
textBox1_KeyPr...
textBox2_KeyPr...
txtContNumber...


txtCustName_T...
txtTicketId_Text...
txtTicketRate_T...
txtTime_TextCh...
txtTimeIn_KeyP...
txtTimeInHr_Ke...
txtTimeOut_Key...
txtTimeOutHr_...
txtTotalPerson_...
txtTotalPerson_...

5. Method Description

The different classes are used in the system.

Method description of MainLogin class.

S.N.	Method Names	IDE/Self made	Description
1	`private void loginexit_Click(object sender, EventArgs e)`	IDE made	Priavte void method which closes the program.
2	`private void asadmin_Click(object sender, EventArgs e)`	IDE made	Private void method which opens adminlogin form.
3	`private void asemployee_Click(object sender, EventArgs e)`	IDE made	Private void method which opens stafflogin form.

Table 1 Method description of MainLogin Class.

Method description of AdminLogin class.

S.N.	Method Names	IDE/Self made	Description
1	`private void button1_Click(object sender, EventArgs e)`	IDE made	Private void method which takes user back to mainlogin page.
2	`private void asadmin_Click(object sender, EventArgs e)`	IDE made	Private void method which takes user to adminpanel form.

Table 2 Method description of AdminLogin class.

Method description of AdminPanel class.

S.N.	Method Names	IDE/Self made	Description
1	`private void BindData(String filePath)`	Self made.	Private void method which imports .csv file data to datagridview table.
2	`private void saveUpdatedTicketprice(object sender, EventArgs e)`	Self made.	Private void method which saves updated ticket price from admin.
3	`private void txtfiveHr_KeyPress(object sender, KeyPressEventArgs e)`	IDE made	Private void method which doesn't allow staff to insert non number keys.

Table 3 Method description of AdminPanel Class.

Method description of StaffLogin class.

S.N.	Method Names	IDE/Self made	Description
1	`private void asadmin_Click(object sender, EventArgs e)`	IDE made	Private void method which validates staff password.
2	`private void button1_Click(object sender, EventArgs e)`	IDE made	Private void method which takes user back to Mainlogin form.

Table 4 Method description of StaffLogin class.

Method description of StaffPanel class.

S.N.	Method Names	IDE/Self made	Description
1	`private void btnCustomer_Click(object sender, EventArgs e)`	Self made	Private void method which allows staff to add customer to the system.
2	`public void BindData(String filePath)`	Self-made	Public void method which helps to display .csv data to datagridview table.
3	`private int[] SystemBubbleSort(int[] ary)`	Self made	Private int method which sorts given array.
4	`private void performBubbleSort(string category)`	Selfmade	Private void method which sorts list on the basis of give category.
5	`private void displayReportDataInaTable(string[] weeklyData, string tableType)`	Selfmade	Private void method which takes weekly data and table type to display daily report
6	`private string getCurrentWeekFirstDate()`	Selfmade	Priavte string method which gets current date.
7	`private void generateWeeklyReport(object sender, EventArgs e)`	Selfmade	Private void method which generates weekly report.
8	`private int[] ReadCustomerCSVFile(string fileName)`	Selfmade	Private void method which reads customer.csv file.
9	`private void displayDailyReport(object sender, EventArgs e)`	Selfmade	Private void method which displays daily report.
10	`private string[] getdailyReportData()`	Selfmade	Private String method which gets report data from customer.csv file to show it in datagridview table.
11	`private string[] getWeeklyReportDataforgraph()`	Selfmade	Private String method which shows the weekly report in datagridview table.

12	`private void cbSortby_SelectionChangeCommitted(object sender, EventArgs e)`	Selfmade	Private void method which sorts weekly datagrid table on the basis of total earning and name of the day.
13	`private void drawWeeklyBarChart(int sunEarn, int sunVisit, int monEarn, int monVisit, int tueEarn, int tueVisit, int wedEarn, int wedVisit, int thuEarn, int thuVisit, int friEarn, int friVisit, int satEarn, int satVisit)`	Selfmade	Private void method which draws weekly report in bar graph.

Table 5 Method description of Staff Panel.

Method description of AddVisitors class.

S.N.	Method Names	IDE/Self made	Description
1	`private void BindData(String filePath)`	Self made	Private void method which reads the data from CSV file to display in datagridview table.
2	`private void updateTotalhourspend()`	Self made	A private void method that calculates total hour spent by visitor.
3	`private void btnSaveVisitorsDetails(object sender, EventArgs e)`	Self made	Priavte void method which saves visitors details in CSV file.
4	`private void countTicketidNo(string fileName)`	Self made	Private void method which generates unique ticket id for customer automatically.
5	`private void txtTotalPerson_KeyUp(object sender, KeyEventArgs e)`	Self made	Private void method which calculates ticket amount automatically on the basis of total number of visitors.
6	`private void autoCalculateTotalAmount()`	Self made	Private void method which shows calculated ticket price amount in a textbox
7	`private void cbCategory_SelectionChangeCommitted(object sender, EventArgs e)`	Self made	Private void method which takes data from select combo box item to calculate price of their number of enrollment.

Table 6 Method description of AddVisitors class.

6. Flow Chart

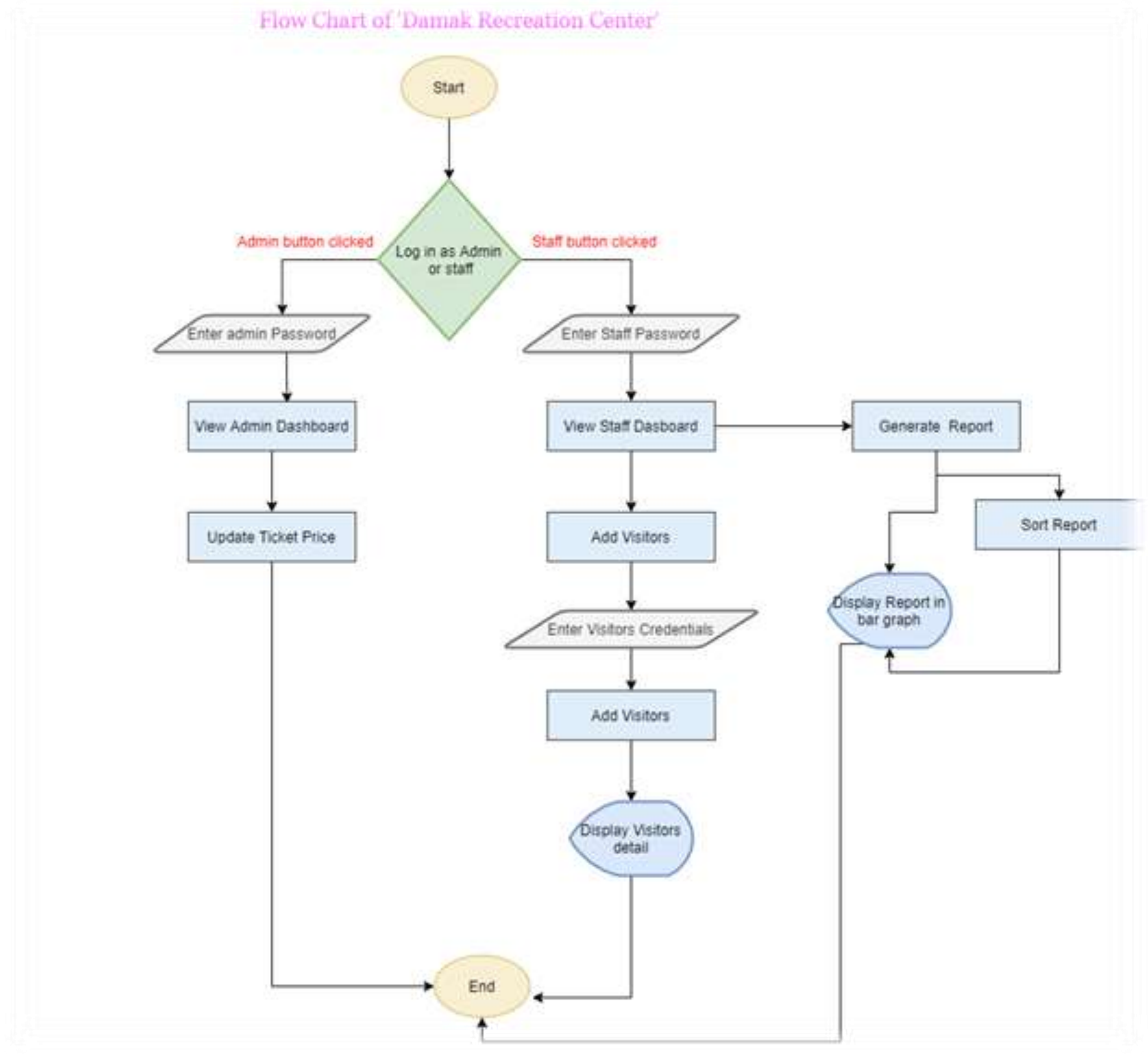


Figure 23 Flow Chart of "Damak Recreation Center System".

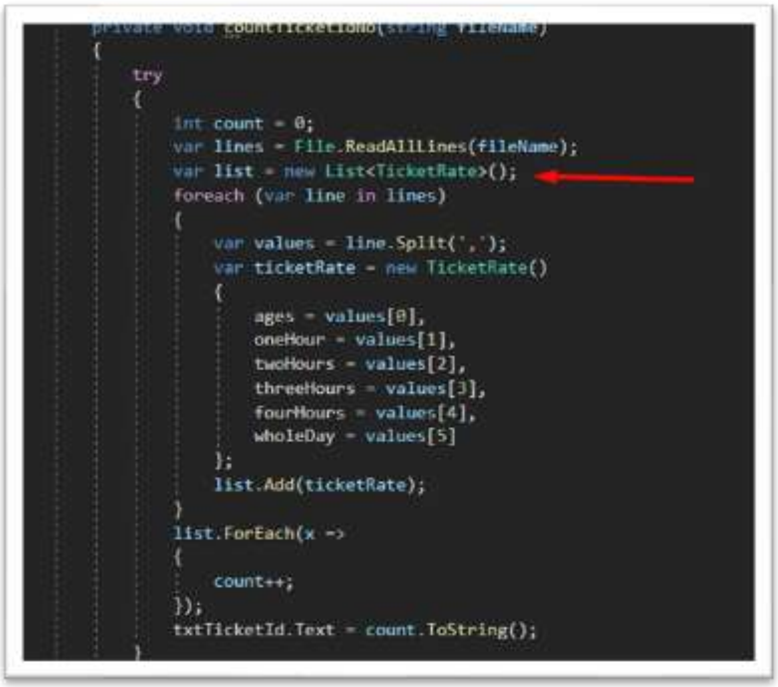
In the above flow chart diagram, we can see that a user can use the system either being an admin or a staff. To get the privileges of admin, the user must enter the correct admin password. After entering as admin, the admin can edit and update the price of a ticket. Whereas on the other hand, if the user enters a correct password on the staff login form, then the staff can add visitors to the system also generate a report of the recreation center based on its day, total earning, and a total number of visitors.

7. Data Structure And Algorithm

7.1. Data Structure used.

7.1.1. List

A List<T> is a collection of strongly typed objects which can be easily accessed by index and has methods for sorting, searching, and even modifying list (TutorialsTeacher, n.d.).

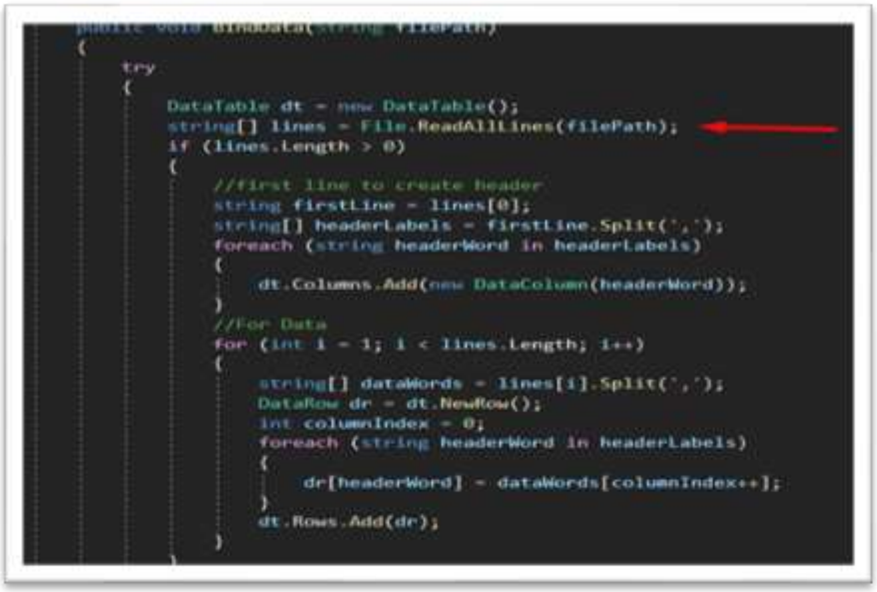


```
private void countTicketIdNo(string fileName)
{
    try
    {
        int count = 0;
        var lines = File.ReadAllLines(fileName);
        var list = new List<TicketRate>();
        foreach (var line in lines)
        {
            var values = line.Split(',');
            var ticketRate = new TicketRate()
            {
                ages = values[0],
                oneHour = values[1],
                twoHours = values[2],
                threeHours = values[3],
                fourHours = values[4],
                wholeDay = values[5]
            };
            list.Add(ticketRate);
        }
        list.ForEach(x =>
        {
            count++;
        });
        txtTicketId.Text = count.ToString();
    }
```

Figure 24 Use of list in generating unique serial ticket id for each visitor.

7.1.2. Array

An array is a group of similar-typed variables which are referred to by a common name. Arrays are used to store multiple values in a single variable, instead of declaring separate variables for each value. A square bracket is used to declare an array and define the variable type. In most cases, we need to store a large amount of data of a similar type. To store those huge amounts of data, we have to define numerous variables which would be tough to memorize all variable names while writing code. So to eradicate that issue, we need an array (R, 2021).

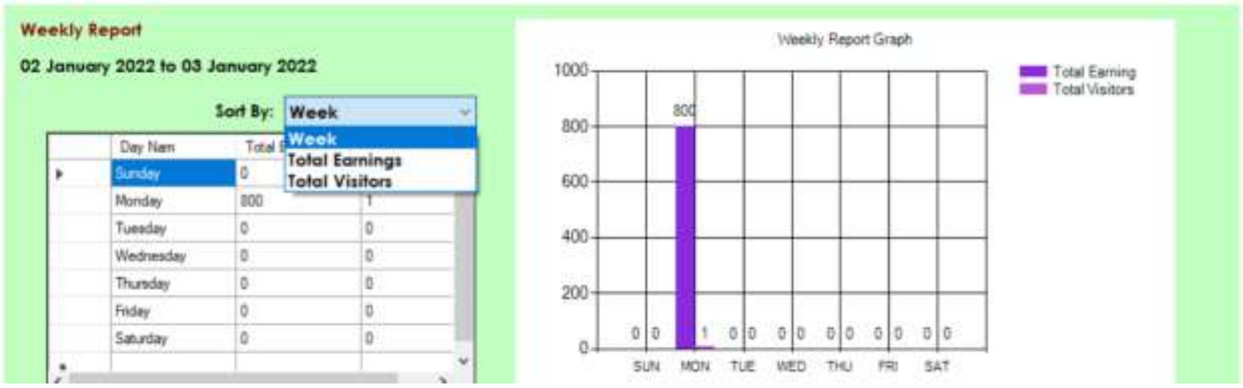


```
public void BindData(string filePath)
{
    try
    {
        DataTable dt = new DataTable();
        string[] lines = File.ReadAllLines(filePath);
        if (lines.Length > 0)
        {
            //first line to create header
            string firstLine = lines[0];
            string[] headerLabels = firstLine.Split(',');
            foreach (string headerWord in headerLabels)
            {
                dt.Columns.Add(new DataColumn(headerWord));
            }
            //For Data
            for (int i = 1; i < lines.Length; i++)
            {
                string[] dataWords = lines[i].Split(',');
                DataRow dr = dt.NewRow();
                int columnIndex = 0;
                foreach (string headerWord in headerLabels)
                {
                    dr[headerWord] = dataWords[columnIndex++];
                }
                dt.Rows.Add(dr);
            }
```

Figure 25 Array used to store line of .csv files

7.2. Algorithm Used

Bubble sort algorithm is used to sort the weekly report based on days of the week, total earnings of each day, and total visitors of each day in a bar graph.



Weekly Report

02 January 2022 to 03 January 2022

Sort By: Week

Day Nam	Total E	
Sunday	0	
Monday	800	1
Tuesday	0	0
Wednesday	0	0
Thursday	0	0
Friday	0	0
Saturday	0	0

Figure 26 Sorting feature in 'Damak Recreation Center System'.

Flow Chart of sorting algorithm.

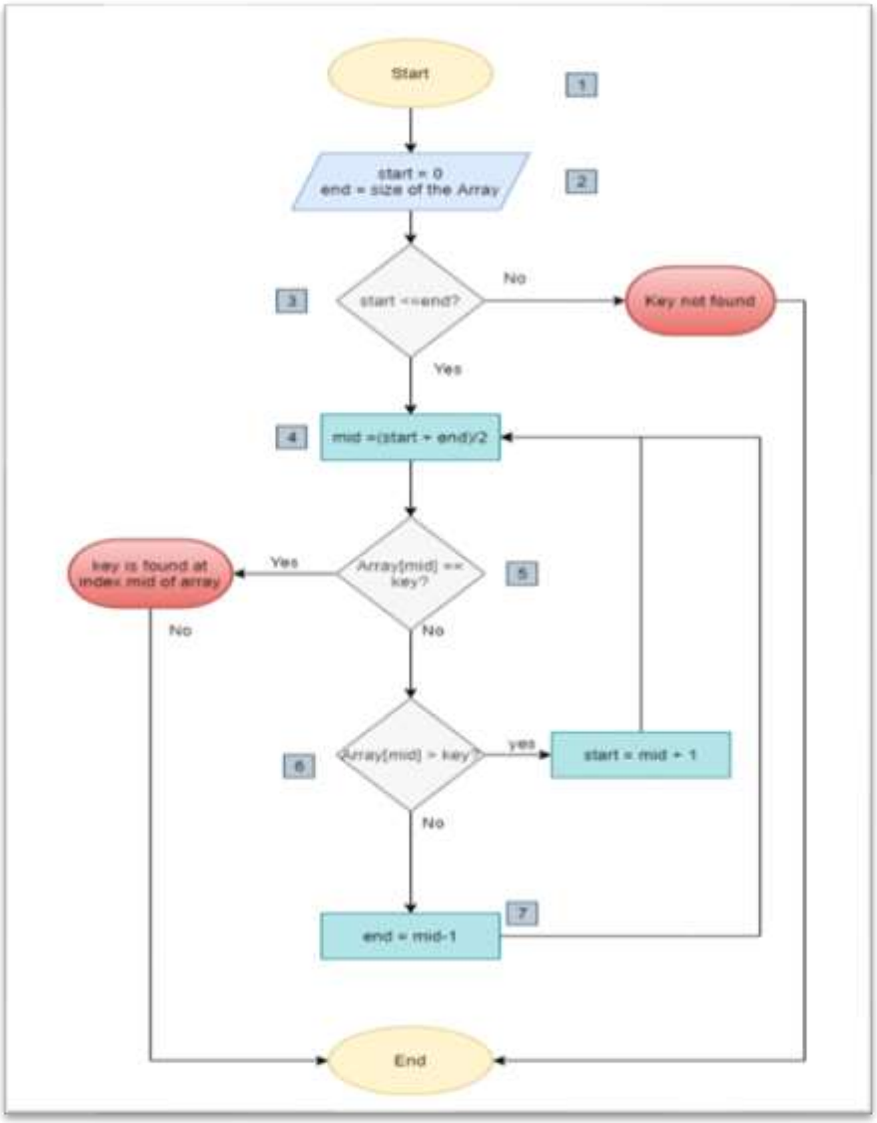
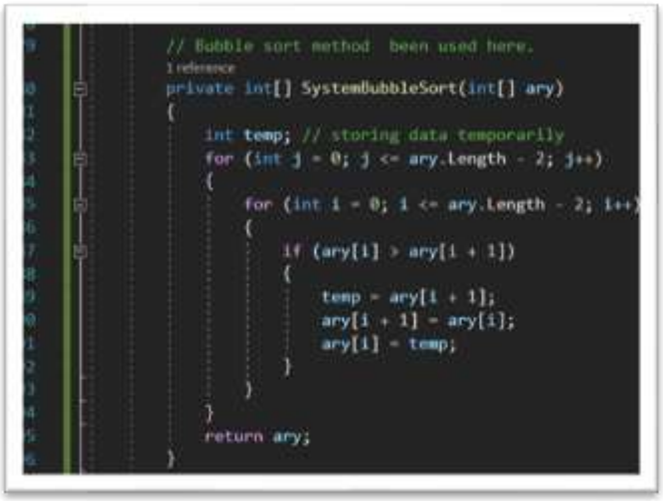


Figure 27 FlowChart of sorting algorithm.

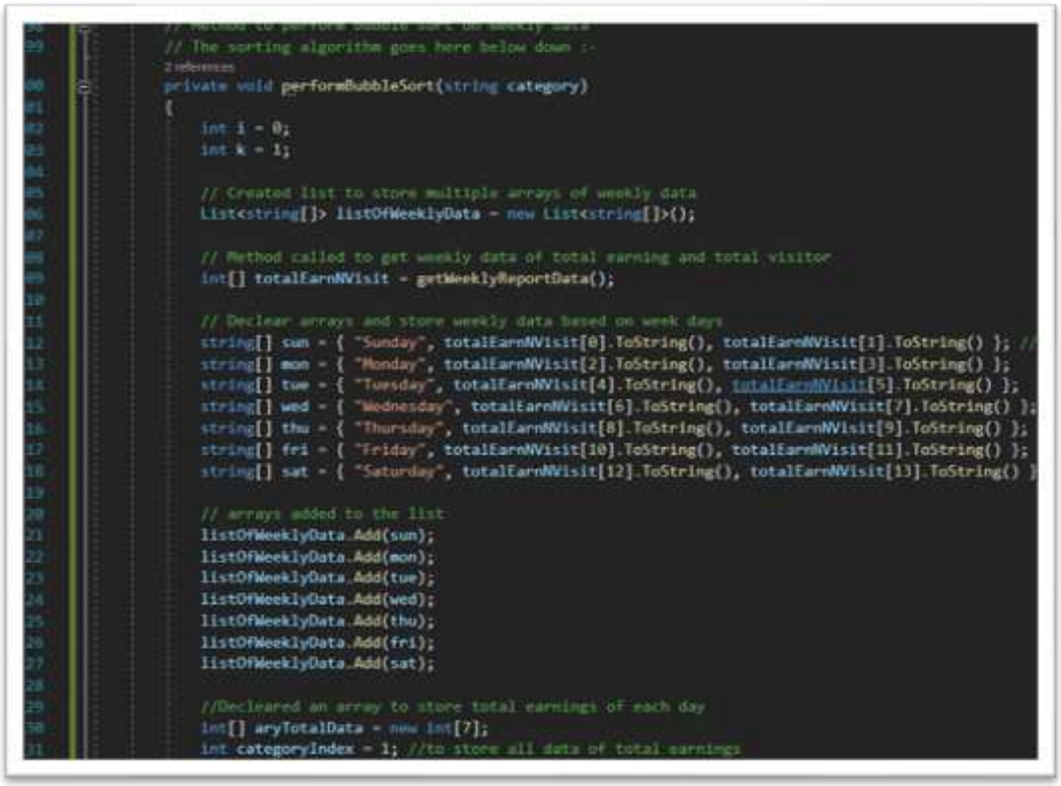
In the above flowchart, we can see that in 2, the first start value is zero, and the end value is the size of an array. If the start is less than the end value (at 3) then the whole code will not run else mid (at 4) will be storing the middle value of the array. If the middle value of the array is the key value (at 5) then the whole process will end (5 to 7) else, if the value turns out to be greater than the key value (at 6) then the start value will be (mid + 1) else the end value will be (mid - 1) (at 7). And again the whole code will run from 2.

Used code:



```
// Bubble sort method  been used here.
1 reference
private int[] SystemBubbleSort(int[] ary)
{
    int temp; // storing data temporarily
    for (int j = 0; j <= ary.Length - 2; j++)
    {
        for (int i = 0; i <= ary.Length - 2; i++)
        {
            if (ary[i] > ary[i + 1])
            {
                temp = ary[i + 1];
                ary[i + 1] = ary[i];
                ary[i] = temp;
            }
        }
    }
    return ary;
}
```

Figure 28 Bubble sort algorithm code in the system.



```
// Method to perform bubble sort on weekly data
// The sorting algorithm goes here below down :-
2 references
private void performBubbleSort(string category)
{
    int i = 0;
    int k = 1;

    // Created list to store multiple arrays of weekly data
    List<string[]> listOfWeeklyData = new List<string[]>();

    // Method called to get weekly data of total earning and total visitor
    int[] totalEarnNVisit = getWeeklyReportData();

    // Declear arrays and store weekly data based on week days
    string[] sun = { "Sunday", totalEarnNVisit[0].ToString(), totalEarnNVisit[1].ToString() }; //
    string[] mon = { "Monday", totalEarnNVisit[2].ToString(), totalEarnNVisit[3].ToString() };
    string[] tue = { "Tuesday", totalEarnNVisit[4].ToString(), totalEarnNVisit[5].ToString() };
    string[] wed = { "Wednesday", totalEarnNVisit[6].ToString(), totalEarnNVisit[7].ToString() };
    string[] thu = { "Thursday", totalEarnNVisit[8].ToString(), totalEarnNVisit[9].ToString() };
    string[] fri = { "Friday", totalEarnNVisit[10].ToString(), totalEarnNVisit[11].ToString() };
    string[] sat = { "Saturday", totalEarnNVisit[12].ToString(), totalEarnNVisit[13].ToString() }

    // arrays added to the list
    listOfWeeklyData.Add(sun);
    listOfWeeklyData.Add(mon);
    listOfWeeklyData.Add(tue);
    listOfWeeklyData.Add(wed);
    listOfWeeklyData.Add(thu);
    listOfWeeklyData.Add(fri);
    listOfWeeklyData.Add(sat);

    //Decleared an array to store total earnings of each day
    int[] aryTotalData = new int[7];
    int categoryIndex = 1; //to store all data of total earnings
```

Figure 29 Sorting according to selected category.

8. Reflection of Learning Experience

When we stand in front of a mirror, we see ourselves due to the reflection of light through our body to mirror forming a virtual image. Similarly, this reflection portion would be forming a basic experience achieved from completing CW. The CW was to develop a desktop application using the C# .net framework. I was not familiar with C# and Visual Studio IDE. It was very interesting and fun doing the CW. Building 'Damak Recreation Center System' using the IDE with its fabulous feature of drag and drop helped me to build the system smoothly. Building the system with an object-oriented approach was challenging for me. I am not so good with coding if sincerely to be told, but saving and retrieving different CSV files with its objects' with serialization gave me more depth and knowledge of OOP.

The CW played a great role for especially me to revise what we studied in the semester. Not only to recall the lessons but also it enhanced the skill of research and doing work on my own, which boosted my confidence and knowledge. During the journey, some of the life lessons that I have learned and want to reflect on are:

i) <u>Procrastination</u>

I could have finished coursework many days ago but I procrastinated my work which led me to rush in the last few days of submission. We should always remember that slow and steady wins the race.

ii) <u>Begin from the beginning</u>

After I started to worry and rush, I directly started to do coursework without learning the basics. I encourage everyone to start from the basics before doing anything big or small.

iii) <u>Patience</u>

Patience is not the ability to wait, but the ability to keep a good attitude while waiting. We students should make patience our best friend. It will always help us in times of trouble, especially while doing this type of CW.

iv) <u>Learn little by little.</u>

It takes a step at a time to reach the top. E.g. If I would have learned the C# and Visual Studio IDE little by little a few years or months back when I had sacks of free time then it could have opened many doors of opportunity which includes this coursework too. But carless me.

v) <u>Better ask help (Help me !)</u>

We should never hesitate and step back to ask for help or raise a question about what we haven’t understood from a teacher or friend. I didn’t ask for help from anyone which made me suffer bitterly. If requesting help removes your iceberg then we should better do it.

9. Conclusion

9.1. Evaluation of work

In life we see, even long roads have an end to them, and the same goes for this coursework (CW). Finally, to conclude, I've learned about the C#, serialization, object-oriented programming in clarity, got familiar with.net framework, and life lessons during the journey of CW. Their use in the IT world. After finishing the CW I had a blast of happiness and little sadness at the same time.

9.2. Learning Outcomes

Sincerely speaking I learned more about life lessons than the technical part. There were corners of shadows and valleys of intellectual academic emptiness which were fulfilled by this CW but the best outcome of this CW is that I was able to build a complete system using a file system as a database. I learned about file handling, exception handling, serialization techniques, object-oriented approach, using Visual Studio, designing a good project,

9.3. Difficulties Encountered and overcoming it.

A lot of time I needed to ask the module teacher about my CW progress, but it was not easy to meet a suitable timetable for both sir and me. But my basketful thanks goes to Mr. Ravi Rouniyar sir for always being there and helping me when I was in trouble. Learning new things is never easy, if it was easy then everyone would have studied IT and done it. I also suffered pain and depression when I was not able to sort weekly data using serialization technique, was not able to update ticket price rate, and was not able to generate daily and weekly report. Sometimes I was not able to find out good research material. But after all, all that we can do is remain cool, patient, and keep hustling, and that's what I did. Due to which writing this conclusion today is possible. Everything considered, in the end, I would like to say that becoming a few days of C# developer by doing this CW is in some way fun and rewarding. I now have learned to sit in front of a laptop screen for several hours without distraction and complaining. Hope to study at this same pace to be a good software engineer someday and make an impact in others' life.

10. Reference

R, V. (2021, Aug 6). *GreatLearning* . Retrieved from www.mygreatlearning.com: https://www.mygreatlearning.com/blog/what-is-an-array-learn-more-in-one-read/

TutorialsTeacher. (n.d.). *TutorialsTeacher*. Retrieved from www.tutorialsteacher.com: https://www.tutorialsteacher.com/csharp/csharp-list

VisualParadigm_ClassDiagram. (2021, 12 13). Retrieved from VisualParadigm: https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-class-diagram/

wondershare_SystemArchitecture. (2021, 11 20). Retrieved from wondershare: https://www.edrawsoft.com/architecture-diagram.html

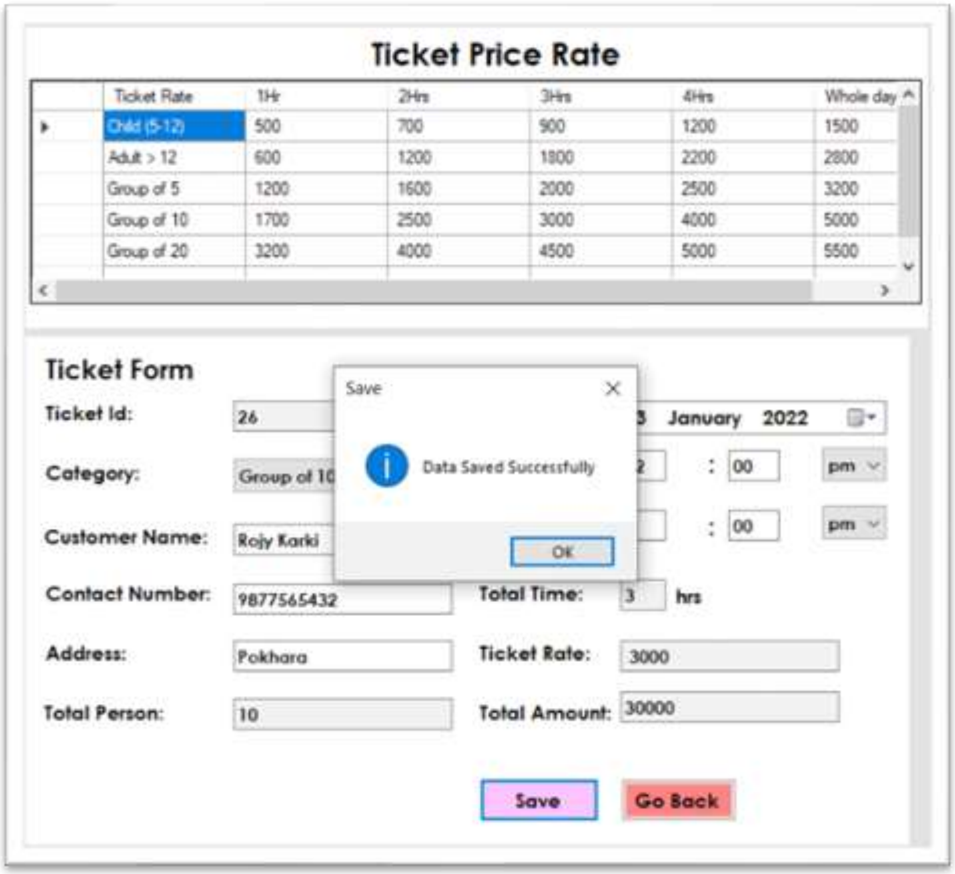
11. Appendix

11.1. Testing

11.1.1. Test 1 of serialization

Objective	To check the serialization of the code.
Action	A new visitor (customer) was added.
Expected Result	A new customer must be added to a CSV file and again retrieve the data to show it in the data grid view table.
Actual Result	The customer was added successfully.
Conclusion	The test was successful.

Table 7 Serialization test 1.



Ticket Price Rate

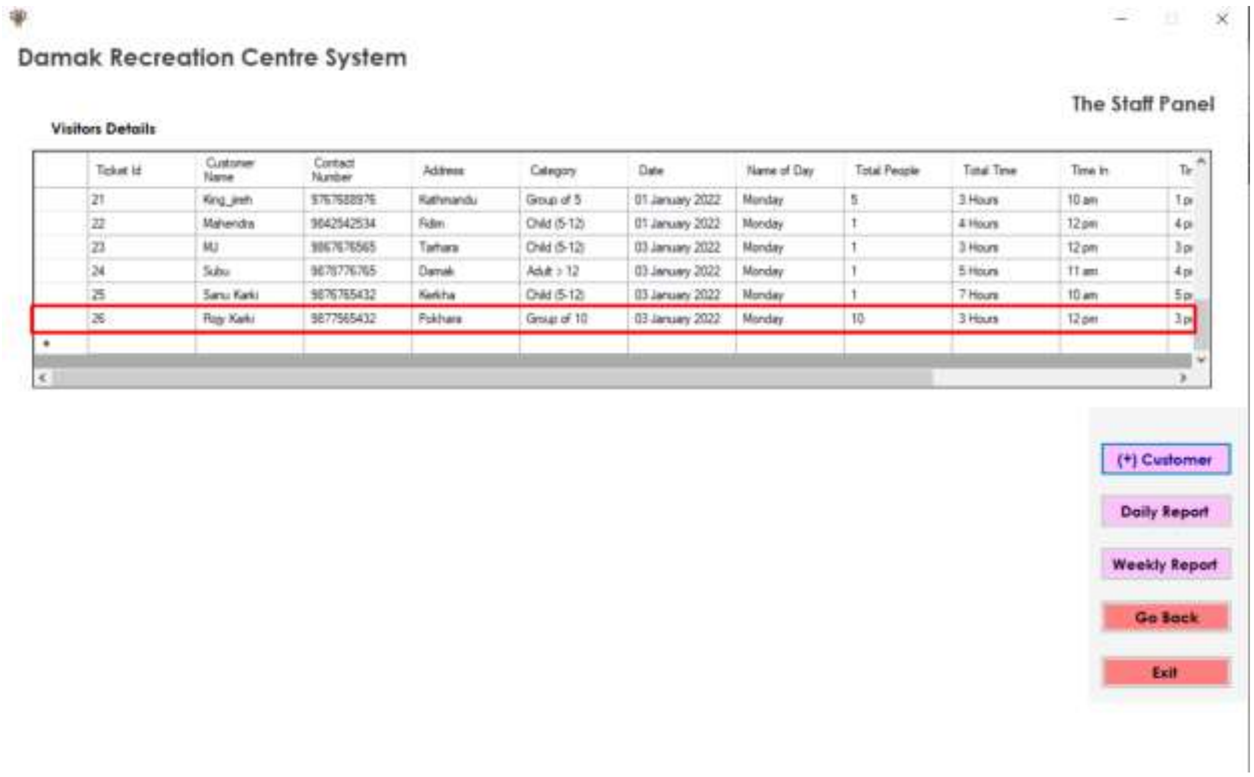
Ticket Rate	1Hr	2Hrs	3Hrs	4Hrs	Whole day
Child (5-12)	500	700	900	1200	1500
Adult > 12	600	1200	1800	2200	2800
Group of 5	1200	1600	2000	2500	3200
Group of 10	1700	2500	3000	4000	5000
Group of 20	3200	4000	4500	5000	5500

Ticket Form

Ticket Id: 26
3 January 2022
Category: Group of 10
2 : 00 pm
: 00 pm
Customer Name: Rojy Karki
Contact Number: 9877565432
Total Time: 3 hrs
Address: Pokhara
Ticket Rate: 3000
Total Person: 10
Total Amount: 30000

Save
Data Saved Successfully

Figure 30 Adding new visitors



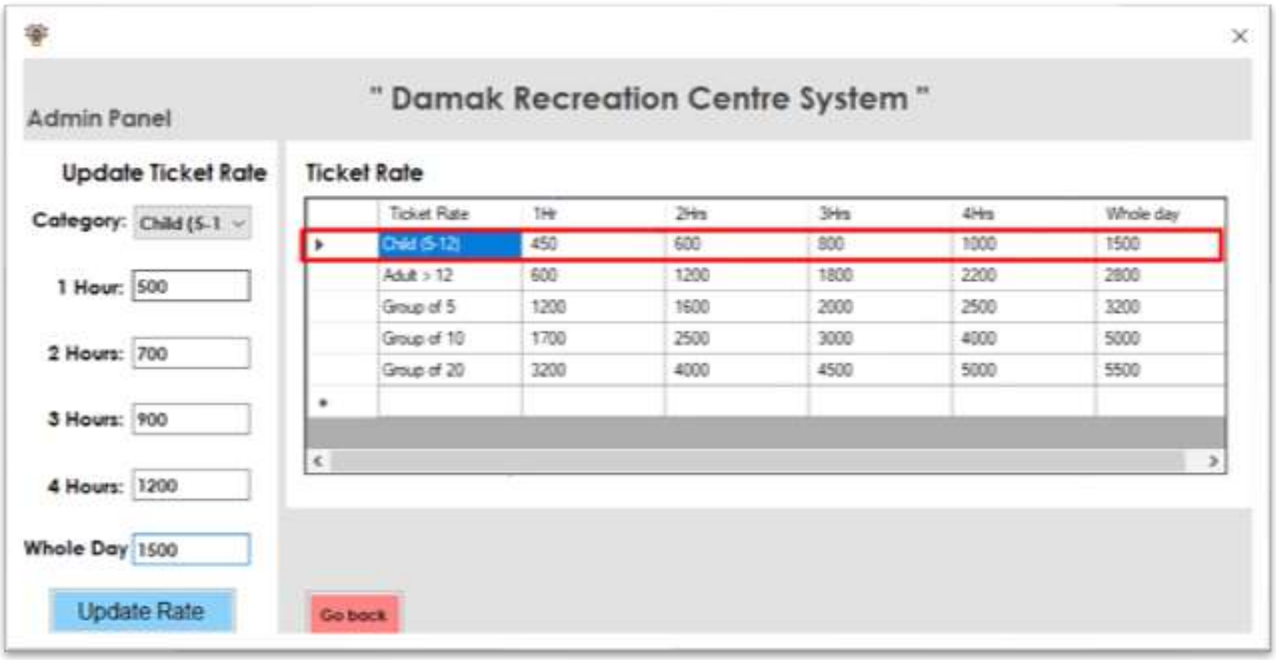
Ticket Id	Customer Name	Contact Number	Address	Category	Date	Name of Day	Total People	Total Time	Time In	Tr
21	King_jeeh	9767688976	Kathmandu	Group of 5	01 January 2022	Monday	5	3 Hours	10 am	1 p
22	Mahendra	9842542534	Fidim	Child (5-12)	01 January 2022	Monday	1	4 Hours	12 pm	4 p
23	MJ	9867676565	Tarhara	Child (5-12)	03 January 2022	Monday	1	3 Hours	12 pm	3 p
24	Subu	9878776765	Damak	Adult > 12	03 January 2022	Monday	1	5 Hours	11 am	4 p
25	Sanu Karki	9876765432	Kerkha	Child (5-12)	03 January 2022	Monday	1	7 Hours	10 am	5 p
26	Rojy Karki	9877565432	Pokhara	Group of 10	03 January 2022	Monday	10	3 Hours	12 pm	3 p

Figure 31 The added customer is shown in the table.

11.1.2. Test 2 of editing and updating Ticket Price Rate

Objective	To edit and update ticket price rate.
Action	Customer category was selected and inserted new ticket price.
Expected Result	A new customer must be added.
Actual Result	A new customer was added and shown in the data grid view table.
Conclusion	The test was successful.

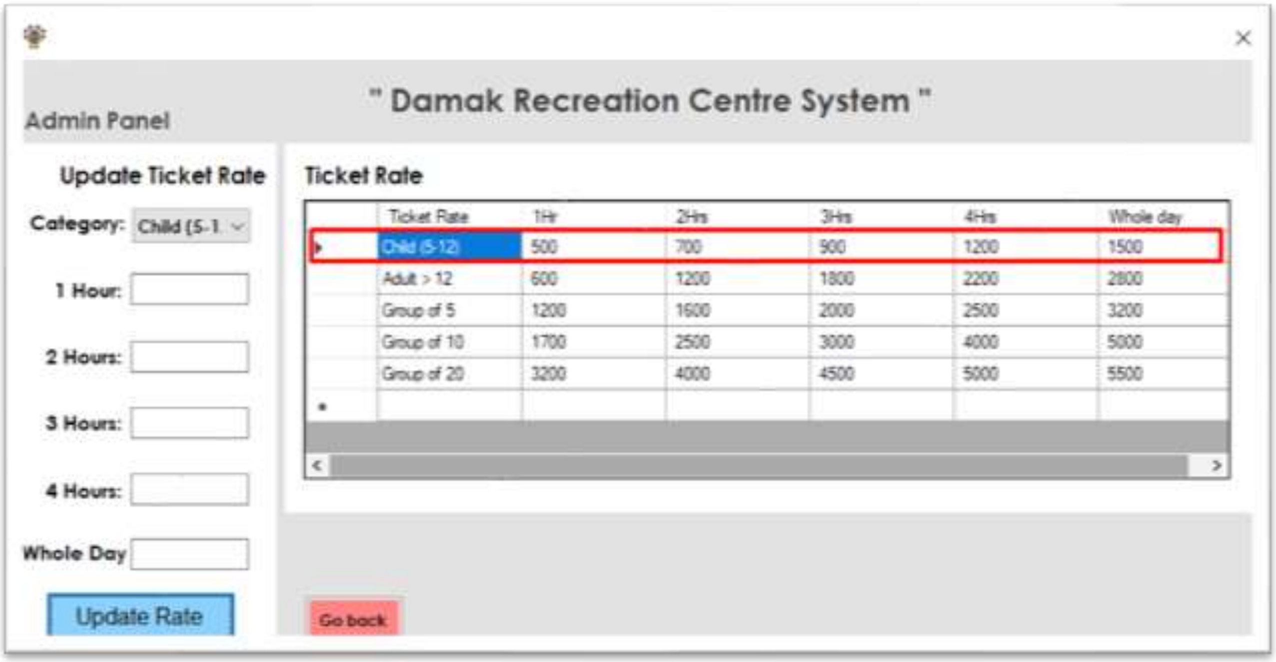
Table 8 Editing and updating ticket price rate test 2.



Category:	Child (5-1
1 Hour:	500
2 Hours:	700
3 Hours:	900
4 Hours:	1200
Whole Day	1500

Ticket Rate	1Hr	2Hrs	3Hrs	4Hrs	Whole day
Child (5-12)	450	600	800	1000	1500
Adult > 12	600	1200	1800	2200	2800
Group of 5	1200	1600	2000	2500	3200
Group of 10	1700	2500	3000	4000	5000
Group of 20	3200	4000	4500	5000	5500

Figure 32 Editing ticket price



" Damak Recreation Centre System "

Admin Panel

Update Ticket Rate

Category: Child (5-1

1 Hour:

2 Hours:

3 Hours:

4 Hours:

Whole Day

Update Rate

Ticket Rate

Ticket Rate	1Hr	2Hrs	3Hrs	4Hrs	Whole day
Child (5-12)	500	700	900	1200	1500
Adult > 12	600	1200	1800	2200	2800
Group of 5	1200	1600	2000	2500	3200
Group of 10	1700	2500	3000	4000	5000
Group of 20	3200	4000	4500	5000	5500

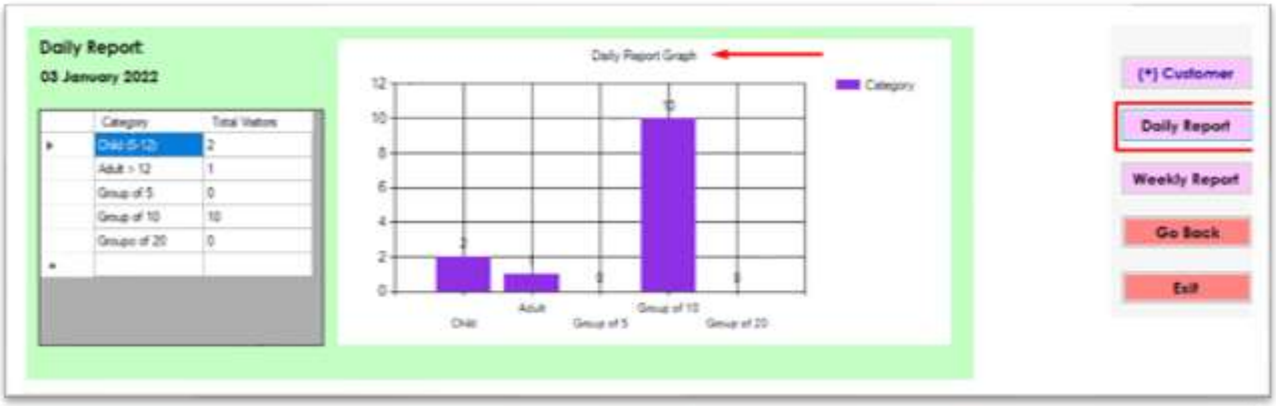
Go back

Figure 33 The ticket price was updated successfully.

11.1.3. Test 3 Generate daily report

Objective	To generate a daily report
Action	The daily report button was pressed.
Expected Result	When the button will be pushed, it must generate the report in a bar graph.
Actual Result	The daily report was generated in the bar graph.
Conclusion	The test was successful.

Table 9 Generating a daily report test 3.



Daily Report
03 January 2022

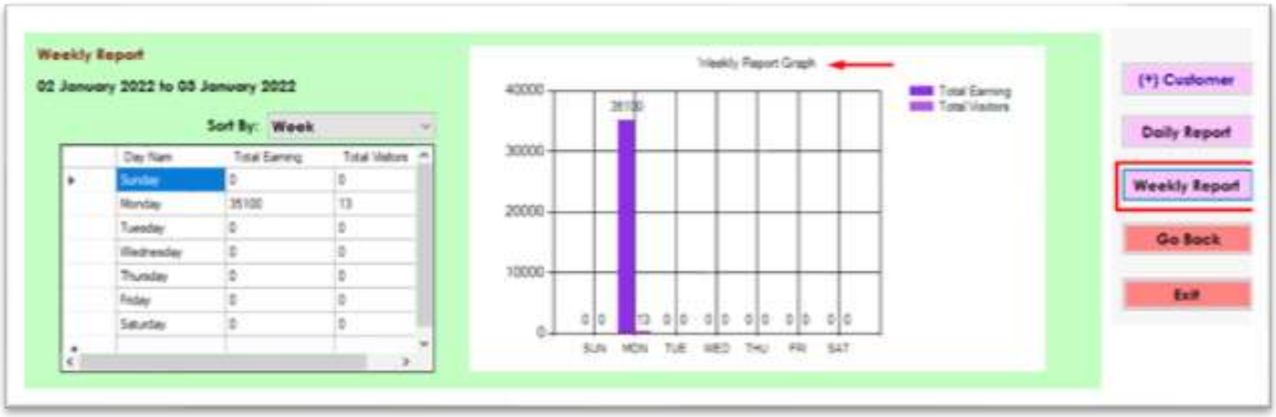
Category	Total Visitors
Child (5-12)	2
Adult > 12	1
Group of 5	0
Group of 10	10
Groups of 20	0

Figure 34 Daily reported been generated.

11.1.4. Test 4 of generating a weekly report

Objective	To generate a weekly report
Action	The Weekly report button was pushed.
Expected Result	The Weekly report must be generated in a bar graph.
Actual Result	The report was generated in the bar graph.
Conclusion	The test was successful.

Table 10 Generating weekly report test 4



Weekly Report

02 January 2022 to 08 January 2022

Sort By: Week

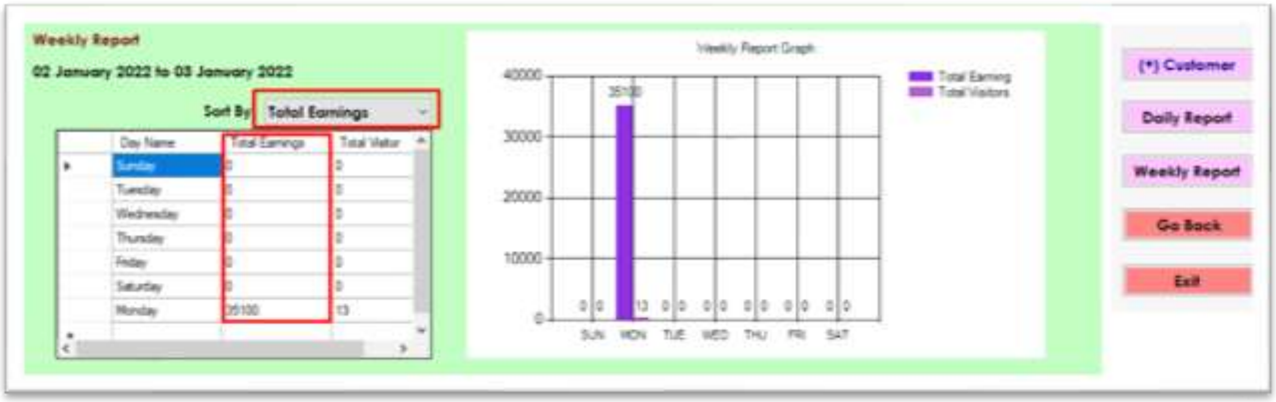
Day Nam	Total Earning	Total Visitors
Sunday	0	0
Monday	35100	13
Tuesday	0	0
Wednesday	0	0
Thursday	0	0
Friday	0	0
Saturday	0	0

Figure 35 Weekly report been generated.

11.1.5. Test 5 of sorting the weekly report based on total earning

Objective	To sort the weekly report based on total earning.
Action	Total earning was selected from the combo box.
Expected Result	The data in the grid view must be sorted in ascending order with the data of the total earning of the week.
Actual Result	The data was sorted.
Conclusion	The test was successful.

Table 11 Sorting weekly report based on total earning test 5


Weekly Report

02 January 2022 to 03 January 2022

Sort By: Total Earnings

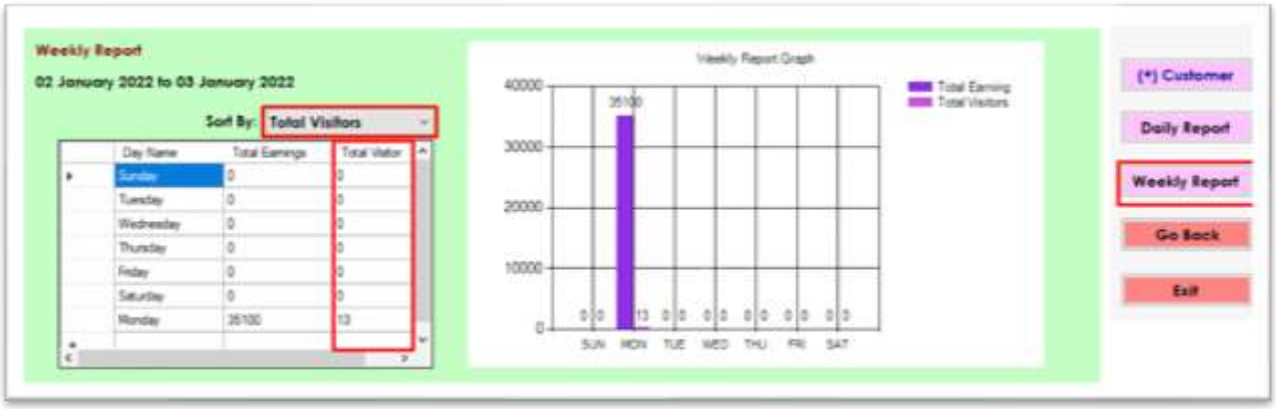
Day Name	Total Earnings	Total Visitor
Sunday	0	0
Tuesday	0	0
Wednesday	0	0
Thursday	0	0
Friday	0	0
Saturday	0	0
Monday	35100	13

Figure 36 Sorting the weekly report based on total earning.

11.1.6. Test 6 of Sorting the weekly report based on total visitors

Objective	To sort the weekly report based on total visitors.
Action	Total visitor item was selected from the combo box.
Expected Result	The data in the grid view table must be sorted in ascending order concerning day and total visitors.
Actual Result	The data was sorted.
Conclusion	The test was successful.

Table 12 Sorting weekly report based on total visitors test 6.


Weekly Report

02 January 2022 to 03 January 2022

Sort By: Total Visitors

Day Name	Total Earnings	Total Visitor
Sunday	0	0
Tuesday	0	0
Wednesday	0	0
Thursday	0	0
Friday	0	0
Saturday	0	0
Monday	35100	13

Figure 37 Sorting the weekly report based on total visitors.