National Institute of Co-operative Development

Polgolla

Diploma in Information Technology – 2019/2020

Ticket Management System

Final Project

Student Name	: - W.G.M.M.Ajmal
ID Number	:-DIT/FT/2019/14/363
Submission Date	: -2020/12/14

DECLARATION

I declare that the project entitled is an outcome of my own effort under the guidance of Mr. Dimuthu Kumara. The project is submitted to the National Institute of Co-operative development for the partial fulfilment of the Diploma in Information Technology course.

Signature of Course Coordinator

Date

ACKNOWLEDGEMENT

My most profound gratitude goes to my, Course Coordinator – Sectional Head of IT Mrs. I. Namali Nanayakara who gave me the opportunity to do this project. I am extending my sincere and heartfelt thanks to our esteemed guide Mr. Kumara Dimuthu, who provided the overall support and encouragement and for showing us the right way. I am extending my sincere thanks to our respected Computer Instructor of the IT Division at National Institute of Co-operative Development Mr. Chanaka Dissanayake, for allowing us to use the facilities available. Finally, I would like to thank my parents and friends for the support and encouragement they have given us during the course of my work. Words cannot fully express my appreciation to all individual who have contributed to this work.

ABSTRACT

Ticket Management System by Manual way is tedious process, since it involves work load and time consumption. In this system, we can easily manage the all ticket system operations.

The main feature of the Developed Ticket Management System is to allocate for the passengers and manage employees. Identification of the drawbacks of the existing system leads to the development computerized ticket management system that will be compatible to the existing system with the system which is more user friendly and more GUI oriented. We can improve the efficiency of the system, thus overcome the drawbacks of the existing ticket management system. Less human error, Strength and strain of manual labor can be reduced, High security, Data Redundancy can be avoided to some extent, Data consistency, Easy to handle, Easy to update the data, Easy record keeping Backup data can be easily generated.

This project is carried out using Visual Studio 2015 Enterprises as Front-End and MySQL sever database as Back-End. This Ticket Management System has Admin Login, Guest Login. This ticket management system is running through these dual login systems. Admin can manage all the operations in the system and a guest can log into the system and can reserve their bookings for their travel. This system will ease the work for passengers who willing to book tickets for their travel.

Table of Contents

LIST OF FIGURES

LIST OF TABLES

CHAPTER 01

INTRODUCTION

Introduction

1.1Background

This Ticket Management System is developed for aircore ticketing services in favor of the ticket management team which help themselves to save the records of the employees, passengers. It helps them from the manual work from which it is very difficult to find the record of the passengers, employees, payments, meals, bookings and other things.

This solution is developed on the plight of the ticket management team; through this they cannot require so eminent person to handle and manage the affairs of the passengers & employees in the services, all you need to do is to login as admin and you can see the information of all the passengers, employees, bookings and other things.

This system is fully computerized and having more flexible and safe options, and it is more securable way of store records of company therefore this TMS is much useful for the current ticketing functions.

1.2 Purpose

The purpose of this project is to make an automated system to carry out the various operations of Aircore Ticketing Company. This Ticket Management System will provide the ease of use to the admin of the company by performing all the work on a computer system rather than following a manual approach. This approach helps improving the reliability of the data maintained and provides a fast and efficient interface for the users of the software.

1.3 Objectives and Scope

This software product the Ticket management to improve their services for all the employees and passengers. This also reduce the manual work of the person in admin panel and bundle of registers that were search when to find the information of the previous passenger, employee, payments and bookings. Through this system we can save the records of the passengers, employees and others in database. The database of the system will help to record all the details of the passengers, employees and others at once and help to manage records at any time.

- To automate each and every activity of the manual system, which increases its throughput.
- To provide a quick response with very accurate information as and when required.
- Reduce the cost of maintenance.

1.4 Project Schedule

As I scheduled my project work I have started my project work on October 1st week, it took 1 week for requirement gathering for the project, and another 2 weeks for Designing part of the project. At the starting of the 1 week of November I have started to code my project and it took 2 weeks, after that I have taken a testing part to my project and, debugged errors. As I scheduled I have completed my project on end of the November (4th Week).

1.5 Approach

Classical Waterfall Model

In order fulfill the requirements as highlighted in the objectives I have chosen Classical Waterfall Model to be used in this system to be developed. It is very easy to handle and identify the user requirements and easy to develop the project according to the user requirements. Therefore, I have chosen this software model to develop the project. Reasons for choosing this methodology is, it supports high user involvement, easy to use & understand, taking reviews in the completion of every phase and appropriate model for this project.

The phases of Classical Waterfall Model are;

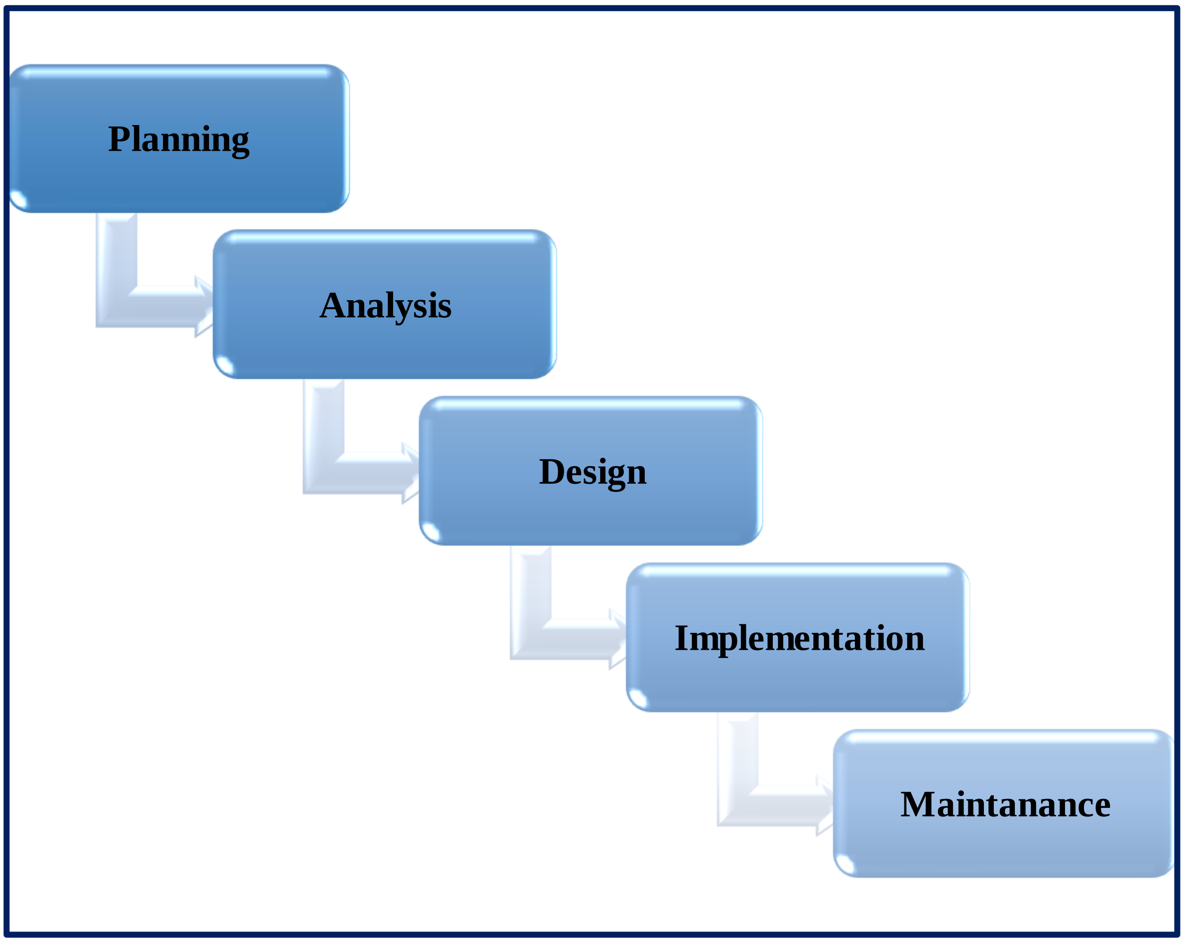


Figure 1 Classical Waterfall Model – Approach of Project

CHAPTER 02

SOFTWARE REQUIREMENT SPECIFICATION

Software Requirement Specification

2.1 User Requirement Specification

The user requirements for this system is to make the system fast and flexible, less prone to error, reduce expenses and save the time.

- Less human error.
- Strength and strain of manual labor can be reduced.
- High Security.
- Data redundancy.
- Data consistency.
- Easy to handle.
- Easy data updating.

2.2 System Requirement Specification

2.2.1 Functional System Requirements

It defines the functional requirements that applicable to the Hostel Management System. These are the sub modules of the system.

- Admin
- Guest
- Meals
- Hotel
- Transport
- Reservation
- Tally up
- Manage

2.2.3 Non-Functional Requirement Specification

- Performance Requirements.
- Safety Requirements.
- Security Requirements.
- Flexibility
- Reusability

2.3 Hardware Requirements.

- Processor
- RAM
- Hard Disk
- Keyboard
- Monitor or LCD

2.4 Software Requirements.

Operating System:	Microsoft Windows.
Front-end Tool:	Visual Basic 2015 –Enterprise.
Back-end Tool:	MySQL Database.

2.5 Software Tool Used

- **Front-End Tool – Visual Basic 2015 Enterprises**

Microsoft Visual Studio is an integrated development environment (IDE) from Microsoft. It is used to develop computer programs, as well as websites, web apps, web services and mobile apps. Visual Studio uses Microsoft software development platforms such as Windows API, Windows Forms, Windows Presentation Foundation, Windows Store and Microsoft Silverlight. The integrated debugger works both as a source-level debugger and a machine-level debugger.

I have used Visual Studio tool to code this software, and also I have used this software to testing part of the system and debugged errors by testing the system. It is a very efficient IDE to develop any software.

- **Back-End Tool – MySQL**

MySQL is an Oracle-backed open source relational database management system based on Structured Query Language. MySQL runs on virtually all platforms, including Linux, UNIX and Windows. Although it can be used in a wide range of applications, MySQL is most often associated with web applications and software development.

I have used MySQL database to save all records of the system. It eases the work of the project development.

CHAPTER 03

SYSTEM ANALYSIS

3.System Analysis

3.1 Data Collection Methods

- Documentations (Reports, PDF, Books etc.)
- Observations
- Questionnaires

3.2 Existing System

The existing Ticket Management System is a manual way of handing records of ticketing services.

- More human error.
- More strength and strain of manual labor needed.
- Repetition of some procedures.
- Lack of security.
- Data redundancy.
- Record keeping is difficult.
- Time consuming.

3.3 Proposed System

The proposed Ticket Management System is fully computerized system. It provides easy and quick wat to access the records. It has authorization schemes. It reduces the consuming of time. Through this system we can save the records of the passengers, employees and other module records in database easily. It provides an easy way to manage (Add, Update, Delete, View, Search, Filter) records of ticketing services.

3.4 Literature Review

- According to the view of (Patrick J. TooleSudhir KrishnaswamyNima MoayediDavid N. LordFrancisco J. Gutierrez, 2007) embodiments of the invention include a system and set of processes for managing tickets. The system maintains a database of tickets belonging to a company and the allotment of the tickets to employees and clients. The system provides interfaces for requesting tickets and managing ticket requests. The system optionally provides automated request processing, ticket resale and purchase and electronic distribution.
- According to the view of (panelNorazahMohd , SukiaNorbayahMohd Sukib, 2017) In the context of intense market competition, airlines are enriching their business operations by offering flight ticket booking apps that can be downloaded on mobile devices. This study aims to examine the intention of individuals to use such apps, and uses Structural Equation Modelling (SEM) to analyses the data gathered from individuals in Malaysia. Perceived usefulness represents the greatest influence on individuals in respect of their intention to engage with such an app offered on a mobile device. Airline companies should consider using advances in ICT within their overall portfolio of marketing strategies, if they wish to become more competitive in the current market. They should utilize the interactive and attractive features of online channels in order to encourage more individuals to try their flight ticket booking apps on mobile devices. The proposed model could be used as a baseline model in future research.

- According to the view of (Kola Ayanlowo, O. Shoewu , Segun O. Olatinwo Omitola , Damilola D. Babalola, 2014) hostels Consequently it has increased knowledge and helped produce a population of enlightened citizens who can easily abide by the rules of civilized society and contribute meaningfully to the process of democratic governance. Most of the newly established educational institutions however, are using the old conventional techniques for managing their assets especially hostel facilities. These old techniques with its inherent limitations have impacted negatively on the overall organizational efficiency of this educational systems. In this paper, the development of an automated hostel accommodation management system is proposed.

- Finally, according to my view, I have introduced a feature in my system that a passenger can surf the travel agency details under the supervision of the receptionist in the computer system, therefore passenger can know more information about the travelling services, if the passenger wishes to allocate a ticket, she/he can log into the system by getting a username and password from the receptionist and log into the system. Passenger can fill a register form which is shown in the system to reserve/get their id. Admin can manage all the details about passengers such as view, search, update, delete and filter. As it is admin can add and manage the hotel records as well as admin can add and manage employee records who are working in the company and admin can manage the all other modules too. I hope it will work out very efficiently in future for any ticket management systems. This Ticket Management System works very flexibly and efficiently for the ticket service companies and support the user to operate the system very easily.

CHAPTER 04

SYSTEM DESIGN

4.System Design

4.1 System Architecture

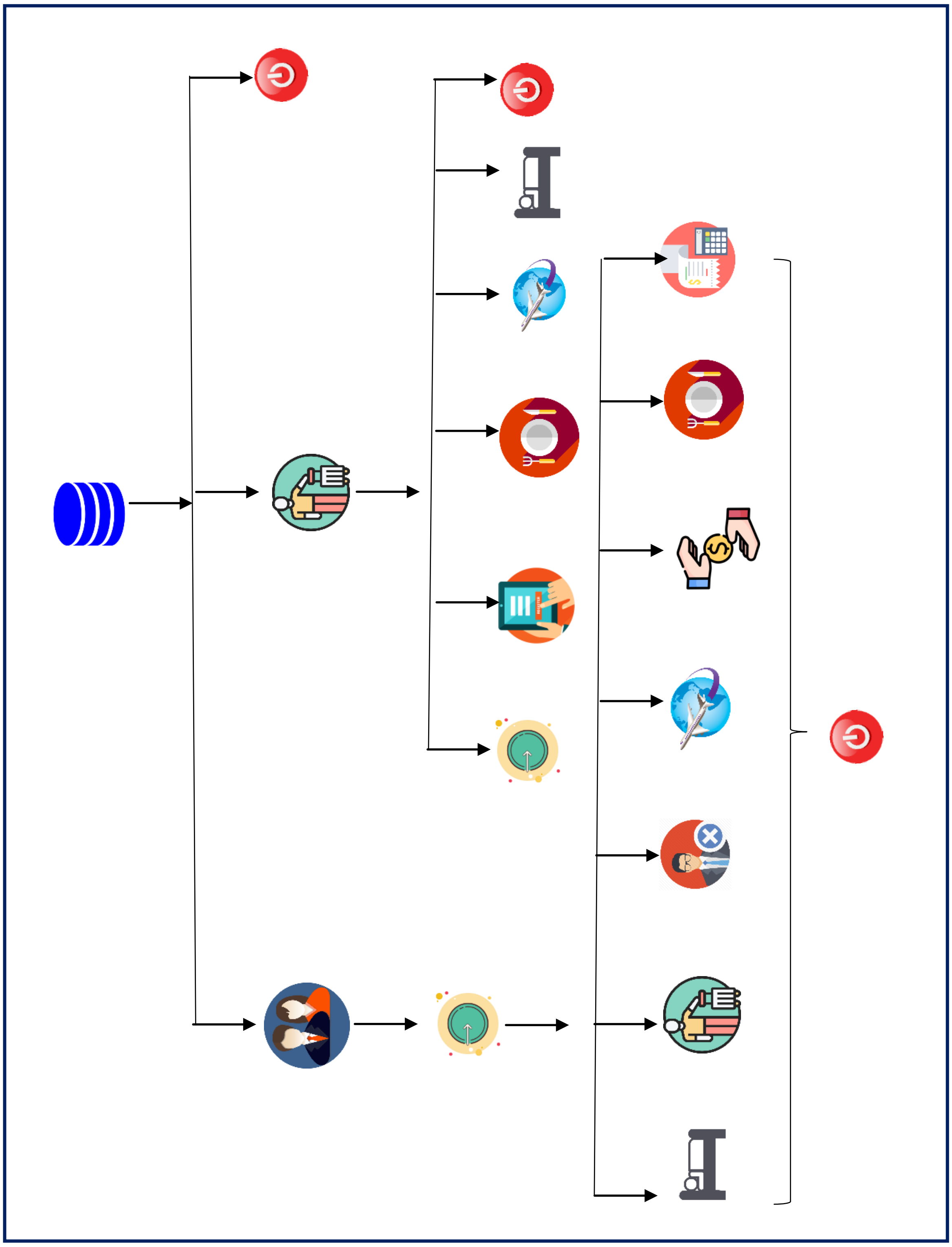


Figure 2 System Architecture Graphical View

4.2 ER-Diagram

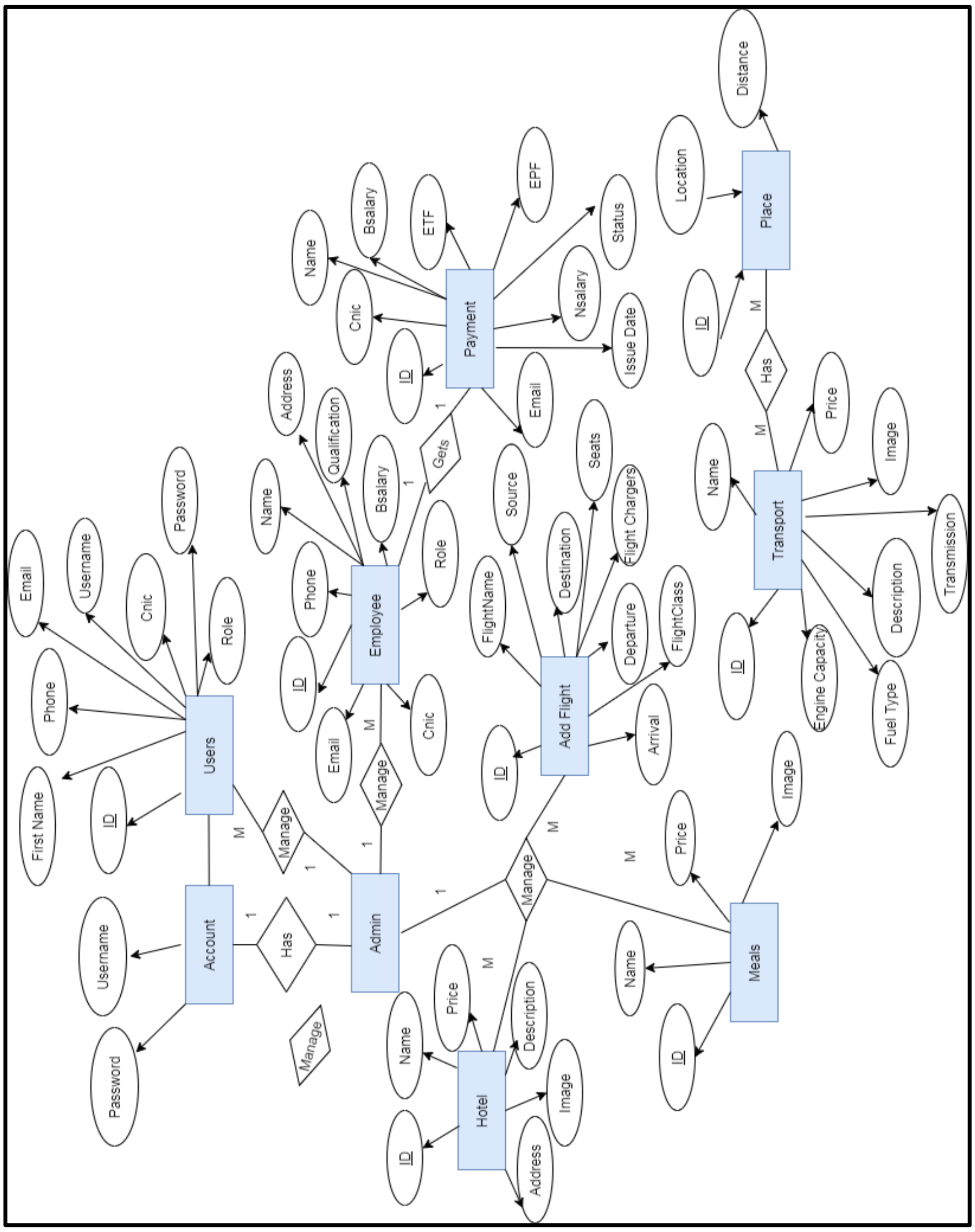
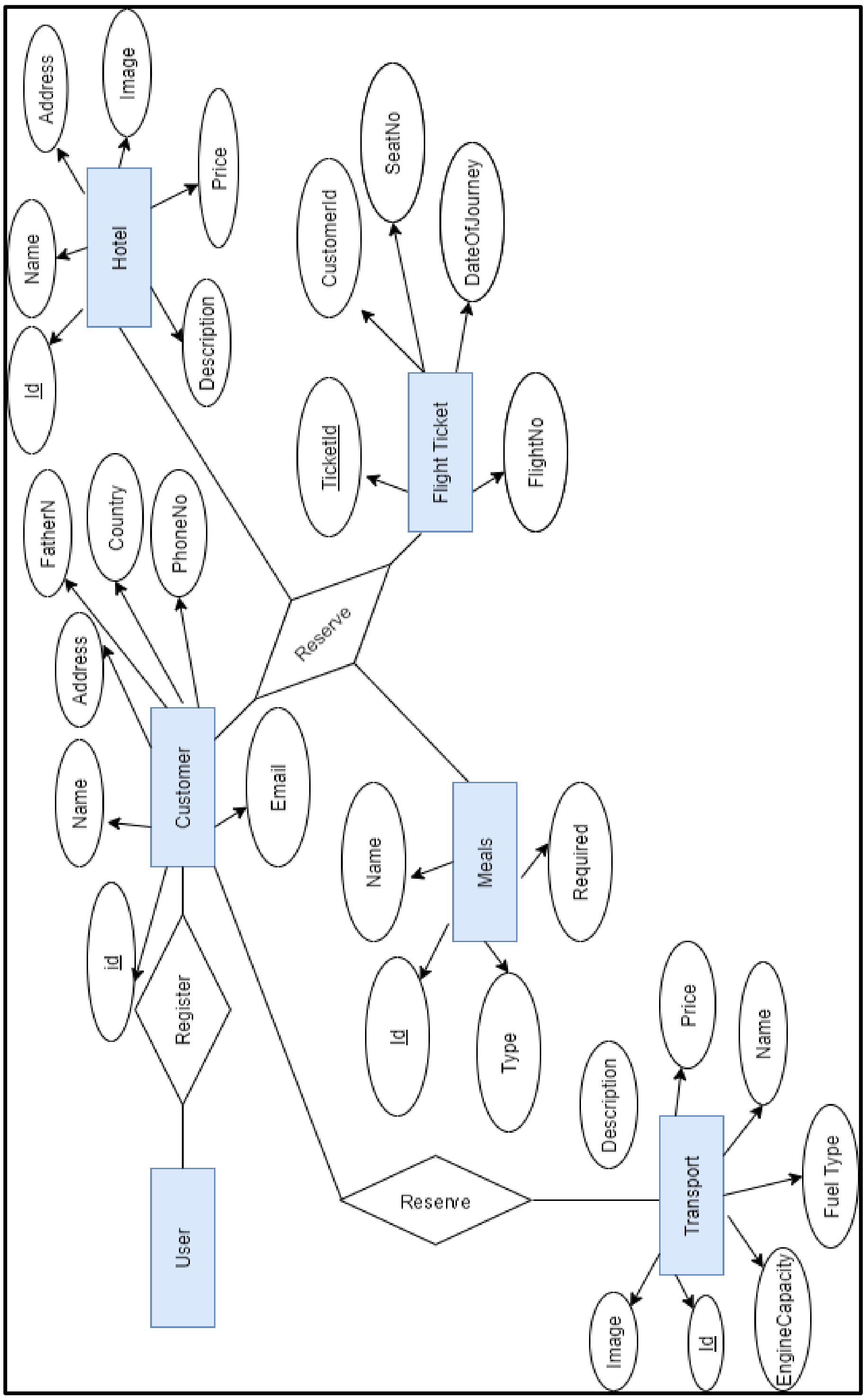


Figure 3 ER-Diagram – Ticket Management System



User
Register
Customer
id
Name
Address
FatherN
Country
PhoneNo
Email
Reserve
Hotel
Id
Name
Address
Image
Price
Description
Flight Ticket
TicketId
CustomerId
SeatNo
DateOfJourney
FlightNo
Meals
Name
Required
Id
Type
Reserve
Transport
Description
Price
Name
Fuel Type
EngineCapacity
Id
Image

4.3Use Case Diagram

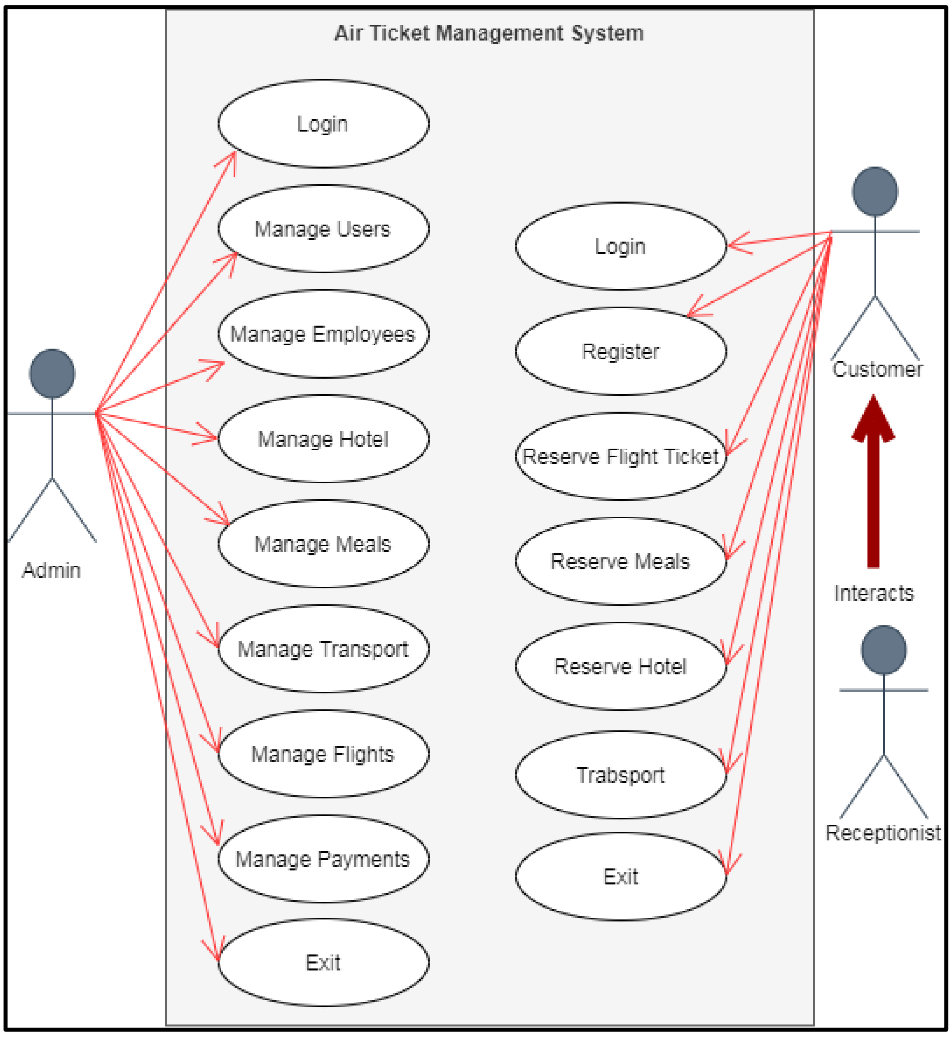


Figure 4 Use Case Diagram – Ticket Management System

CHAPTER 05

SYSTEM IMPLIMENTATION

5. SYSTEM IMPLIMENTATION

5.1 Interfaces

5.1.1. Loading Page

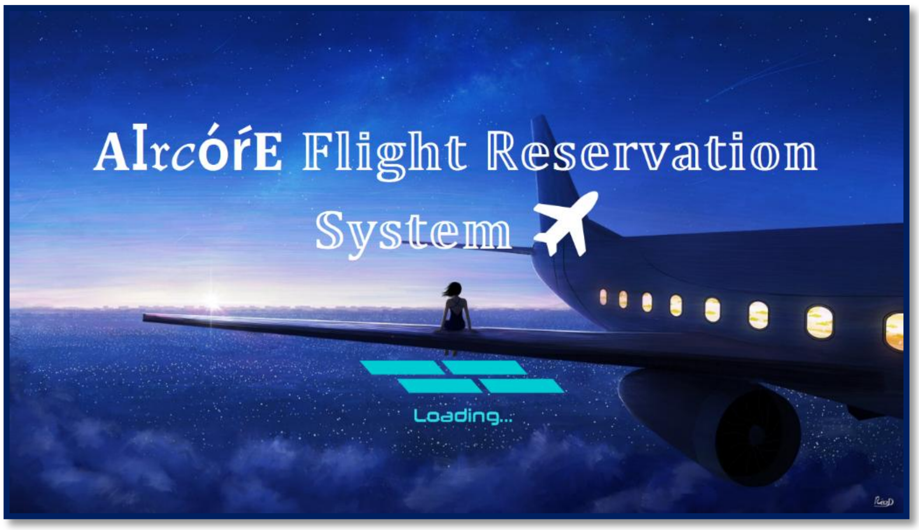


Figure 5 Loading Page

5.1.2. Admin Login

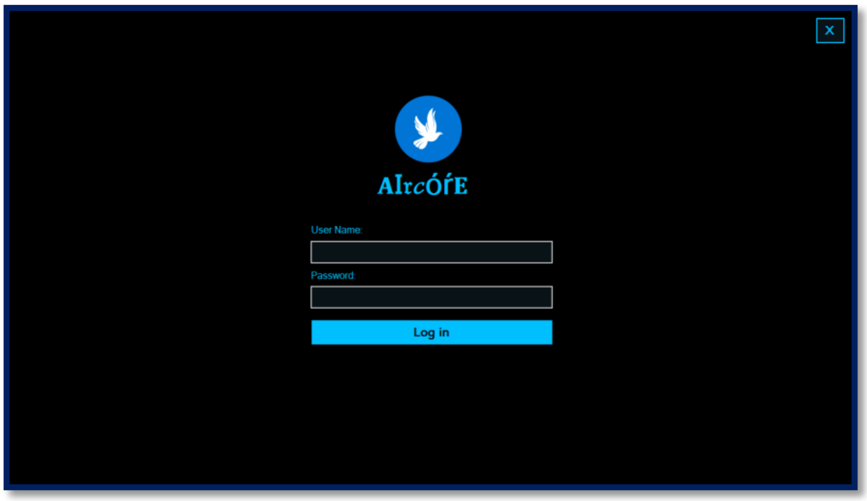


Figure 6 Admin Login

5.1.3. Admin & User Home Page

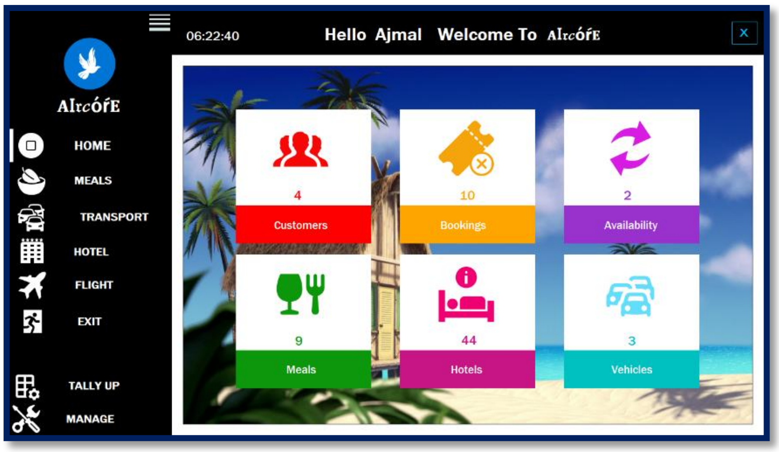


Figure 7 Admin Home Page

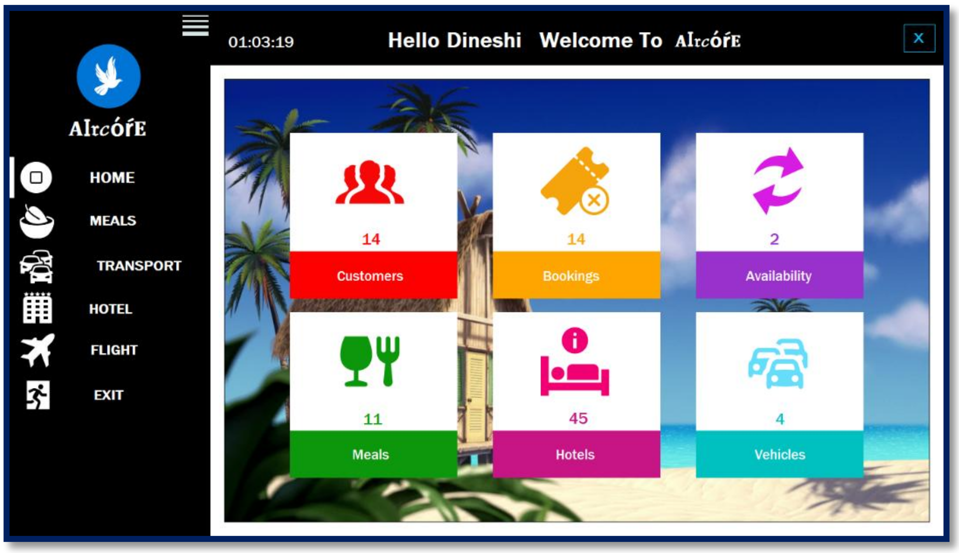


Figure 8 User Home Page

5.1.4. Customer Registration

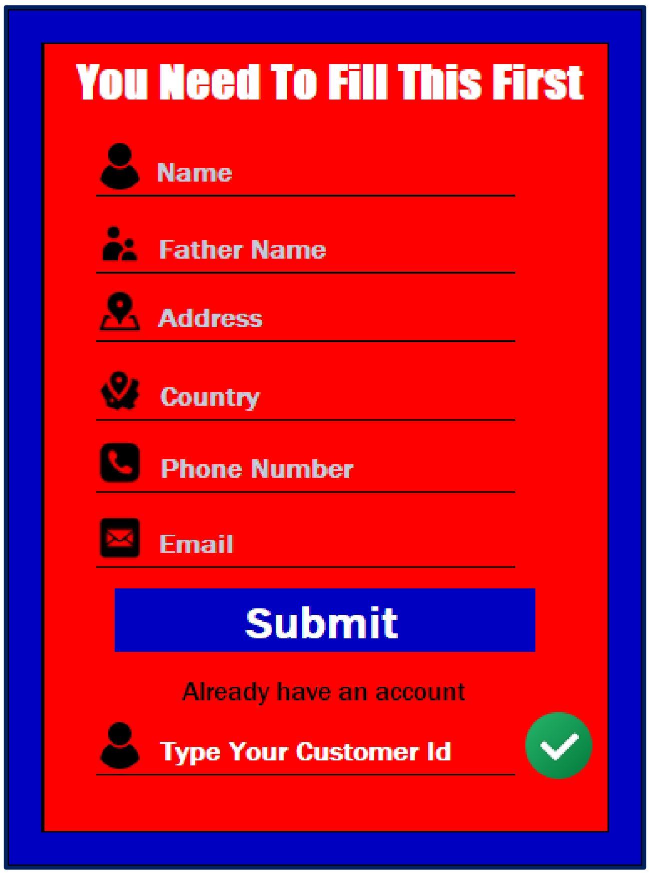
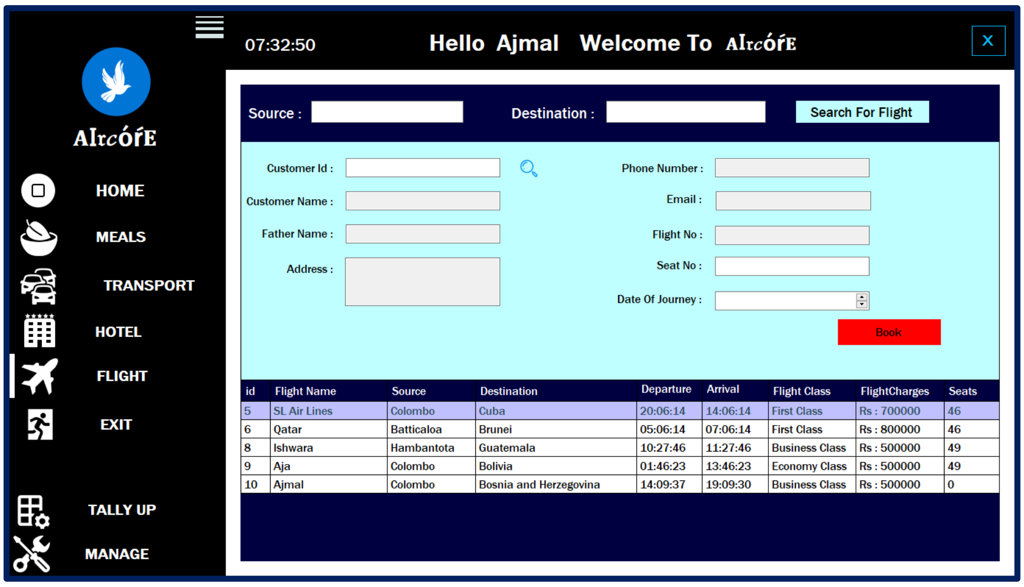


Figure 9 Customer Registration

5.1.5. Flight

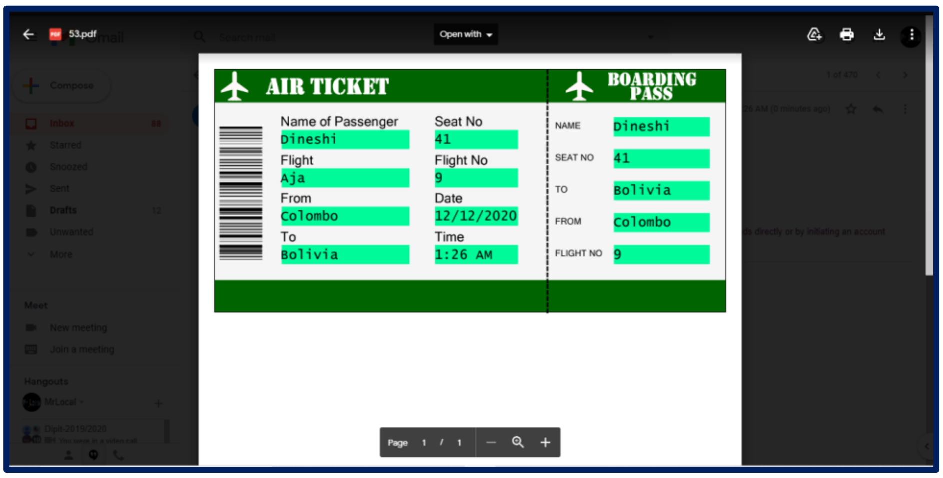


07:32:50 Hello Ajmal Welcome To AIrcófE

id	Flight Name	Source	Destination	Departure	Arrival	Flight Class	FlightCharges	Seats
5	SL Air Lines	Colombo	Cuba	20:06:14	14:06:14	First Class	Rs : 700000	46
6	Qatar	Batticaloa	Brunei	05:06:14	07:06:14	First Class	Rs : 800000	46
8	Ishwara	Hambantota	Guatemala	10:27:46	11:27:46	Business Class	Rs : 500000	49
9	Aja	Colombo	Bolivia	01:46:23	13:46:23	Economy Class	Rs : 500000	49
10	Ajmal	Colombo	Bosnia and Herzegovina	14:09:37	19:09:30	Business Class	Rs : 500000	0

Figure 10 Fight

5.1.6. Flight Ticket (Customer)


AIR TICKET

Name of Passenger: Dineshi
Seat No: 41
Flight: Aja
Flight No: 9
From: Colombo
Date: 12/12/2020
To: Bolivia
Time: 1:26 AM

BOARDING PASS

NAME: Dineshi
SEAT NO: 41
TO: Bolivia
FROM: Colombo
FLIGHT NO: 9

Figure 11 Flight Ticket

5.1.7. Loading Page for Hotel View

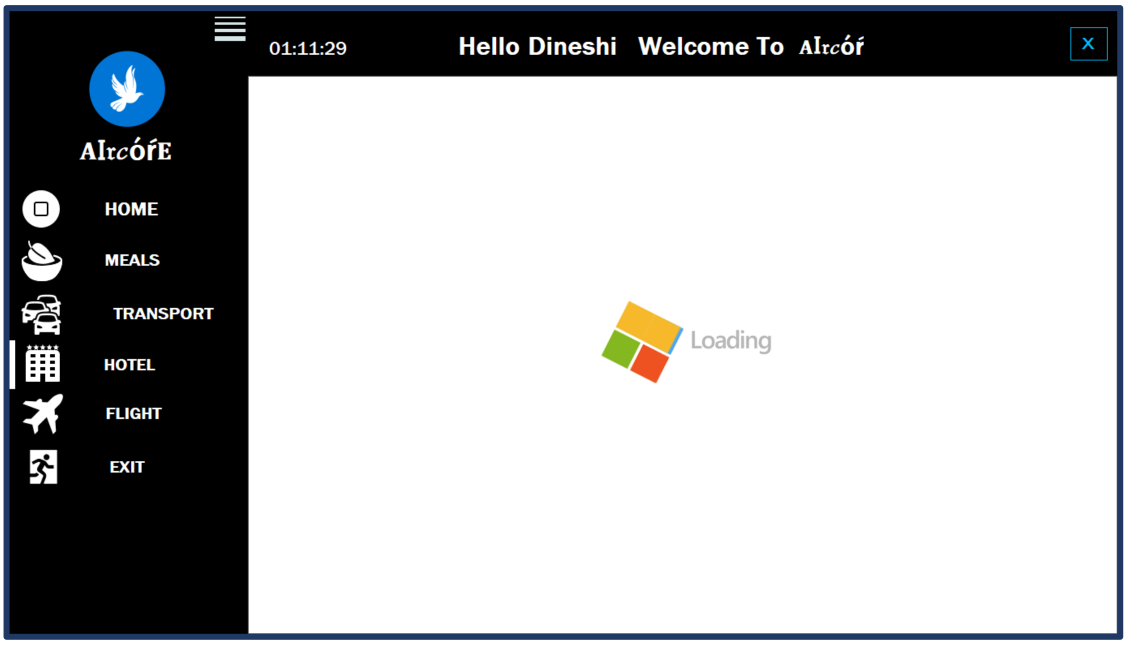
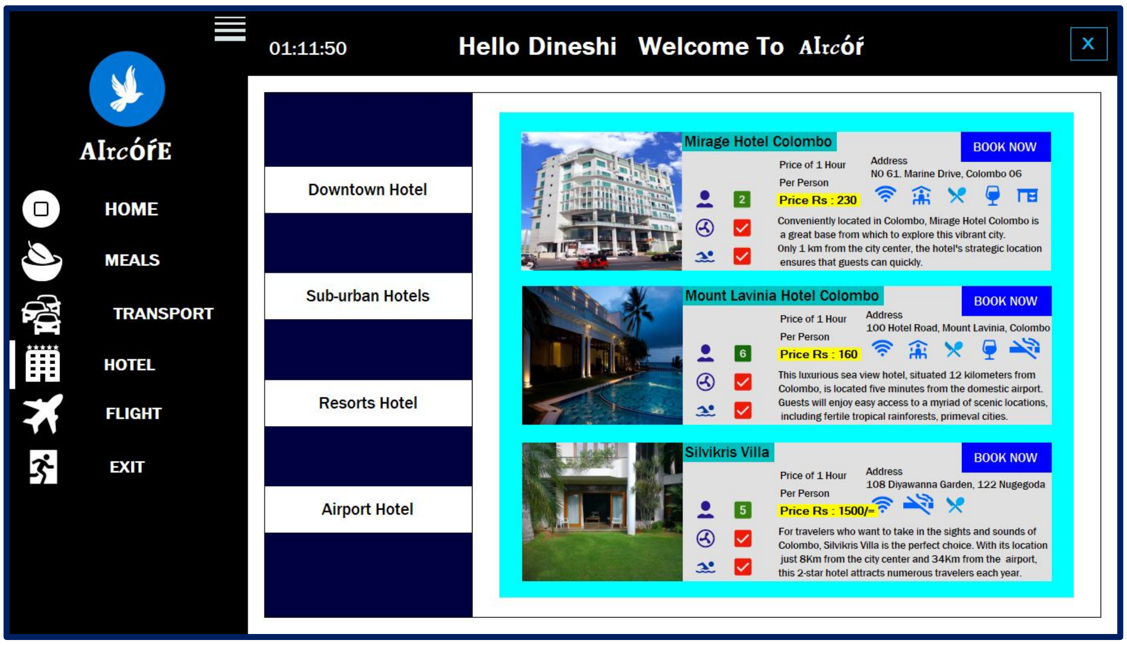

01:11:29 Hello Dineshi Welcome To AIrcóŕ

Figure 12 Loading Page for Hotel View

5.1.8. View of Hotels


01:11:50 Hello Dineshi Welcome To AIrcóŕ

Mirage Hotel Colombo

Price of 1 Hour Per Person
Price Rs : 230
Address
NO 61. Marine Drive, Colombo 06
2

Conveniently located in Colombo, Mirage Hotel Colombo is a great base from which to explore this vibrant city. Only 1 km from the city center, the hotel's strategic location ensures that guests can quickly.

Mount Lavinia Hotel Colombo

Price of 1 Hour Per Person
Price Rs : 160
Address
100 Hotel Road, Mount Lavinia, Colombo
6

This luxurious sea view hotel, situated 12 kilometers from Colombo, is located five minutes from the domestic airport. Guests will enjoy easy access to a myriad of scenic locations, including fertile tropical rainforests, primeval cities.

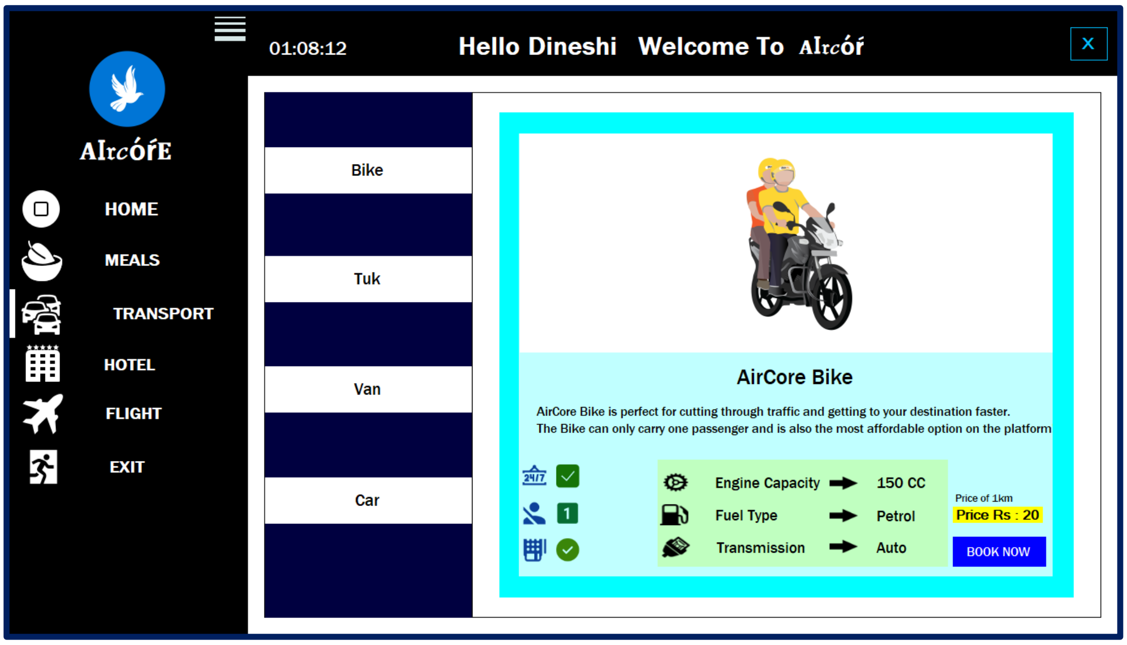
Silvikris Villa

Price of 1 Hour Per Person
Price Rs : 1500/=
Address
108 Diyawanna Garden, 122 Nugegoda
5

For travelers who want to take in the sights and sounds of Colombo, Silvikris Villa is the perfect choice. With its location just 8Km from the city center and 34Km from the airport, this 2-star hotel attracts numerous travelers each year.

Figure 13 View of Hotels

5.1.9. Transport Options


01:08:12 Hello Dineshi Welcome To AIrcóŕ

AirCore Bike

AirCore Bike is perfect for cutting through traffic and getting to your destination faster.
The Bike can only carry one passenger and is also the most affordable option on the platform

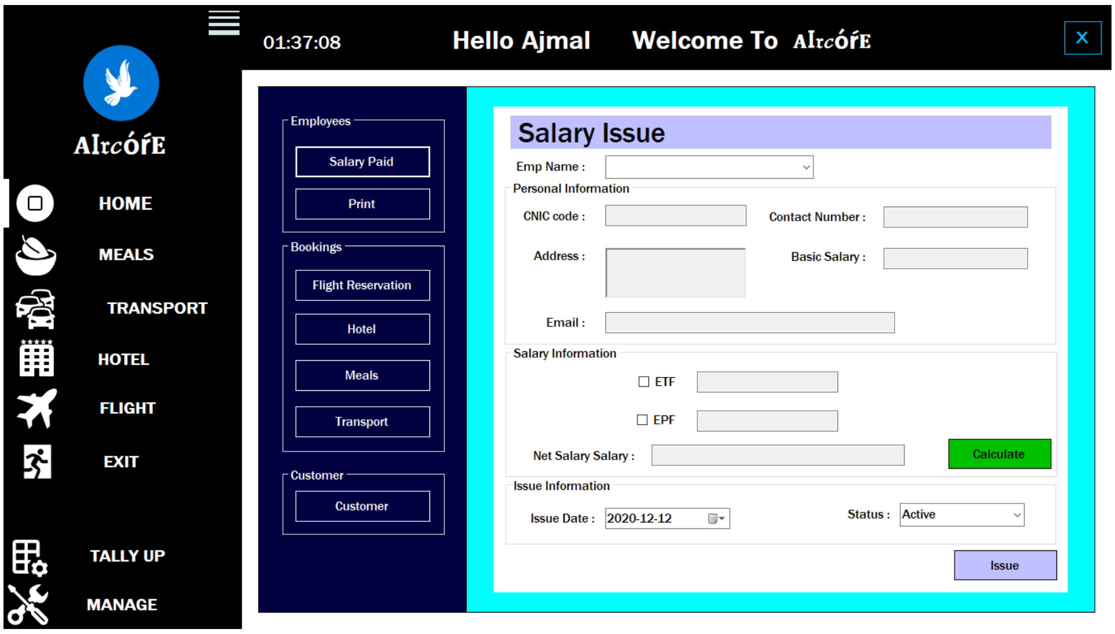
Engine Capacity	150 CC
Fuel Type	Petrol
Transmission	Auto

1

Price of 1km
Price Rs : 20

Figure 14 Transport Options

5.1.10. Issue Salary



Emp Name :

CNIC code :

Contact Number :

Address :

Basic Salary :

Email :

☐ ETF

☐ EPF

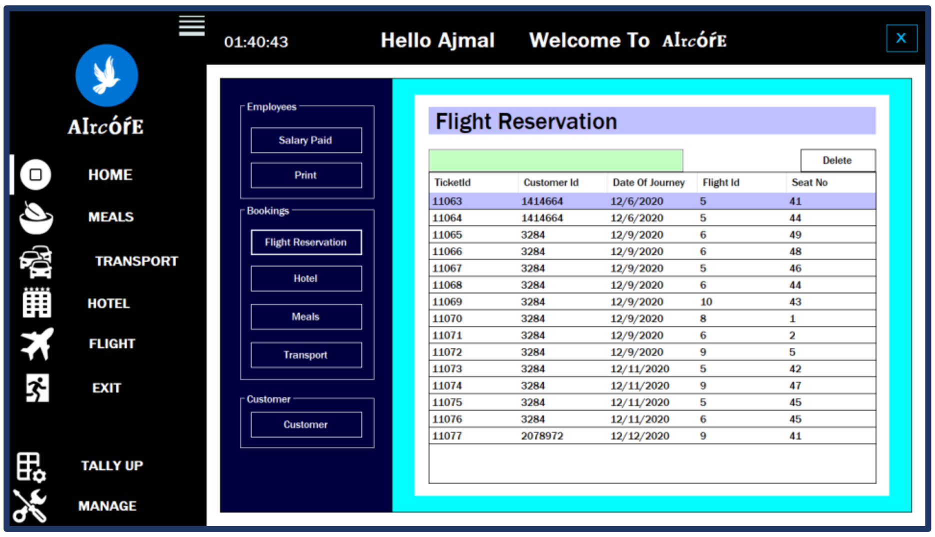
Net Salary Salary :

Issue Date : 2020-12-12

Status : Active

Figure 15 Issue Salary

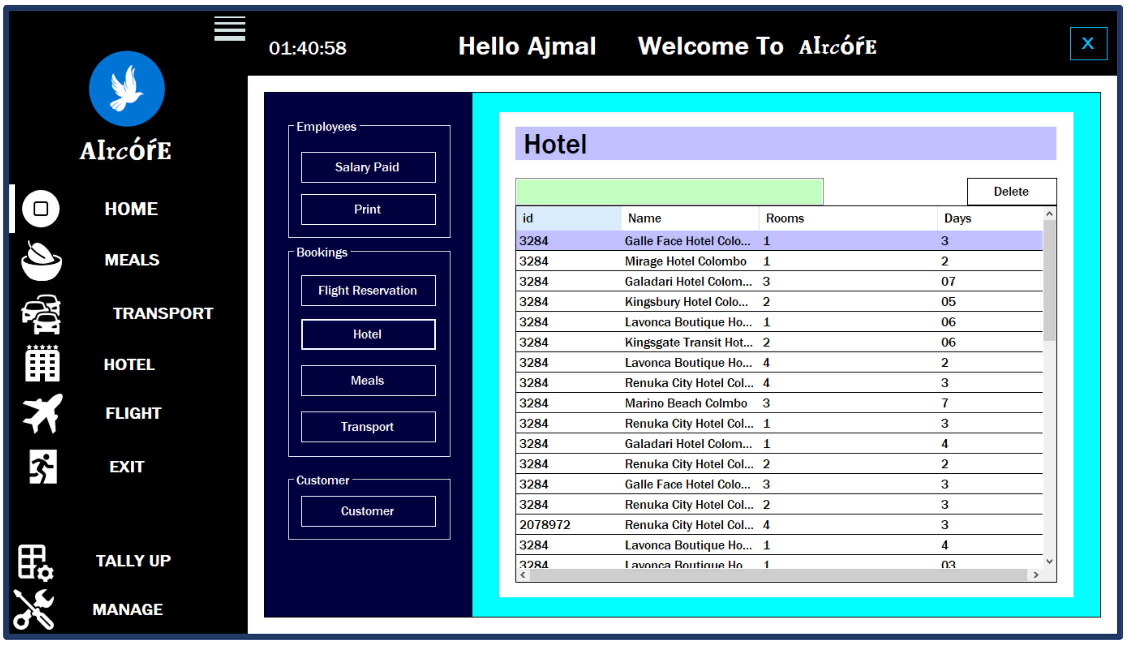
5.1.11. Report of Flights



TicketId	Customer Id	Date Of Journey	Flight Id	Seat No
11063	1414664	12/6/2020	5	41
11064	1414664	12/6/2020	5	44
11065	3284	12/9/2020	6	49
11066	3284	12/9/2020	6	48
11067	3284	12/9/2020	5	46
11068	3284	12/9/2020	6	44
11069	3284	12/9/2020	10	43
11070	3284	12/9/2020	8	1
11071	3284	12/9/2020	6	2
11072	3284	12/9/2020	9	5
11073	3284	12/11/2020	5	42
11074	3284	12/11/2020	9	47
11075	3284	12/11/2020	5	45
11076	3284	12/11/2020	6	45
11077	2078972	12/12/2020	9	41

Figure 16 Report of Flights

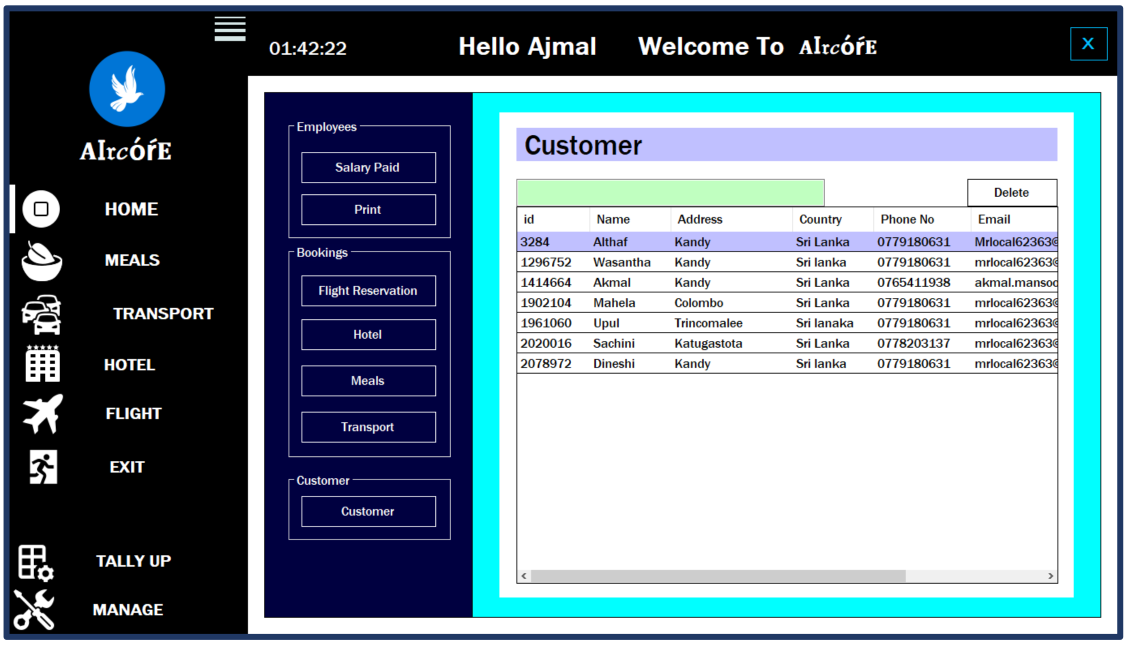
5.1.12. Report of Hotel



id	Name	Rooms	Days
3284	Galle Face Hotel Colo...	1	3
3284	Mirage Hotel Colombo	1	2
3284	Galadari Hotel Colom...	3	07
3284	Kingsbury Hotel Colo...	2	05
3284	Lavonca Boutique Ho...	1	06
3284	Kingsgate Transit Hot...	2	06
3284	Lavonca Boutique Ho...	4	2
3284	Renuka City Hotel Col...	4	3
3284	Marino Beach Colmbo	3	7
3284	Renuka City Hotel Col...	1	3
3284	Galadari Hotel Colom...	1	4
3284	Renuka City Hotel Col...	2	2
3284	Galle Face Hotel Colo...	3	3
3284	Renuka City Hotel Col...	2	3
2078972	Renuka City Hotel Col...	4	3
3284	Lavonca Boutique Ho...	1	4
3284	Lavonca Boutique Ho	1	03

Figure 17 Report of Hotels

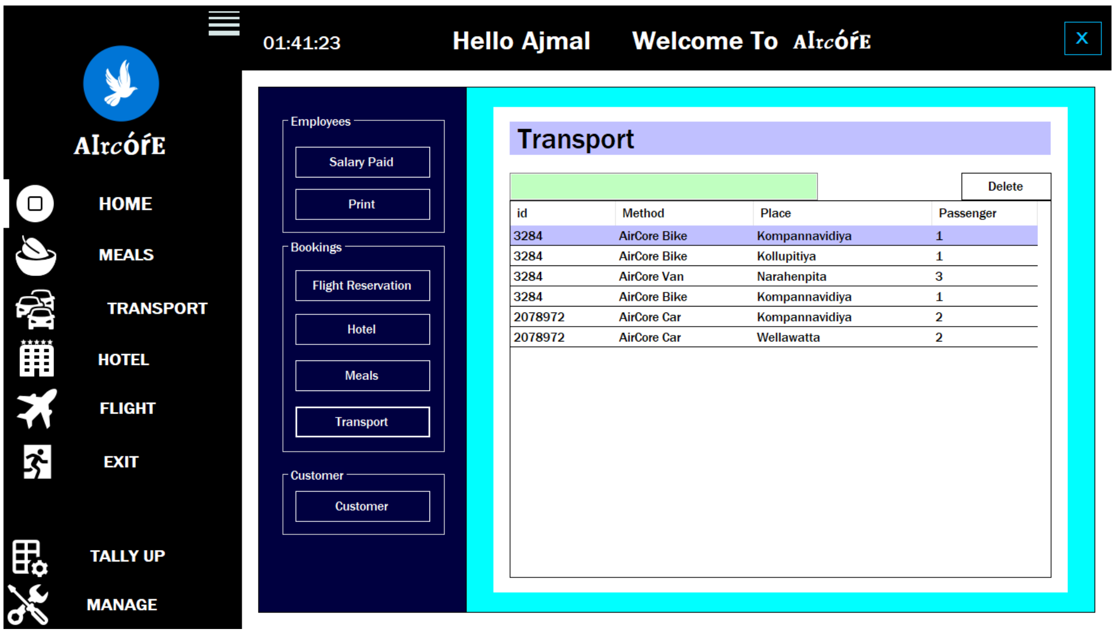
5.1.13. Report of Customer



id	Name	Address	Country	Phone No	Email
3284	Althaf	Kandy	Sri Lanka	0779180631	Mrlocal62363@
1296752	Wasantha	Kandy	Sri lanka	0779180631	mrlocal62363@
1414664	Akmal	Kandy	Sri Lanka	0765411938	akmal.manso
1902104	Mahela	Colombo	Sri Lanka	0779180631	mrlocal62363@
1961060	Upul	Trincomalee	Sri lanaka	0779180631	mrlocal62363@
2020016	Sachini	Katugastota	Sri Lanka	0778203137	mrlocal62363@
2078972	Dineshi	Kandy	Sri lanka	0779180631	mrlocal62363@

Figure 18 Report of Customer

5.1.14. Report of Transport

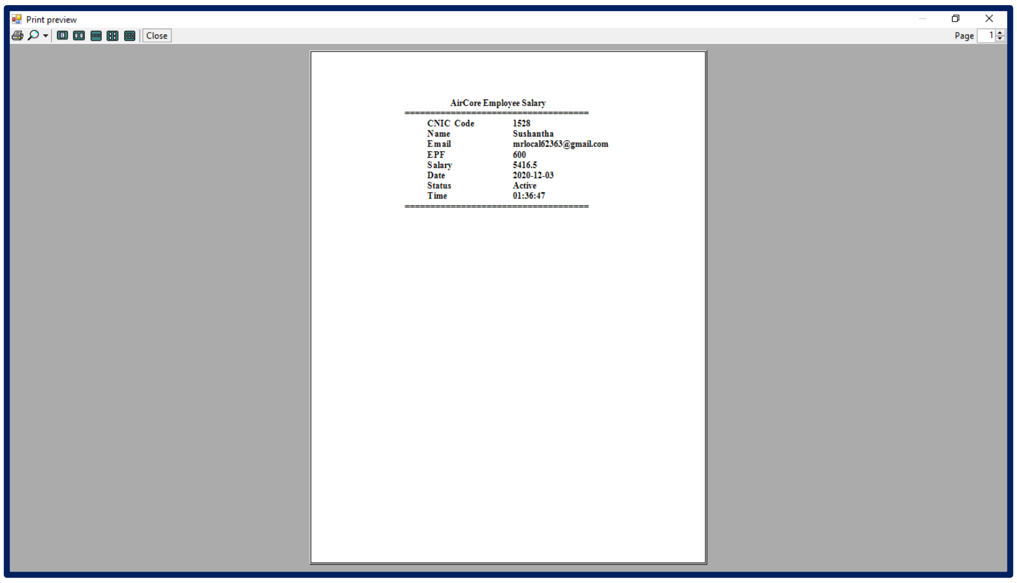


Transport

id	Method	Place	Passenger
3284	AirCore Bike	Kompannavidiya	1
3284	AirCore Bike	Kollupitiya	1
3284	AirCore Van	Narahenpita	3
3284	AirCore Bike	Kompannavidiya	1
2078972	AirCore Car	Kompannavidiya	2
2078972	AirCore Car	Wellawatta	2

Figure 19 Report of Transport

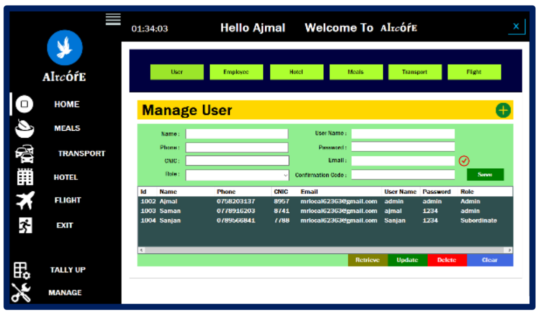
5.1.15. Print Preview of Employee Salary Receipt



```
           AirCore Employee Salary
====================================
     CNIC  Code          1528
     Name                Sushantha
     Email               mrlocal62363@gmail.com
     EPF                 600
     Salary              5416.5
     Date                2020-12-03
     Status              Active
     Time                01:36:47
====================================
```

Figure 20 Employee Salary Receipt

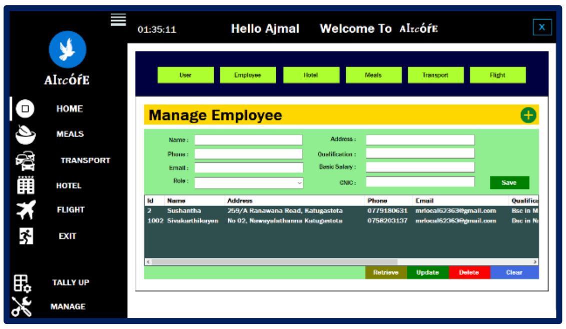
5.1.16. Manage Users



Id	Name	Phone	CNIC	Email	User Name	Password	Role
1002	Ajmal	0758203137	8957	mrlocal62363@gmail.com	admin	admin	Admin
1003	Saman	0778916203	8741	mrlocal62363@gmail.com	ajmal	1234	admin
1004	Sanjan	0789566841	7788	mrlocal62363@gmail.com	Sanjan	1234	Subordinate

Figure 21 Manage Users

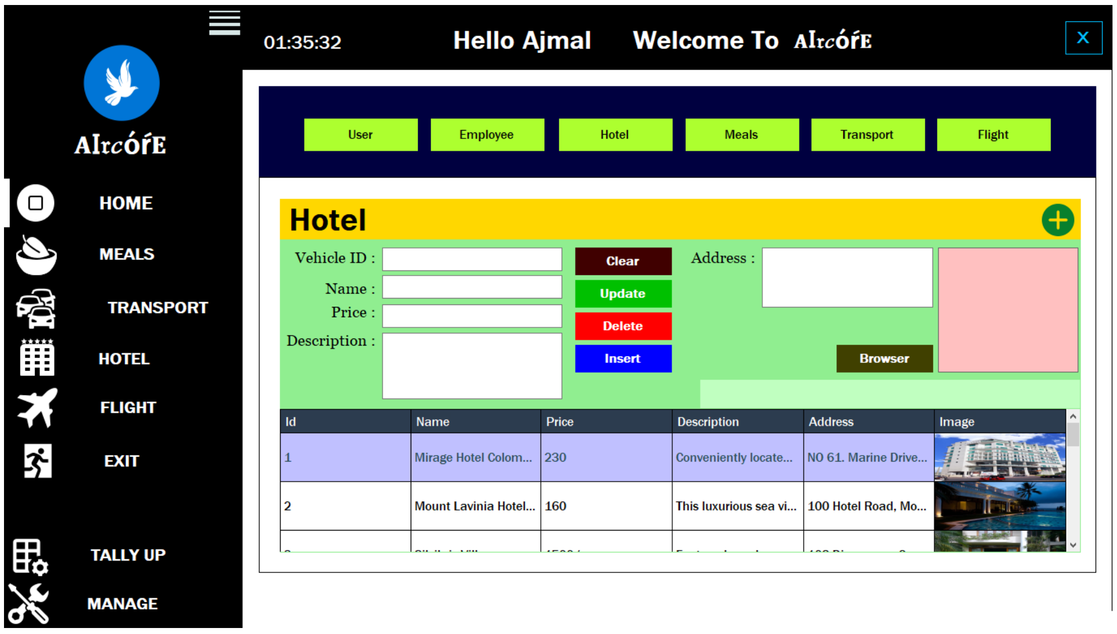
5.1.17. Manage Employee



Id	Name	Address	Phone	Email	Qualifica
2	Sushantha	259/A Ranawana Road, Katugastota	0779180631	mrlocal62363@gmail.com	Bsc in M
1002	Sivakarthikayen	No 02, Nawayalathanna Katugastota	0758203137	mrlocal62363@gmail.com	Bsc in N

Figure 22 Manage Employee

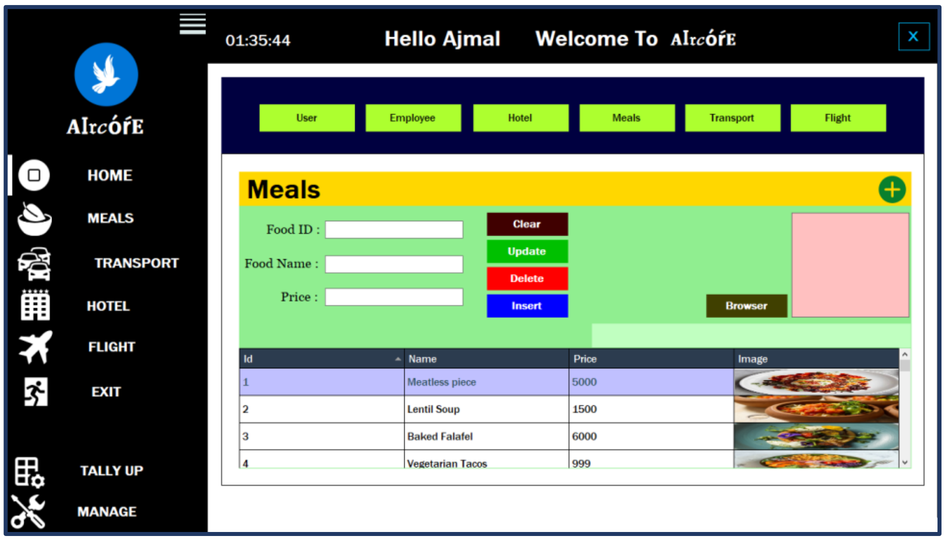
5.1.18. Manage Hotel



Id	Name	Price	Description	Address	Image
1	Mirage Hotel Colom...	230	Conveniently locate...	NO 61. Marine Drive...	
2	Mount Lavinia Hotel...	160	This luxurious sea vi...	100 Hotel Road, Mo...	

Figure 23 Manage Hotel

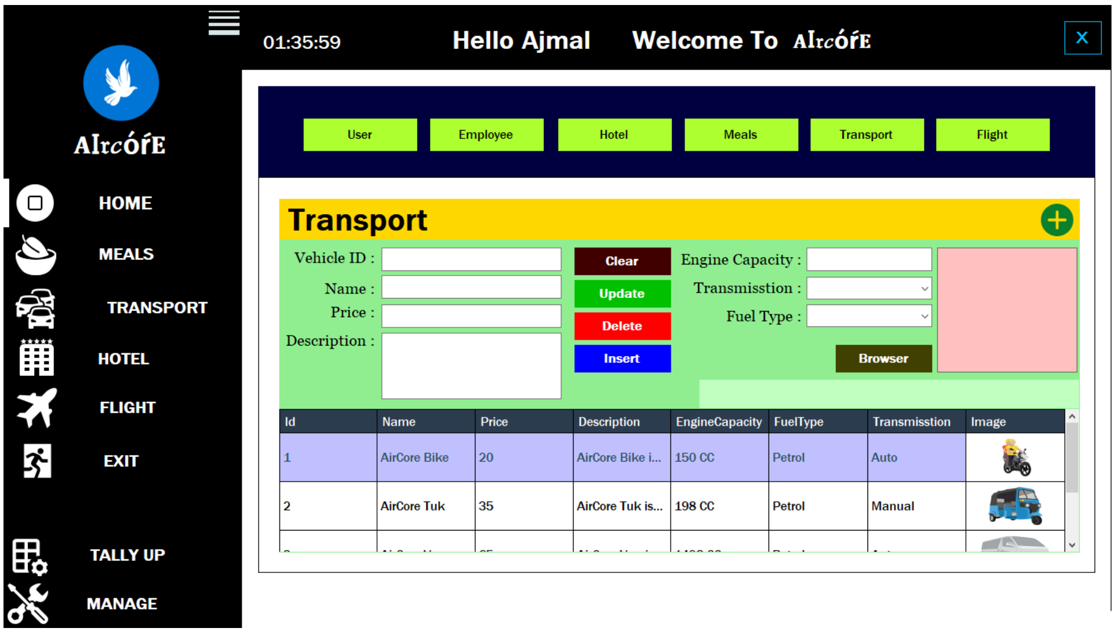
5.1.19. Manage Meals



Id	Name	Price	Image
1	Meatless piece	5000	
2	Lentil Soup	1500	
3	Baked Falafel	6000	
4	Vegetarian Tacos	999	

Figure 24 Manage Meals

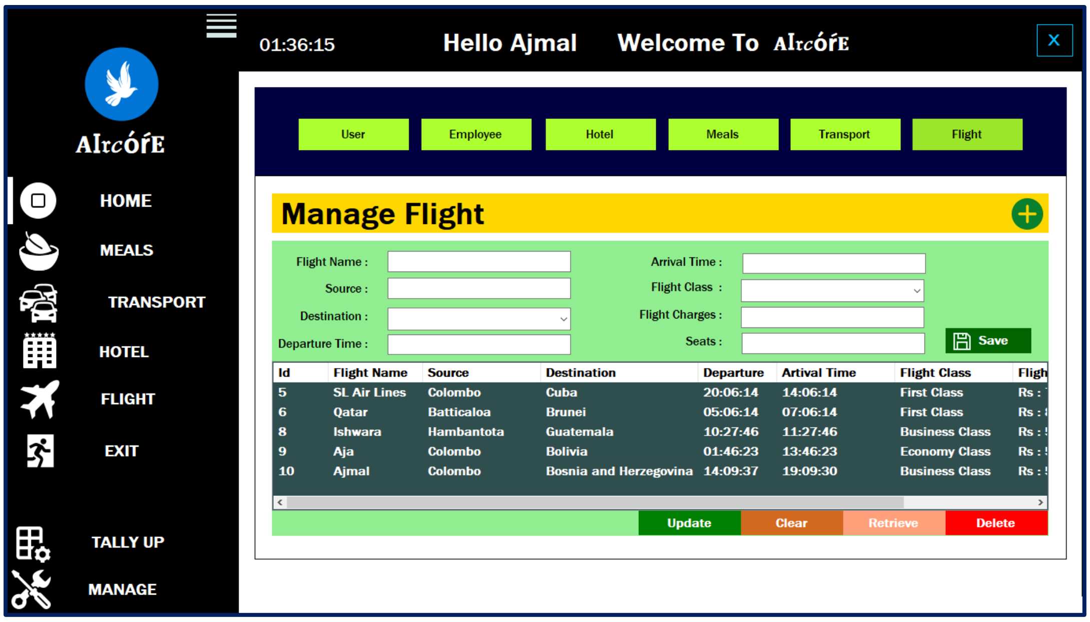
5.1.20. Manage Transport



Id	Name	Price	Description	EngineCapacity	FuelType	Transmisstion	Image
1	AirCore Bike	20	AirCore Bike i...	150 CC	Petrol	Auto	
2	AirCore Tuk	35	AirCore Tuk is...	198 CC	Petrol	Manual	
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	

Figure 25 Manage Transport

5.1.21. Manage Flight



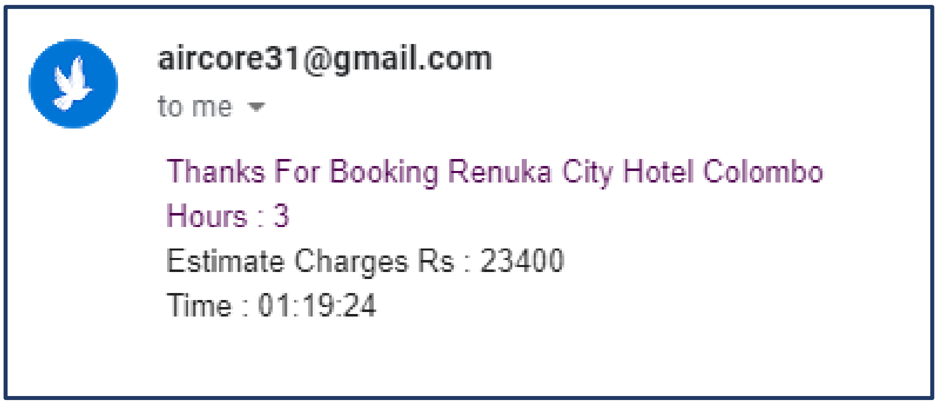
Manage Flight

Flight Name :
Source :
Destination :
Departure Time :
Arrival Time :
Flight Class :
Flight Charges :
Seats :

Id	Flight Name	Source	Destination	Departure	Artival Time	Flight Class	Fligh
5	SL Air Lines	Colombo	Cuba	20:06:14	14:06:14	First Class	Rs :
6	Qatar	Batticaloa	Brunei	05:06:14	07:06:14	First Class	Rs :
8	Ishwara	Hambantota	Guatemala	10:27:46	11:27:46	Business Class	Rs :
9	Aja	Colombo	Bolivia	01:46:23	13:46:23	Economy Class	Rs :
10	Ajmal	Colombo	Bosnia and Herzegovina	14:09:37	19:09:30	Business Class	Rs :

Figure 26 Manage Flight

5.1.22. Email view for hotel booking

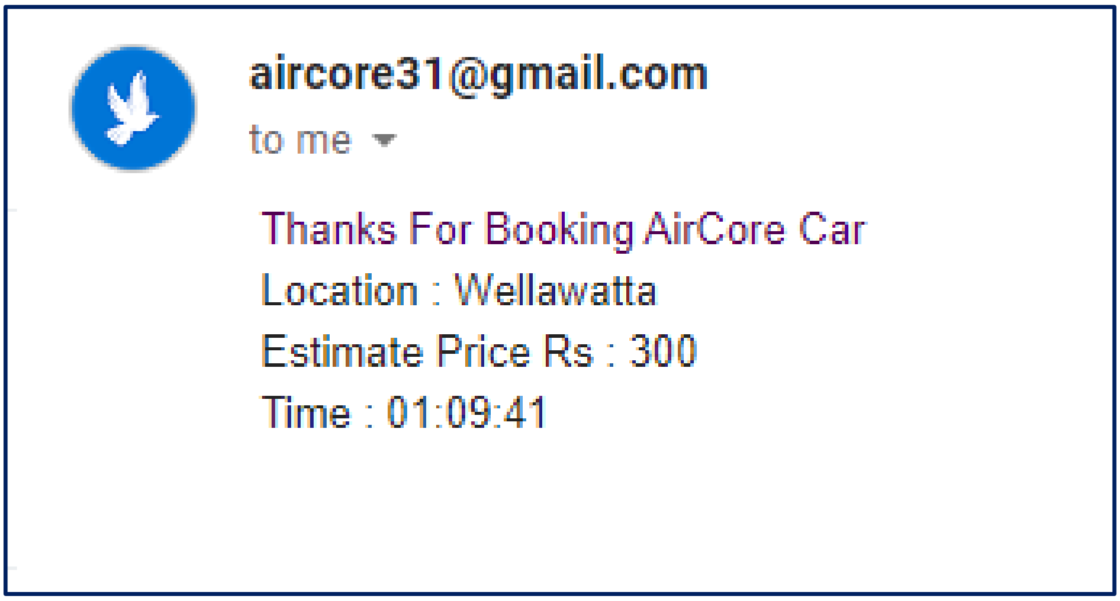


aircore31@gmail.com
to me

Thanks For Booking Renuka City Hotel Colombo
Hours : 3
Estimate Charges Rs : 23400
Time : 01:19:24

Figure 27 Email View for hotel booking

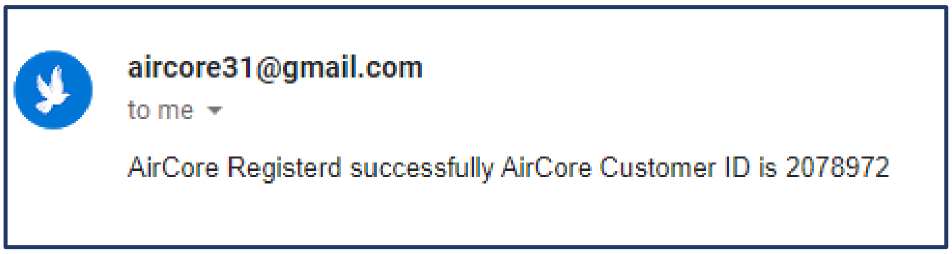
5.1.23. Email view for hiring vehicles


aircore31@gmail.com
to me

Thanks For Booking AirCore Car
Location : Wellawatta
Estimate Price Rs : 300
Time : 01:09:41

Figure 28 Email view for hiring vehicles

5.1.24. Email message registered customers

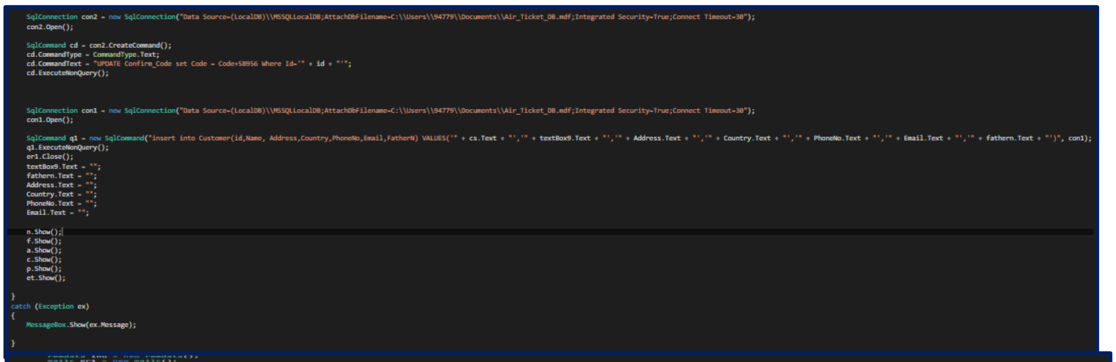

aircore31@gmail.com
to me

AirCore Registerd successfully AirCore Customer ID is 2078972

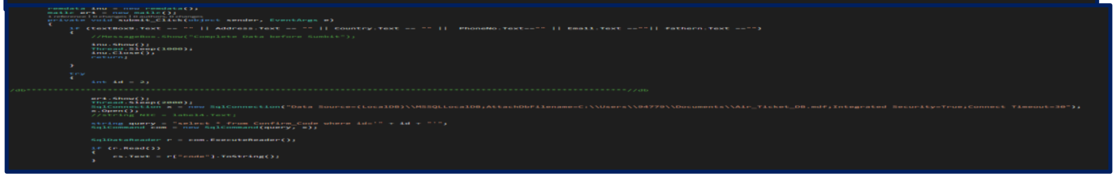
Figure 29 Email message for registered Customers

5.2 Coding (Selected)

5.2.1. Coding for Customer Module



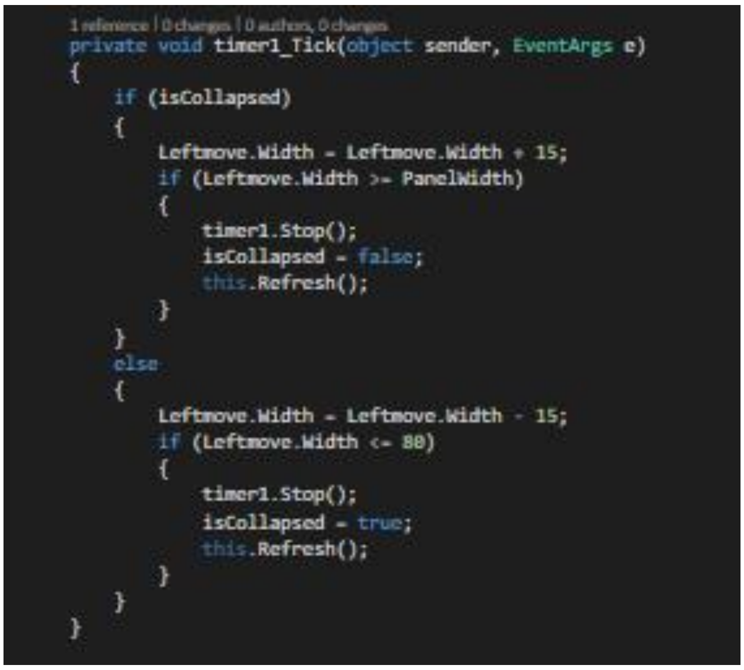
```csharp
    SqlConnection con2 = new SqlConnection("Data Source=(LocalDB)\\MSSQLLocalDB;AttachDbFilename=C:\\Users\\94779\\Documents\\Air_Ticket_DB.mdf;Integrated Security=True;Connect Timeout=30");
    con2.Open();

    SqlCommand cd = con2.CreateCommand();
    cd.CommandType = CommandType.Text;
    cd.CommandText = "UPDATE Confirm_Code set Code = Code+58956 Where Id='" + id + "'";
    cd.ExecuteNonQuery();



    SqlConnection con1 = new SqlConnection("Data Source=(LocalDB)\\MSSQLLocalDB;AttachDbFilename=C:\\Users\\94779\\Documents\\Air_Ticket_DB.mdf;Integrated Security=True;Connect Timeout=30");
    con1.Open();

    SqlCommand q1 = new SqlCommand("insert into Customer(id,Name, Address,Country,PhoneNo,Email,FatherN) VALUES('" + cs.Text + "','" + textBox9.Text + "','" + Address.Text + "','" + Country.Text + "','" + PhoneNo.Text + "','" + Email.Text + "','" + fathern.Text + "')", con1);
    q1.ExecuteNonQuery();
    er1.Close();
    textBox9.Text = "";
    fathern.Text = "";
    Address.Text = "";
    Country.Text = "";
    PhoneNo.Text = "";
    Email.Text = "";

    n.Show();
    f.Show();
    a.Show();
    c.Show();
    p.Show();
    et.Show();

}
catch (Exception ex)
{
    MessageBox.Show(ex.Message);

}
```



```csharp
        remdata inu = new remdata();
        maile er1 = new maile();
        1 reference | 0 changes | 0 authors, 0 changes
        private void submit_Click(object sender, EventArgs e)
        {
            if (textBox9.Text == "" || Address.Text == "" || Country.Text == "" ||  PhoneNo.Text=="" || Email.Text ==""|| fathern.Text =="")
            {
                //MessageBox.Show("Complete Data before Sumbit");

                inu.Show();
                Thread.Sleep(1000);
                inu.Close();
                return;

            }

            try
            {
                int id = 2;
/db****************************************************************************************//db

                er1.Show();
                Thread.Sleep(2000);
                SqlConnection x = new SqlConnection("Data Source=(LocalDB)\\MSSQLLocalDB;AttachDbFilename=C:\\Users\\94779\\Documents\\Air_Ticket_DB.mdf;Integrated Security=True;Connect Timeout=30");
                x.Open();
                //string NIC = label4.Text;

                string query = "select * from Confirm_Code where id='" + id + "'";
                SqlCommand com = new SqlCommand(query, x);

                SqlDataReader r = com.ExecuteReader();

                if (r.Read())
                {
                    cs.Text = r["code"].ToString();
                }
```

Figure 30 Coding for customer module

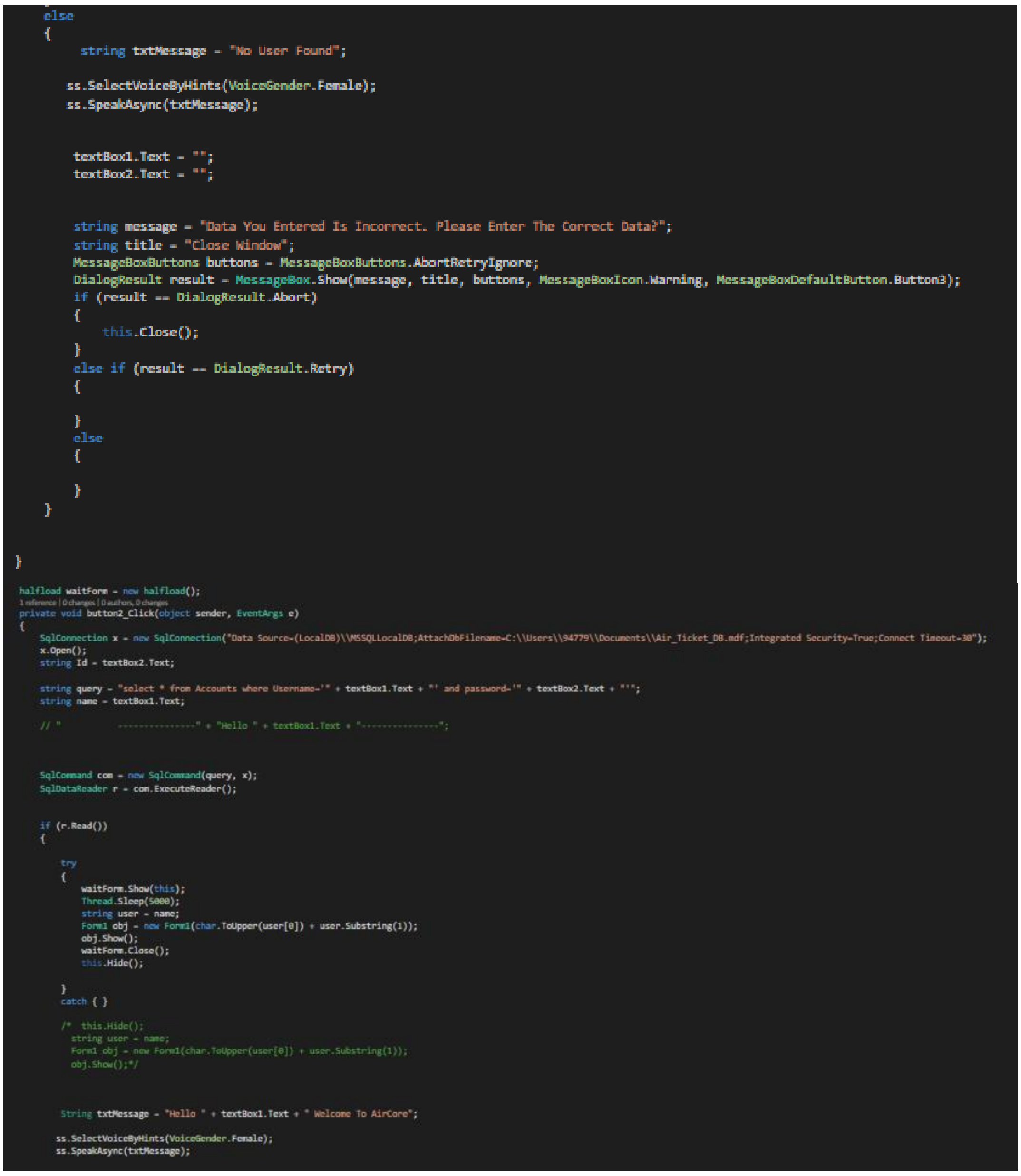
5.2.2. Coding for Dashboard



```csharp
1 reference | 0 changes | 0 authors, 0 changes
private void timer1_Tick(object sender, EventArgs e)
{
    if (isCollapsed)
    {
        Leftmove.Width = Leftmove.Width + 15;
        if (Leftmove.Width >= PanelWidth)
        {
            timer1.Stop();
            isCollapsed = false;
            this.Refresh();
        }
    }
    else
    {
        Leftmove.Width = Leftmove.Width - 15;
        if (Leftmove.Width <= 80)
        {
            timer1.Stop();
            isCollapsed = true;
            this.Refresh();
        }
    }
}
```

Figure 31 Coding for dashboard

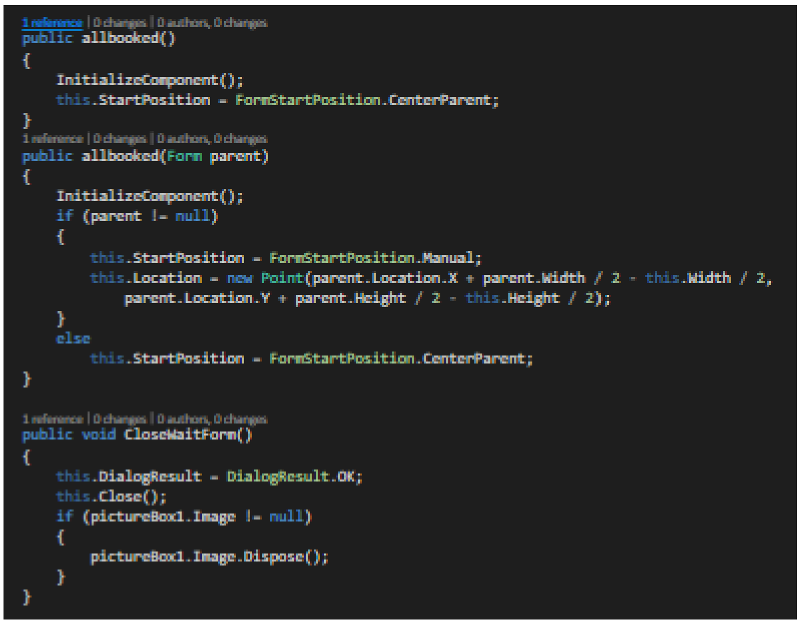
5.2.3. Coding for Login



```
    else
    {
         string txtMessage = "No User Found";

       ss.SelectVoiceByHints(VoiceGender.Female);
       ss.SpeakAsync(txtMessage);


        textBox1.Text = "";
        textBox2.Text = "";


        string message = "Data You Entered Is Incorrect. Please Enter The Correct Data?";
        string title = "Close Window";
        MessageBoxButtons buttons = MessageBoxButtons.AbortRetryIgnore;
        DialogResult result = MessageBox.Show(message, title, buttons, MessageBoxIcon.Warning, MessageBoxDefaultButton.Button3);
        if (result == DialogResult.Abort)
        {
            this.Close();
        }
        else if (result == DialogResult.Retry)
        {

        }
        else
        {

        }
    }


}
halfload waitForm = new halfload();
1 reference | 0 changes | 0 authors, 0 changes
private void button2_Click(object sender, EventArgs e)
{
    SqlConnection x = new SqlConnection("Data Source=(LocalDB)\\MSSQLLocalDB;AttachDbFilename=C:\\Users\\94779\\Documents\\Air_Ticket_DB.mdf;Integrated Security=True;Connect Timeout=30");
    x.Open();
    string Id = textBox2.Text;

    string query = "select * from Accounts where Username='" + textBox1.Text + "' and password='" + textBox2.Text + "'";
    string name = textBox1.Text;

    // "          ---------------" + "Hello " + textBox1.Text + "---------------";



    SqlCommand com = new SqlCommand(query, x);
    SqlDataReader r = com.ExecuteReader();


    if (r.Read())
    {

        try
        {
            waitForm.Show(this);
            Thread.Sleep(5000);
            string user = name;
            Form1 obj = new Form1(char.ToUpper(user[0]) + user.Substring(1));
            obj.Show();
            waitForm.Close();
            this.Hide();

        }
        catch { }

        /*  this.Hide();
          string user = name;
          Form1 obj = new Form1(char.ToUpper(user[0]) + user.Substring(1));
          obj.Show();*/



        String txtMessage = "Hello " + textBox1.Text + " Welcome To AirCore";

       ss.SelectVoiceByHints(VoiceGender.Female);
       ss.SpeakAsync(txtMessage);
```

Figure 32 Coding for Login

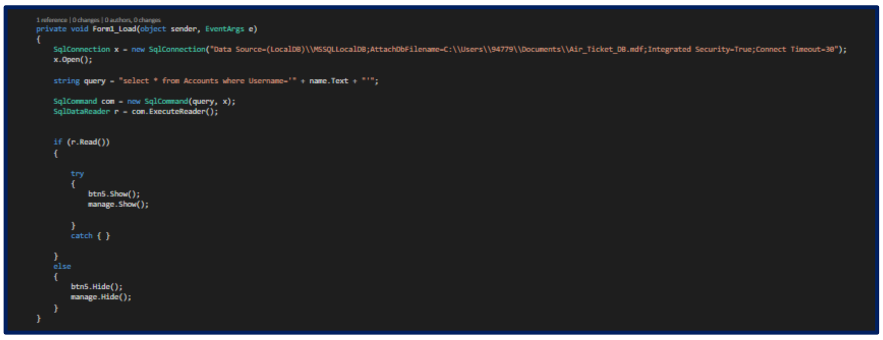
5.2.4. Coding for Message Box



```
1 reference | 0 changes | 0 authors, 0 changes
public allbooked()
{
    InitializeComponent();
    this.StartPosition = FormStartPosition.CenterParent;
}
1 reference | 0 changes | 0 authors, 0 changes
public allbooked(Form parent)
{
    InitializeComponent();
    if (parent != null)
    {
        this.StartPosition = FormStartPosition.Manual;
        this.Location = new Point(parent.Location.X + parent.Width / 2 - this.Width / 2,
            parent.Location.Y + parent.Height / 2 - this.Height / 2);
    }
    else
        this.StartPosition = FormStartPosition.CenterParent;
}

1 reference | 0 changes | 0 authors, 0 changes
public void CloseWaitForm()
{
    this.DialogResult = DialogResult.OK;
    this.Close();
    if (pictureBox1.Image != null)
    {
        pictureBox1.Image.Dispose();
    }
}
```

Figure 33 Coding for Message Box

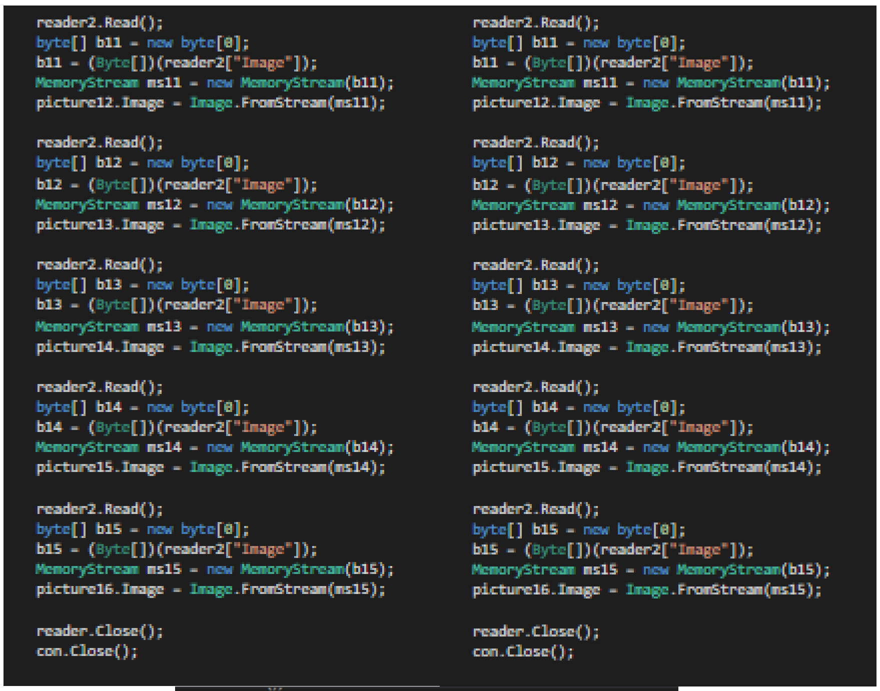
5.2.5. Coding for Login



```
1 reference | 0 changes | 0 authors, 0 changes
private void Form1_Load(object sender, EventArgs e)
{
    SqlConnection x = new SqlConnection("Data Source=(LocalDB)\\MSSQLLocalDB;AttachDbFilename=C:\\Users\\94779\\Documents\\Air_Ticket_DB.mdf;Integrated Security=True;Connect Timeout=30");
    x.Open();

    string query = "select * from Accounts where Username='" + name.Text + "'";

    SqlCommand com = new SqlCommand(query, x);
    SqlDataReader r = com.ExecuteReader();


    if (r.Read())
    {

        try
        {
            btn5.Show();
            manage.Show();

        }
        catch { }

    }
    else
    {
        btn5.Hide();
        manage.Hide();
    }
}
```

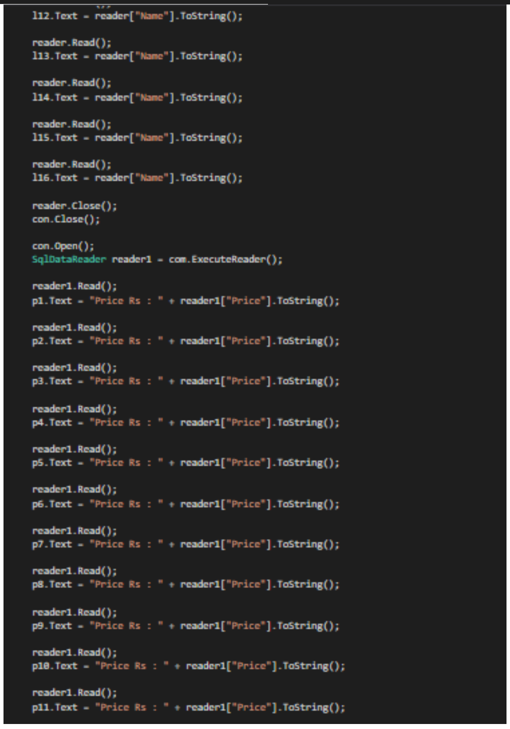
Figure 34 Coding for Login

5.2.5. Coding for Display Meals (Selected)



```
reader2.Read();
byte[] b11 = new byte[0];
b11 = (Byte[])(reader2["Image"]);
MemoryStream ms11 = new MemoryStream(b11);
picture12.Image = Image.FromStream(ms11);

reader2.Read();
byte[] b12 = new byte[0];
b12 = (Byte[])(reader2["Image"]);
MemoryStream ms12 = new MemoryStream(b12);
picture13.Image = Image.FromStream(ms12);

reader2.Read();
byte[] b13 = new byte[0];
b13 = (Byte[])(reader2["Image"]);
MemoryStream ms13 = new MemoryStream(b13);
picture14.Image = Image.FromStream(ms13);

reader2.Read();
byte[] b14 = new byte[0];
b14 = (Byte[])(reader2["Image"]);
MemoryStream ms14 = new MemoryStream(b14);
picture15.Image = Image.FromStream(ms14);

reader2.Read();
byte[] b15 = new byte[0];
b15 = (Byte[])(reader2["Image"]);
MemoryStream ms15 = new MemoryStream(b15);
picture16.Image = Image.FromStream(ms15);

reader.Close();
con.Close();
```

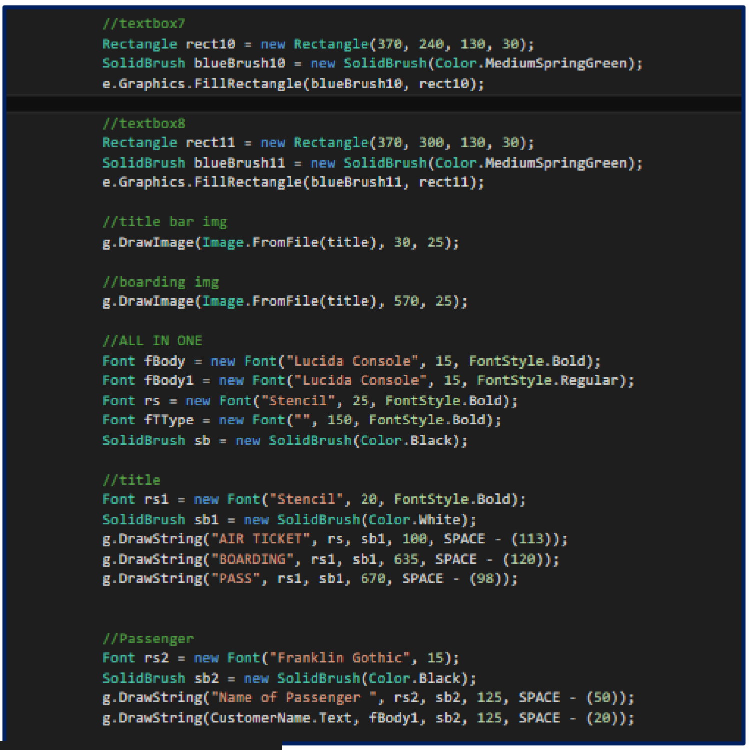
```
reader2.Read();
byte[] b11 = new byte[0];
b11 = (Byte[])(reader2["Image"]);
MemoryStream ms11 = new MemoryStream(b11);
picture12.Image = Image.FromStream(ms11);

reader2.Read();
byte[] b12 = new byte[0];
b12 = (Byte[])(reader2["Image"]);
MemoryStream ms12 = new MemoryStream(b12);
picture13.Image = Image.FromStream(ms12);

reader2.Read();
byte[] b13 = new byte[0];
b13 = (Byte[])(reader2["Image"]);
MemoryStream ms13 = new MemoryStream(b13);
picture14.Image = Image.FromStream(ms13);

reader2.Read();
byte[] b14 = new byte[0];
b14 = (Byte[])(reader2["Image"]);
MemoryStream ms14 = new MemoryStream(b14);
picture15.Image = Image.FromStream(ms14);

reader2.Read();
byte[] b15 = new byte[0];
b15 = (Byte[])(reader2["Image"]);
MemoryStream ms15 = new MemoryStream(b15);
picture16.Image = Image.FromStream(ms15);

reader.Close();
con.Close();
```



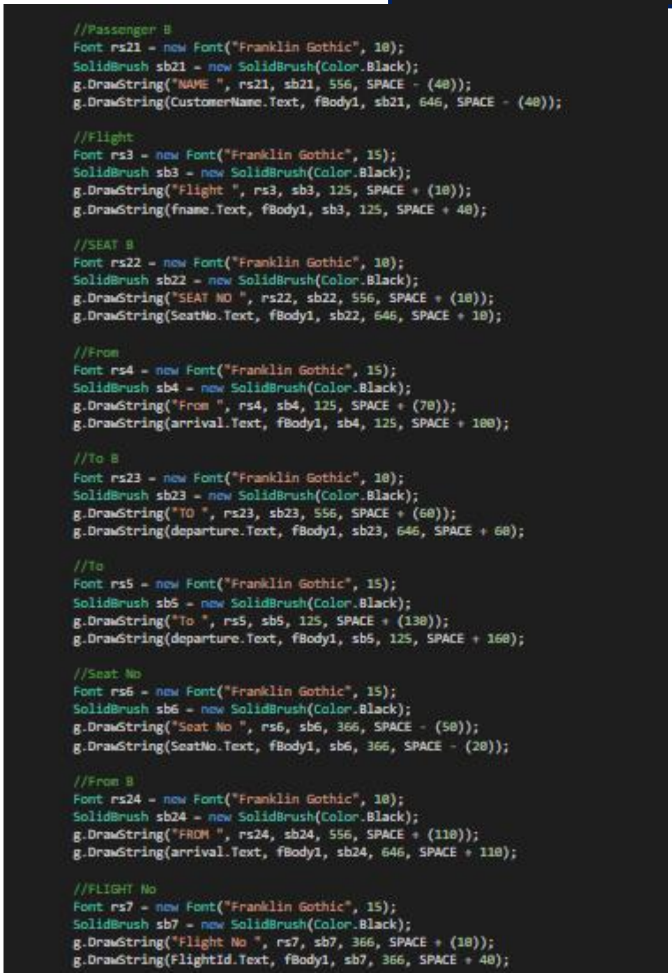
```
l12.Text = reader["Name"].ToString();

reader.Read();
l13.Text = reader["Name"].ToString();

reader.Read();
l14.Text = reader["Name"].ToString();

reader.Read();
l15.Text = reader["Name"].ToString();

reader.Read();
l16.Text = reader["Name"].ToString();

reader.Close();
con.Close();

con.Open();
SqlDataReader reader1 = com.ExecuteReader();

reader1.Read();
p1.Text = "Price Rs : " + reader1["Price"].ToString();

reader1.Read();
p2.Text = "Price Rs : " + reader1["Price"].ToString();

reader1.Read();
p3.Text = "Price Rs : " + reader1["Price"].ToString();

reader1.Read();
p4.Text = "Price Rs : " + reader1["Price"].ToString();

reader1.Read();
p5.Text = "Price Rs : " + reader1["Price"].ToString();

reader1.Read();
p6.Text = "Price Rs : " + reader1["Price"].ToString();

reader1.Read();
p7.Text = "Price Rs : " + reader1["Price"].ToString();

reader1.Read();
p8.Text = "Price Rs : " + reader1["Price"].ToString();

reader1.Read();
p9.Text = "Price Rs : " + reader1["Price"].ToString();

reader1.Read();
p10.Text = "Price Rs : " + reader1["Price"].ToString();

reader1.Read();
p11.Text = "Price Rs : " + reader1["Price"].ToString();
```

Figure 35 Coding for Display Meals

5.2.6. Coding for Flight Bookings (Selected)



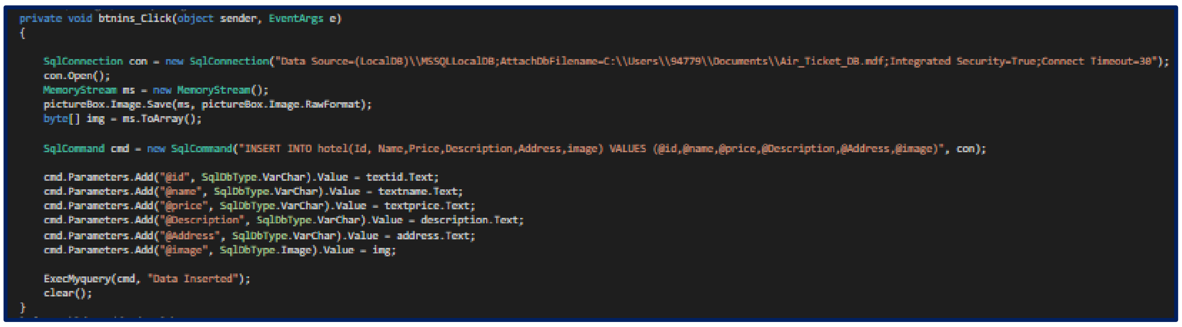
```
//textbox7
Rectangle rect10 = new Rectangle(370, 240, 130, 30);
SolidBrush blueBrush10 = new SolidBrush(Color.MediumSpringGreen);
e.Graphics.FillRectangle(blueBrush10, rect10);

//textbox8
Rectangle rect11 = new Rectangle(370, 300, 130, 30);
SolidBrush blueBrush11 = new SolidBrush(Color.MediumSpringGreen);
e.Graphics.FillRectangle(blueBrush11, rect11);

//title bar img
g.DrawImage(Image.FromFile(title), 30, 25);

//boarding img
g.DrawImage(Image.FromFile(title), 570, 25);

//ALL IN ONE
Font fBody = new Font("Lucida Console", 15, FontStyle.Bold);
Font fBody1 = new Font("Lucida Console", 15, FontStyle.Regular);
Font rs = new Font("Stencil", 25, FontStyle.Bold);
Font fTType = new Font("", 150, FontStyle.Bold);
SolidBrush sb = new SolidBrush(Color.Black);

//title
Font rs1 = new Font("Stencil", 20, FontStyle.Bold);
SolidBrush sb1 = new SolidBrush(Color.White);
g.DrawString("AIR TICKET", rs, sb1, 100, SPACE - (113));
g.DrawString("BOARDING", rs1, sb1, 635, SPACE - (120));
g.DrawString("PASS", rs1, sb1, 670, SPACE - (98));


//Passenger
Font rs2 = new Font("Franklin Gothic", 15);
SolidBrush sb2 = new SolidBrush(Color.Black);
g.DrawString("Name of Passenger ", rs2, sb2, 125, SPACE - (50));
g.DrawString(CustomerName.Text, fBody1, sb2, 125, SPACE - (20));
```



```
//Passenger B
Font rs21 = new Font("Franklin Gothic", 10);
SolidBrush sb21 = new SolidBrush(Color.Black);
g.DrawString("NAME ", rs21, sb21, 556, SPACE - (40));
g.DrawString(CustomerName.Text, fBody1, sb21, 646, SPACE - (40));

//Flight
Font rs3 = new Font("Franklin Gothic", 15);
SolidBrush sb3 = new SolidBrush(Color.Black);
g.DrawString("Flight ", rs3, sb3, 125, SPACE + (10));
g.DrawString(fname.Text, fBody1, sb3, 125, SPACE + 40);

//SEAT B
Font rs22 = new Font("Franklin Gothic", 10);
SolidBrush sb22 = new SolidBrush(Color.Black);
g.DrawString("SEAT NO ", rs22, sb22, 556, SPACE + (10));
g.DrawString(SeatNo.Text, fBody1, sb22, 646, SPACE + 10);

//From
Font rs4 = new Font("Franklin Gothic", 15);
SolidBrush sb4 = new SolidBrush(Color.Black);
g.DrawString("From ", rs4, sb4, 125, SPACE + (70));
g.DrawString(arrival.Text, fBody1, sb4, 125, SPACE + 100);

//To B
Font rs23 = new Font("Franklin Gothic", 10);
SolidBrush sb23 = new SolidBrush(Color.Black);
g.DrawString("TO ", rs23, sb23, 556, SPACE + (60));
g.DrawString(departure.Text, fBody1, sb23, 646, SPACE + 60);

//To
Font rs5 = new Font("Franklin Gothic", 15);
SolidBrush sb5 = new SolidBrush(Color.Black);
g.DrawString("To ", rs5, sb5, 125, SPACE + (130));
g.DrawString(departure.Text, fBody1, sb5, 125, SPACE + 160);

//Seat No
Font rs6 = new Font("Franklin Gothic", 15);
SolidBrush sb6 = new SolidBrush(Color.Black);
g.DrawString("Seat No ", rs6, sb6, 366, SPACE - (50));
g.DrawString(SeatNo.Text, fBody1, sb6, 366, SPACE - (20));

//From B
Font rs24 = new Font("Franklin Gothic", 10);
SolidBrush sb24 = new SolidBrush(Color.Black);
g.DrawString("FROM ", rs24, sb24, 556, SPACE + (110));
g.DrawString(arrival.Text, fBody1, sb24, 646, SPACE + 110);

//FLIGHT No
Font rs7 = new Font("Franklin Gothic", 15);
SolidBrush sb7 = new SolidBrush(Color.Black);
g.DrawString("Flight No ", rs7, sb7, 366, SPACE + (10));
g.DrawString(FlightId.Text, fBody1, sb7, 366, SPACE + 40);
```

Figure 36 Coding for Flight Booking

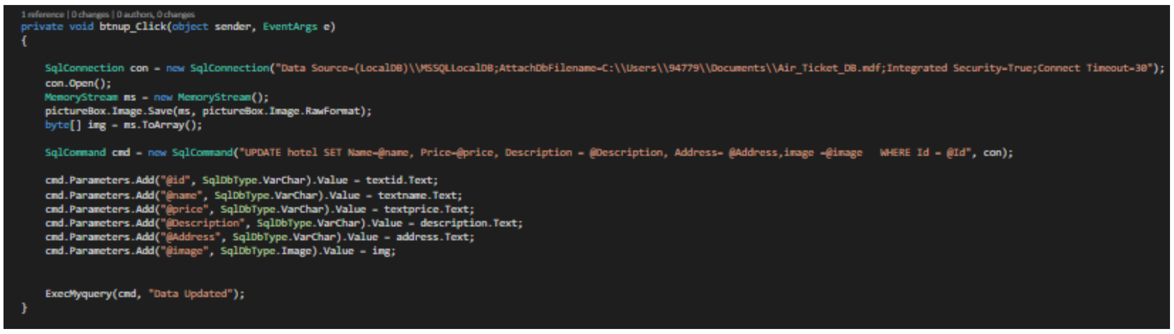
5.2.7. Coding for Insert Records



```csharp
private void btnins_Click(object sender, EventArgs e)
{

    SqlConnection con = new SqlConnection("Data Source=(LocalDB)\\MSSQLLocalDB;AttachDbFilename=C:\\Users\\94779\\Documents\\Air_Ticket_DB.mdf;Integrated Security=True;Connect Timeout=30");
    con.Open();
    MemoryStream ms = new MemoryStream();
    pictureBox.Image.Save(ms, pictureBox.Image.RawFormat);
    byte[] img = ms.ToArray();

    SqlCommand cmd = new SqlCommand("INSERT INTO hotel(Id, Name,Price,Description,Address,image) VALUES (@id,@name,@price,@Description,@Address,@image)", con);

    cmd.Parameters.Add("@id", SqlDbType.VarChar).Value = textid.Text;
    cmd.Parameters.Add("@name", SqlDbType.VarChar).Value = textname.Text;
    cmd.Parameters.Add("@price", SqlDbType.VarChar).Value = textprice.Text;
    cmd.Parameters.Add("@Description", SqlDbType.VarChar).Value = description.Text;
    cmd.Parameters.Add("@Address", SqlDbType.VarChar).Value = address.Text;
    cmd.Parameters.Add("@image", SqlDbType.Image).Value = img;

    ExecMyquery(cmd, "Data Inserted");
    clear();
}
```

Figure 37 Coding for Insert Records

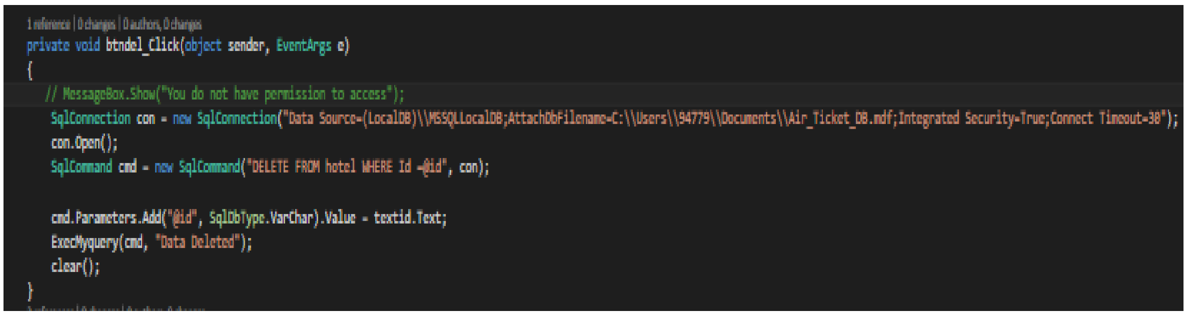
5.2.8. Coding for Update Records



```csharp
1 reference | 0 changes | 0 authors, 0 changes
private void btnup_Click(object sender, EventArgs e)
{

    SqlConnection con = new SqlConnection("Data Source=(LocalDB)\\MSSQLLocalDB;AttachDbFilename=C:\\Users\\94779\\Documents\\Air_Ticket_DB.mdf;Integrated Security=True;Connect Timeout=30");
    con.Open();
    MemoryStream ms = new MemoryStream();
    pictureBox.Image.Save(ms, pictureBox.Image.RawFormat);
    byte[] img = ms.ToArray();

    SqlCommand cmd = new SqlCommand("UPDATE hotel SET Name=@name, Price=@price, Description = @Description, Address= @Address,image =@image   WHERE Id = @Id", con);

    cmd.Parameters.Add("@id", SqlDbType.VarChar).Value = textid.Text;
    cmd.Parameters.Add("@name", SqlDbType.VarChar).Value = textname.Text;
    cmd.Parameters.Add("@price", SqlDbType.VarChar).Value = textprice.Text;
    cmd.Parameters.Add("@Description", SqlDbType.VarChar).Value = description.Text;
    cmd.Parameters.Add("@Address", SqlDbType.VarChar).Value = address.Text;
    cmd.Parameters.Add("@image", SqlDbType.Image).Value = img;


    ExecMyquery(cmd, "Data Updated");
}
```

Figure 38 Coding for Update Records

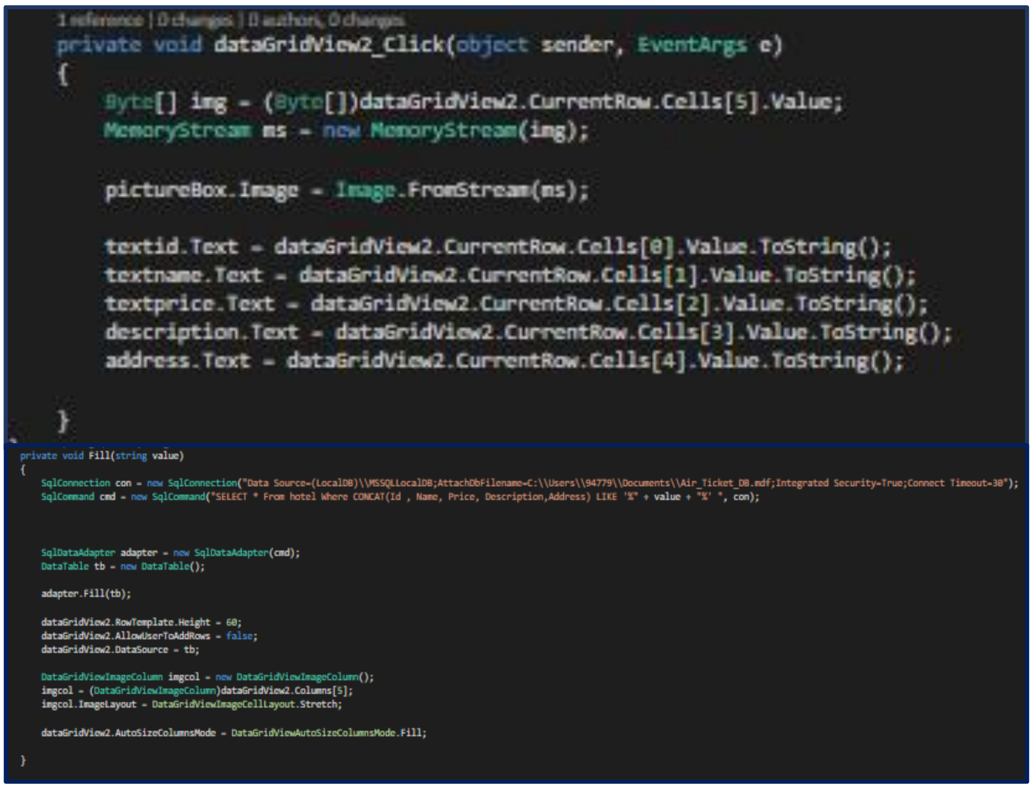
5.2.9. Coding for Delete Records



```csharp
1 reference | 0 changes | 0 authors, 0 changes
private void btndel_Click(object sender, EventArgs e)
{
   // MessageBox.Show("You do not have permission to access");
    SqlConnection con = new SqlConnection("Data Source=(LocalDB)\\MSSQLLocalDB;AttachDbFilename=C:\\Users\\94779\\Documents\\Air_Ticket_DB.mdf;Integrated Security=True;Connect Timeout=30");
    con.Open();
    SqlCommand cmd = new SqlCommand("DELETE FROM hotel WHERE Id =@id", con);

    cmd.Parameters.Add("@id", SqlDbType.VarChar).Value = textid.Text;
    ExecMyquery(cmd, "Data Deleted");
    clear();
}
```

Figure 39 Coding for Delete Records

5.2.10. Coding for Data grid view (Cell Click and view)



```
1 reference | 0 changes | 0 authors, 0 changes
private void dataGridView2_Click(object sender, EventArgs e)
{
    Byte[] img = (Byte[])dataGridView2.CurrentRow.Cells[5].Value;
    MemoryStream ms = new MemoryStream(img);

    pictureBox.Image = Image.FromStream(ms);

    textid.Text = dataGridView2.CurrentRow.Cells[0].Value.ToString();
    textname.Text = dataGridView2.CurrentRow.Cells[1].Value.ToString();
    textprice.Text = dataGridView2.CurrentRow.Cells[2].Value.ToString();
    description.Text = dataGridView2.CurrentRow.Cells[3].Value.ToString();
    address.Text = dataGridView2.CurrentRow.Cells[4].Value.ToString();

}
```

```
private void Fill(string value)
{
    SqlConnection con = new SqlConnection("Data Source=(LocalDB)\\MSSQLLocalDB;AttachDbFilename=C:\\Users\\94779\\Documents\\Air_Ticket_DB.mdf;Integrated Security=True;Connect Timeout=30");
    SqlCommand cmd = new SqlCommand("SELECT * From hotel Where CONCAT(Id , Name, Price, Description,Address) LIKE '%" + value + "%' ", con);



    SqlDataAdapter adapter = new SqlDataAdapter(cmd);
    DataTable tb = new DataTable();

    adapter.Fill(tb);

    dataGridView2.RowTemplate.Height = 60;
    dataGridView2.AllowUserToAddRows = false;
    dataGridView2.DataSource = tb;

    DataGridViewImageColumn imgcol = new DataGridViewImageColumn();
    imgcol = (DataGridViewImageColumn)dataGridView2.Columns[5];
    imgcol.ImageLayout = DataGridViewImageCellLayout.Stretch;

    dataGridView2.AutoSizeColumnsMode = DataGridViewAutoSizeColumnsMode.Fill;

}
```

Figure 40 Coding for Data gird view

5.3 Database Tables

5.3.1. Accounts

Field Name	Data Type
Username	nchar(10)
Password	nchar(10)

Table 1 Accounts

5.3.2. Trans

Field Name	Data Type
Id	int (Auto Increment)
Name	nvarchar(50)
Price	nvarchar(50)
Description	nvarchar(MAX)
EngineCapacity	nvarchar(50)
FuelType	nvarchar(50)
Transmission	nvarchar(50)
Image	image

Table 2 Trans

5.3.3. Tra_booked

Field Name	Data Type
Id	int (Auto Increment)
Method	nvarchar(50)
Place	nvarchar(50)
Passenger	nvarchar(50)

Table 3 Tra_booked

5.3.4. Manage Users

Field Name	Data Type
Id	int (Auto Increment)
First_Name	text
Phone	text
Cnic	text
Username	text
Password	ntext
Role	text

Table 4 Manage Users

5.3.5. Salary

Field Name	Data Type
Id	int (Auto Increment)
Month	ntext
Date	nchar(10)

Table 5 Salary

5.3.6. Place

Field Name	Data Type
Id	int (Auto Increment)
Location	varchar(MAX)
Dis	varchar(50)

Table 6 Place

5.3.7. Payment

Field Name	Data Type
Id	int (Auto Increment)
CNIC	varchar(50)
Name	varchar(50)
Bsalary	varchar(50)
EPF	varchar(50)
ETF	varchar(50)
Nsalary	varchar(50)
Issue_Date	varchar(50)
Status	varchar(50)
Email	varchar(50)

Table 7 Payment

5.3.8. Issue

Field Name	Data Type
Id	int (Auto Increment)
Cnic	varchar(50)
Doc	image

Table 8 Issue

5.3.9 Hotel_booked

Field Name	Data Type
Id	int (Auto Increment)
Name	varchar(50)
Room	varchar(50)
Days	varchar(50)

Table 9 Hotel Booked

5.3.10 Hotel

Field Name	Data Type
Id	int (Auto Increment)
Name	varchar(MAX)
Price	varchar(MAX)
Description	varchar(MAX)
Address	varchar(MAX)
Image	image

Table 10 Hotel

5.3.11 Food

Field Name	Data Type
Id	int (Auto Increment)
Name	varchar(MAX)
Price	varchar(MAX)
Image	image

Table 11 Food

5.3.12 Flight

Field Name	Data Type
FlightId	int (Auto Increment)
FlightName	text
DepaurtureTime	time
Destination	text
FlightClass	nchar(10)
FlightChargers	int
ArrivalTime	time(7)
Date	date
Seats	int

Table 12 Flight

5.3.13 Emp

Field Name	Data Type
Id	int (Auto Increment)
Name	varchar(50)
Address	varchar(50)
Phone	varchar(50)
Email	varchar(50)
Qualification	varchar(50)
Bsalary	varchar(50)
Role	varchar(50)
CNIC	varchar(50)

Table 13 Emp

5.3.14 Customer

Field Name	Data Type
Id	int (Auto Increment)
Name	nvarchar(50)
Address	nvarchar(50)
PhoneNo	nvarchar(50)
Email	nvarchar(50)
Country	nchar(10)
FatherN	nvarchar(50)

Table 14 Customer

5.3.15 Code

Field Name	Data Type
Id	int (Auto Increment)
Code	int

Table 15 Code

5.3.16 Booking

Field Name	Data Type
TicketId	int (Auto Increment)
CustermerId	int
Dateofjourney	date
FlightId	int
SeatNo	int
Country	int
FatherN	int

Table 16 Booking

5.3.17 Booked

Field Name	Data Type
Id	int (Auto Increment)
Name	varchar(50)
Required	varchar(50)
Type	varchar(50)

Table 17 Booked

5.3.18 AddTicket

Field Name	Data Type
Id	int (Auto Increment)
TicketName	nchar(10)
TicketNumber	nchar(10)
Tickets	nchar(10)

Table 18 Add Ticket

5.3.19 AddFlight

Field Name	Data Type
Id	int (Auto Increment)
FlightName	nvarchar(50)
Source	nvarchar(50)
Destination	nvarchar(50)
Departure	time(7)
Arrival	time(7)
FlightClass	nvarchar(50)
FlightCharges	nvarchar(50)
Seats	int

Table 19 Add Flight

CHAPTER 06

EXPERIMENTS & RESULTS

6.EXPERIMENTS & RESULTS

6.1. Testing Admin Login

Test	Check in Success	Check in Failed
Giving correct password	"Login Successful !"	-
Giving incorrect password	-	-

Table 20 Testing Admin Login

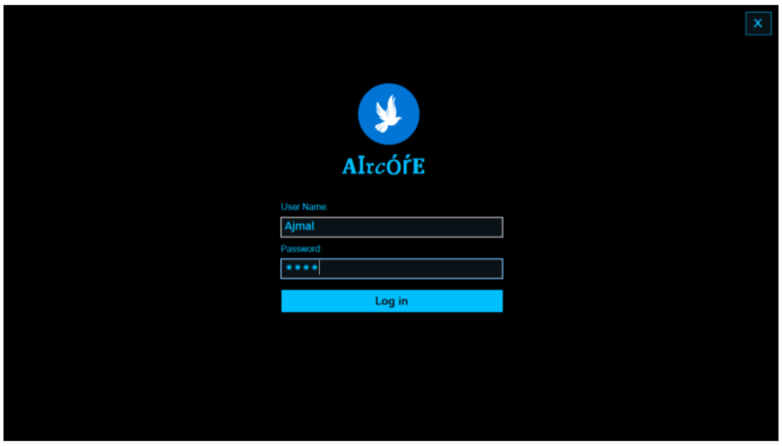


Figure 41 Testing Admin Login 1- HMS

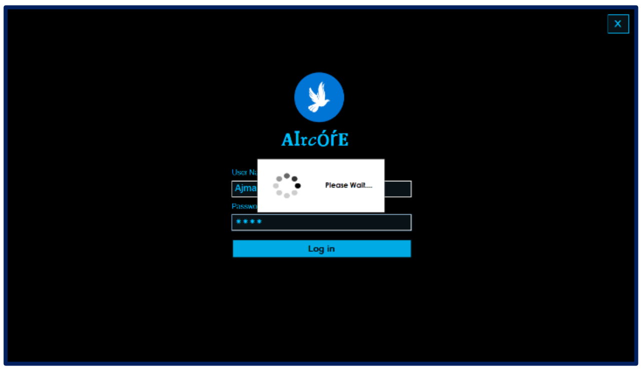
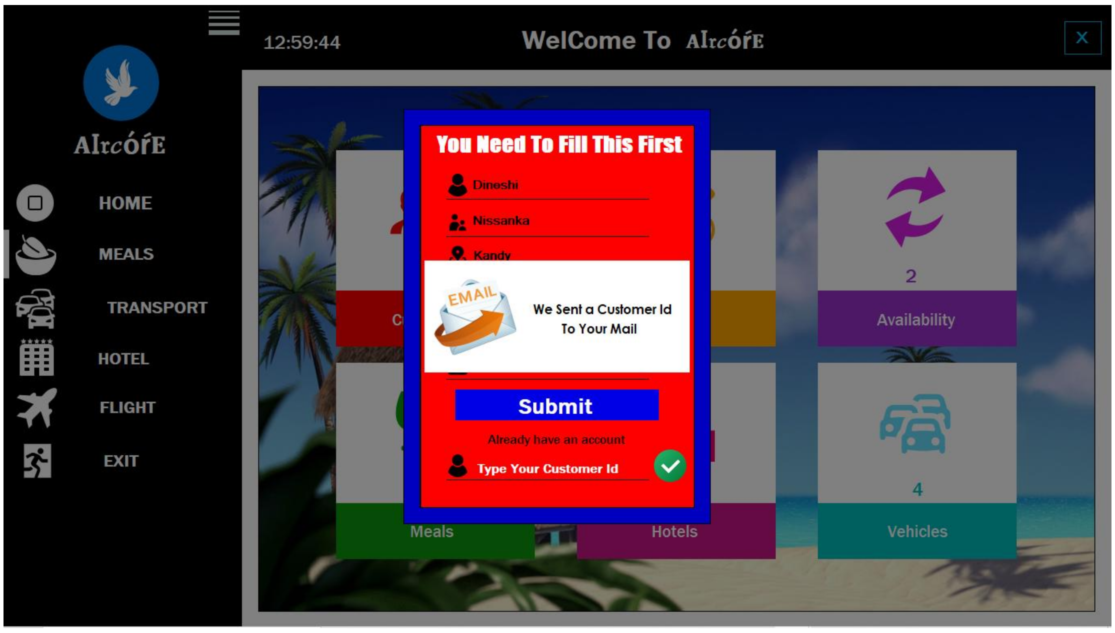


Figure 42 Testing Admin Login 2

6.2. Testing Customer Registration

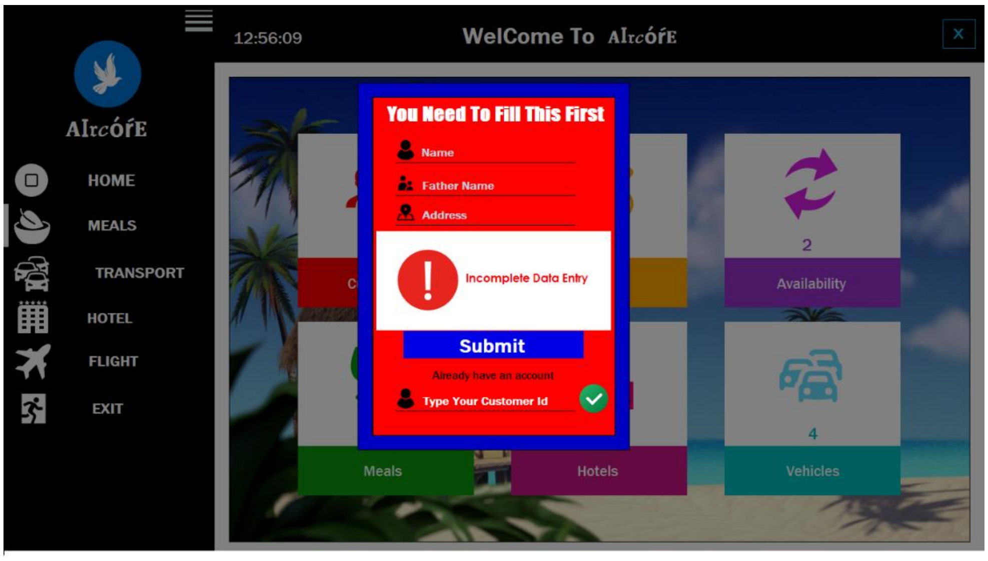
Test	Check in Success	Check in Failed
Giving correct data entries	"We sent a customer id to your mail"	-
Giving incorrect data entries	-	"Incomplete Data Entry"

Table 21 Testing Customer Registration



Dinoshi
Nissanka
Kandy
We Sent a Customer Id To Your Mail

Figure 43 Testing Customer Register 1



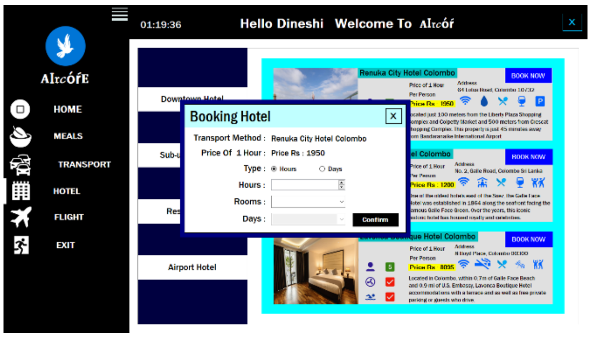
Incomplete Data Entry

Figure 44 Testing Customer Registration

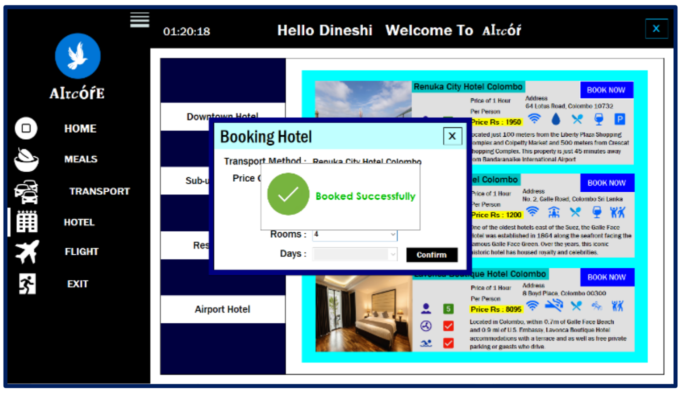
6.3. Testing Hotel Booking

Test	Check in Success	Check in Failed
Input correct records	"Booked Successfully !"	-
Input incorrect records	-	"Incorrect Data Entry "

Table 22 Testing Hotel Booking

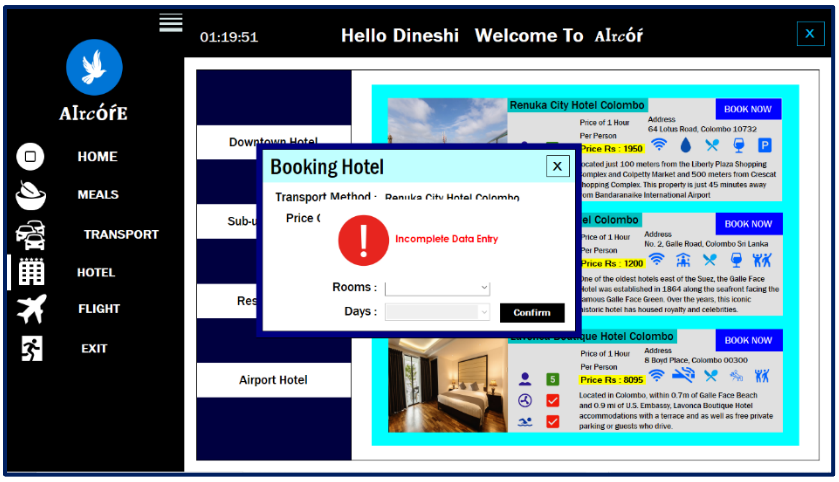


Transport Method : Renuka City Hotel Colombo
Price Of 1 Hour : Price Rs : 1950
Type : Hours Days
Hours :
Rooms :
Days :



Booked Successfully
Rooms : 4
Days :

Figure 45 Testing Hotel Booking 1



Transport Method : Renuka City Hotel Colombo
Incomplete Data Entry
Rooms :
Days :

Renuka City Hotel Colombo
Price of 1 Hour Per Person
Price Rs : 1950
Address 64 Lotus Road, Colombo 10732
Located just 100 meters from the Liberty Plaza Shopping Complex and Colpetty Market and 500 meters from Crescat Shopping Complex. This property is just 45 minutes away from Bandaranaike International Airport

Price Rs : 1200
Address No. 2, Galle Road, Colombo Sri Lanka
One of the oldest hotels east of the Suez, the Galle Face Hotel was established in 1864 along the seafront facing the famous Galle Face Green. Over the years, this iconic historic hotel has housed royalty and celebrities.

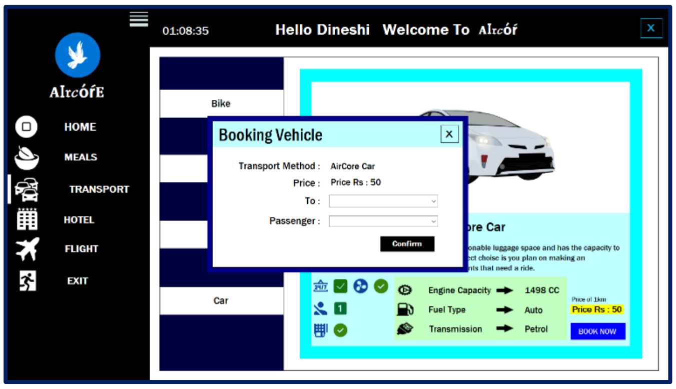
Price Rs : 8095
Address 8 Boyd Place, Colombo 00300
Located in Colombo, within 0.7 m of Galle Face Beach and 0.9 mi of U.S. Embassy, Lavonca Boutique Hotel accommodations with a terrace and as well as free private parking or guests who drive.

Figure 46 Testing Hotel Booking 2

6.4. Testing Vehicle Hiring

Test	Check in Success	Check in Failed
Input appropriate records	"Booked Successfully !"	-
Input inappropriate records	-	"Incorrect Data Entry "

Table 23 Vehicle Hiring


01:08:35 Hello Dineshi Welcome To Alrcóŕ

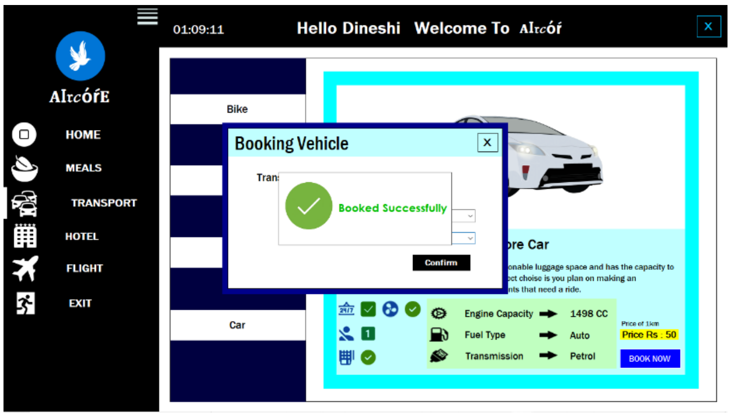
Booking Vehicle

Transport Method : AirCore Car

Price : Price Rs : 50

To :

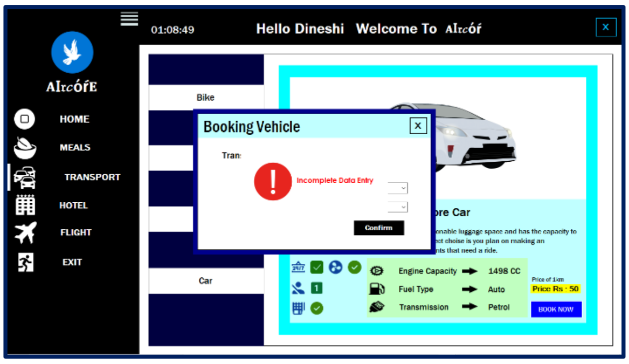
Passenger :


01:09:11 Hello Dineshi Welcome To Alrcóŕ

Booking Vehicle

Booked Successfully

Figure 47 Testing Vehicle Hiring

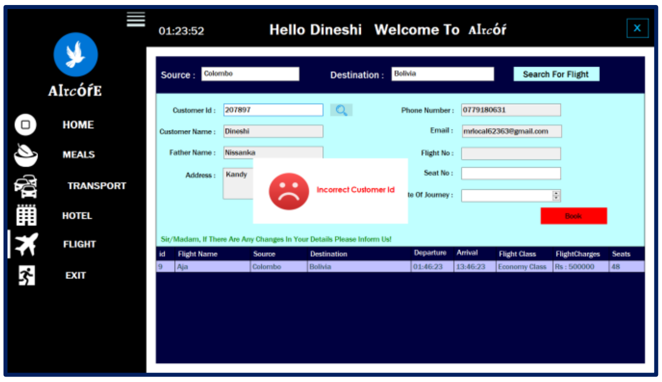

01:08:49 Hello Dineshi Welcome To Alrcóŕ

Booking Vehicle

Incomplete Data Entry

Figure 48 Testing Vehicle Hiring 2

6.5. Testing Flight Options



01:23:52 Hello Dineshi Welcome To AIrcóf

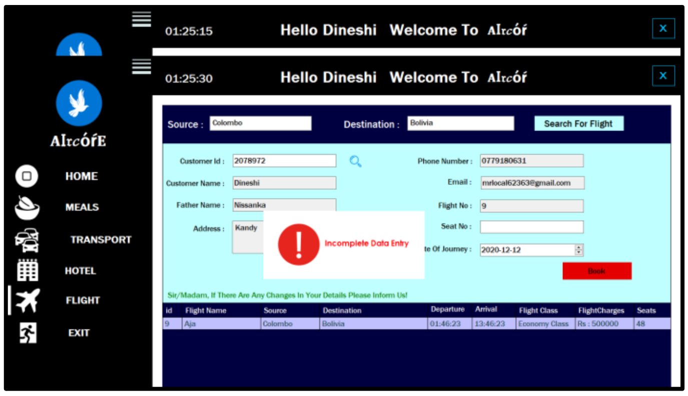
Source : Colombo Destination : Bolivia

Customer Id : 207897
Customer Name : Dineshi
Father Name : Nissanka
Address : Kandy
Phone Number : 0779180631
Email : mrlocal62363@gmail.com
Flight No :
Seat No :
te Of Journey :

Incorrect Customer Id

Sir/Madam, If There Are Any Changes In Your Details Please Inform Us!

id	Flight Name	Source	Destination	Departure	Arrival	Flight Class	FlightCharges	Seats
9	Aja	Colombo	Bolivia	01:46:23	13:46:23	Economy Class	Rs : 500000	48



01:25:15 Hello Dineshi Welcome To AIrcóf

01:25:30 Hello Dineshi Welcome To AIrcóf

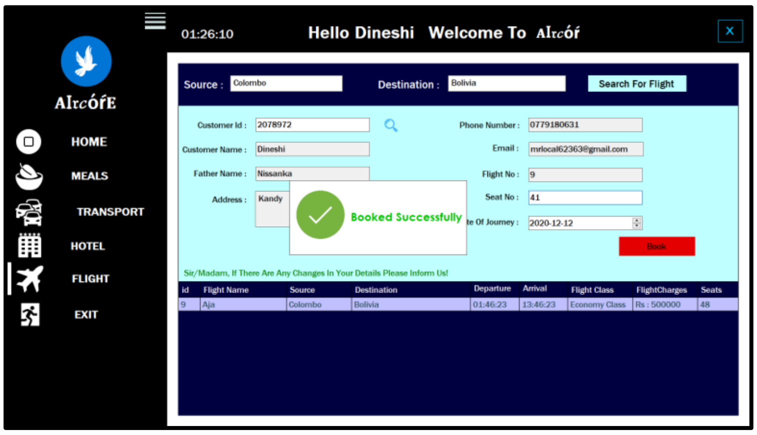
Source : Colombo Destination : Bolivia

Customer Id : 2078972
Customer Name : Dineshi
Father Name : Nissanka
Address : Kandy
Phone Number : 0779180631
Email : mrlocal62363@gmail.com
Flight No : 9
Seat No :
te Of Journey : 2020-12-12

Incomplete Data Entry

Sir/Madam, If There Are Any Changes In Your Details Please Inform Us!

id	Flight Name	Source	Destination	Departure	Arrival	Flight Class	FlightCharges	Seats
9	Aja	Colombo	Bolivia	01:46:23	13:46:23	Economy Class	Rs : 500000	48



01:26:10 Hello Dineshi Welcome To AIrcóf

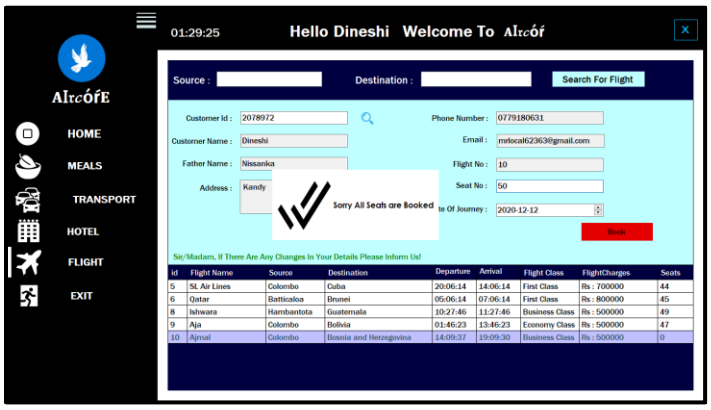
Source : Colombo Destination : Bolivia

Customer Id : 2078972
Customer Name : Dineshi
Father Name : Nissanka
Address : Kandy
Phone Number : 0779180631
Email : mrlocal62363@gmail.com
Flight No : 9
Seat No : 41
te Of Journey : 2020-12-12

Booked Successfully

Sir/Madam, If There Are Any Changes In Your Details Please Inform Us!

id	Flight Name	Source	Destination	Departure	Arrival	Flight Class	FlightCharges	Seats
9	Aja	Colombo	Bolivia	01:46:23	13:46:23	Economy Class	Rs : 500000	48



01:29:25 Hello Dineshi Welcome To AIrcóf

Source : Destination :

Customer Id : 2078972
Customer Name : Dineshi
Father Name : Nissanka
Address : Kandy
Phone Number : 0779180631
Email : mrlocal62363@gmail.com
Flight No : 10
Seat No : 50
te Of Journey : 2020-12-12

Sorry All Seats are Booked

Sir/Madam, If There Are Any Changes In Your Details Please Inform Us!

id	Flight Name	Source	Destination	Departure	Arrival	Flight Class	FlightCharges	Seats
5	SL Air Lines	Colombo	Cuba	20:06:14	14:06:14	First Class	Rs : 700000	44
6	Qatar	Batticaloa	Brunei	05:06:14	07:06:14	First Class	Rs : 800000	45
8	Ishwara	Hambantota	Guatemala	10:27:46	11:27:46	Business Class	Rs : 500000	49
9	Aja	Colombo	Bolivia	01:46:23	13:46:23	Economy Class	Rs : 500000	47
10	Ajmal	Colombo	Bosnia and Herzegovina	14:09:37	19:09:30	Business Class	Rs : 500000	0

Figure 49 Testing Flight Option

7. Conclusion

To conclude the description about the project. The project developed using C#.Net and MySQL is based on the requirement specification of the user and the analysis of the existing system with flexibility and adaptability for the future enhancement. The expanded functionality of the present software requires an appropriate approach towards software development.

Identification of the drawbacks of the existing system leads to the design of automated computerized system that will be compatible to the existing system with the system which is more user-friendly and GUI oriented.

Finally, this Ticket Management System can perform flexibly and efficiently while operate the software and more adaptability with future improvements and enhancements.

8.Future Scope

There are many additional features, which can be planned to be incorporated during the future enhancement of this software. This is a Stand-Alone Ticket Management System; the future version of this software can be a web-based system for ticketing services. At the future version a passenger can surf the flight details through internet and allocate a flight ticket for themselves by register themselves to the ticketing agency. By registering themselves, they will get a username and password for their login. Therefore, passenger can visit to the system whenever and view their flight ticket allocation records.

The proposed software can perform very efficiently, as well as in the future described features can be added to this software.

9.References

G. Rajkumar , T. Sivagama Sundari. (2019). Ticket Management System Based on Finger Print Authentication. *computerscijournal*.

Gutierrez, P. J. (2007, July 12). *https://patents.google.com/.* Retrieved from https://patents.google.com/: https://patents.google.com/

MUYESHI, P. (2009, April 9). *https://www.academia.edu/31425862/TICKET_MANAGEMENT_SYSTEM_Submitted_by*. Retrieved from https://www.academia.edu/: https://www.academia.edu/

panelNorazahMohd , SukiaNorbayahMohd Sukib. (2017). Flight ticket booking app on mobile devices: Examining the determinants of individual intention to use. *Elsevier*.

Patrick J. TooleSudhir KrishnaswamyNima MoayediDavid N. LordFrancisco J. Gutierrez. (2007). *https://patents.google.com/.* Retrieved from https://patents.google.com/

RESHMI RADHAKRISHNAN, RINSHA P.A, ROOPASREE R. (2014, June 27). *ONLINE Ticket HOSTEL MANAGEMENT SYSTEM*. Retrieved from http://dspace.cusat.ac.in/: http://hdl.handle.net/123456789/8250

Sujana, A. (2013). *r Trincomalee Campus*. Retrieved from http://hdl.handle.net: http://hdl.handle.net/123456789/696

Yatera, J. (2014). *Ticket Processes management draft report.* Computer Sciencee.