Software Code and Implementation Document

PROJECT: AIRLINE RESERVATION SYSTEM

BY: UBAIDA WAHEED
SANA AKBAR
TAYYBA SALAMAT

Table of Contents

1. Scope

The AIRLINE TICKET MANAGEMENT SYSTEM offers a lot of opportunities for the customers, who have a great deal of trouble to find their place in the advanced the required plane. This type of system and application software, provides passengers with the opportunity to view the flights and time and destination allotted to them. It will also provide an opportunity to save a ticket, or to change or cancel it, these are the specific terms and conditions that this system offers.

2. Purpose

This document contains Requirement Specifications for AIRLINE TICKET MANAGEMENT SYSTEM. It shows the different aspects of the requirement specifications as the document proceeds. ART management system eliminates people problems that they face while reserving ticket, and helps them to save themselves a seat on plane.

3. Implementation

The implementation of AIRLINE RESERVATION SYSTEM is done using the C#, Microsoft SQL Server Management Studio and Microsoft Visual Studio along with the use of ASP.NET.

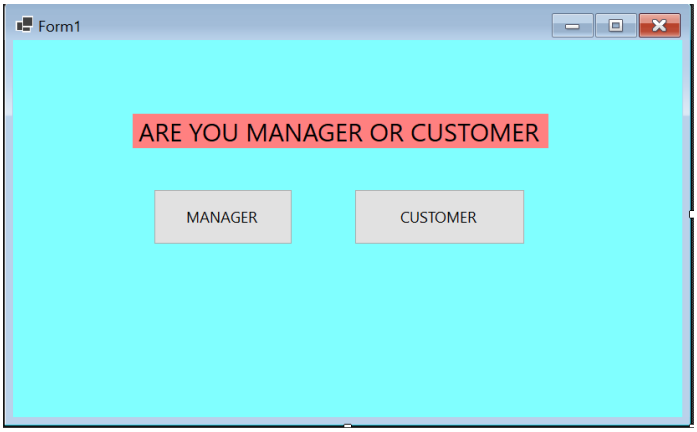
The GUI of the forms is made on the Visual Studio 2022 using the C# language.

For database storage, management and connection Microsoft SQL Server Management Studio is used.

4. Source Code

The source code of the forms for defining the GUI of the AIRLINE MANAGEMENT SYSTEM are as follows:

Form 1:

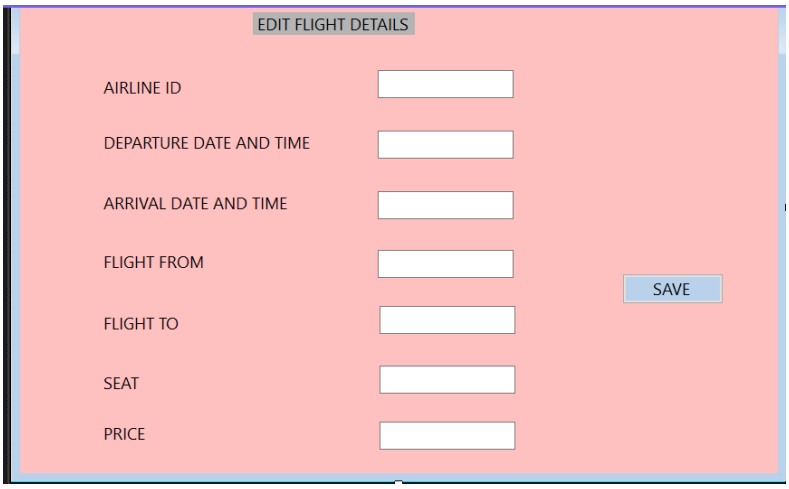


Source Code:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
namespace WinFormsApp1
{
    public partial class Form1 : Form
```

```
    {
        public Form1()
        {
            InitializeComponent();
        }
        private void button1_Click(object sender, EventArgs e)
        {
            Form2 f = new Form2();
            f.Show();
            this.Hide();
        }

        private void Form1_Load(object sender, EventArgs e)
        {

        }

        private void button21_Click(object sender, EventArgs e)
        {
            Form4 ff = new Form4();
            ff.Show();
            this.Hide();
        }
        private void button31_Click(object sender, EventArgs e)
        {
            Form3 ff = new Form3();
            ff.Show();
            this.Hide();
        }
    }
}
```

Form 2:

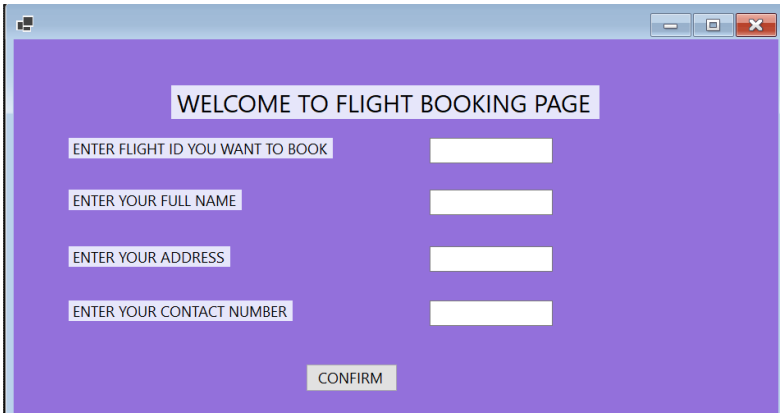


Source Code:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
```

```
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.OleDb;
using System.Data.SqlClient;

namespace WinFormsApp1
{
    public partial class Form2 : Form
    {
        public Form2()
        {
            InitializeComponent();
        }

        public string conString =@"Data Source = DESKTOP-TAEPE9T\SQLEXPRESS; Initial
Catalog=FMS; Integrated Security = True";
        private void button1_Click(object sender, EventArgs e)
        {
            SqlConnection con = new SqlConnection(conString);
            con.Open();
            if (con.State == System.Data.ConnectionState.Open)
            {
                string q = "insert into FLIGHT_DETAILS(AIR_ID, DEPARTURE_DATETIME,
ARRIVAL_DATETIME, FLIGHT_FROM, FLIGHT_TO, SEATS, PRICES) values('" + airid.Text.ToString()
+ "','" + ddt.Text.ToString() + "','"+ adt.Text.ToString()+"','"+ff.Text.ToString() +
"','"+ft.Text.ToString()+"','"+seats.Text.ToString()+"','"+price.Text.ToString()+"')";
                SqlCommand cmd = new SqlCommand(q, con);
                cmd.ExecuteNonQuery();
                MessageBox.Show("FLIGHT DETAILS SAVED");
            }
        }
        private void label8_Click(object sender, EventArgs e)
        {
        }

        private void Form2_Load(object sender, EventArgs e)
        {

        }
    }
}
```

Form 3:

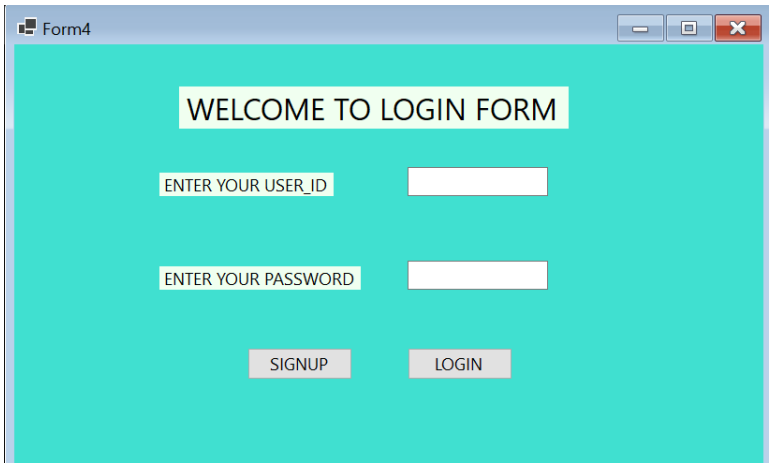


Source Code:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using System.Data.OleDb;
using System.Data.SqlClient;
namespace WinFormsApp1
{
    public partial class Form3 : Form
    {
        public Form3()
        {
            InitializeComponent();
        }

        public string conString = @"Data Source = DESKTOP-TAEPE9T\SQLEXPRESS; Initial
Catalog=FMS; Integrated Security = True";
        private void button1_Click(object sender, EventArgs e)
        {
            SqlConnection con = new SqlConnection(conString);
            con.Open();
            if (con.State == System.Data.ConnectionState.Open)
            {
                string q = "insert into
BOOKED_FLIGHTS(FLIGHT_ID,NAME,ADDRESS,CONTACT_NUMBER)  values('" + fid.Text.ToString() +
"','" + fn.Text.ToString() + "','" + add.Text.ToString() + "','" + cn.Text.ToString() +
"')";
                SqlCommand cmd = new SqlCommand(q, con);
                cmd.ExecuteNonQuery();
                MessageBox.Show("TICKET BOOKED");
            }
        }
        private void label3_Click(object sender, EventArgs e)
        {

        }

        private void Form3_Load(object sender, EventArgs e)
```

```
        {

        }
    }
}
```

Form 4:

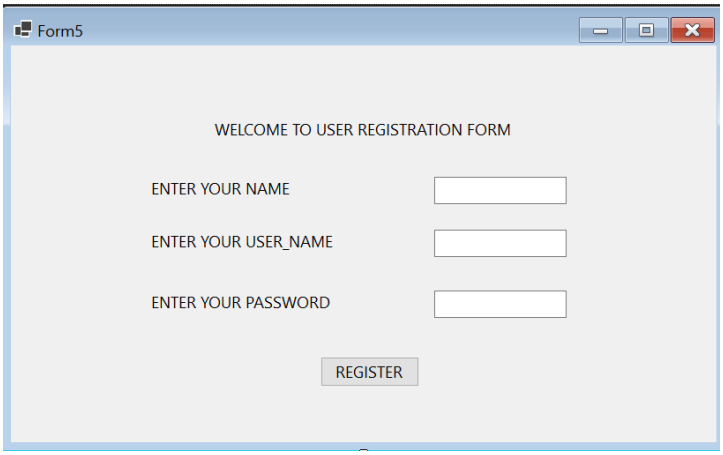


Source Code:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.Data.Sql;
using System.Data.OleDb;
using System.Data.SqlClient;

namespace WinFormsApp1
{
    public partial class Form4 : Form
    {
        public Form4()
    {
        InitializeComponent();
    }
        private void button1_Click(object sender, EventArgs e)
        {

        }

        private void button2_Click(object sender, EventArgs e)
        {
            Application.Exit();

        }
```

```
        private void button1_Click_1(object sender, EventArgs e)
        {
         SqlConnection conString =new SqlConnection(@"Data Source = DESKTOP-
TAEPE9T\SQLEXPRESS; Initial Catalog=FMS; Integrated Security = True");
        string qurey = "SELECT * FROM LOGIN WHERE user_ID='" + uid.Text + "' AND PASS='" +
pas.Text + "'";
            SqlDataAdapter sda = new SqlDataAdapter(qurey, conString);           /* in
above line the program is selecting the whole data from table and the matching it with the
user name and password provided by user. */
            DataTable dt1 = new DataTable(); //this is creating a virtual table
            sda.Fill(dt1);
            DataRowCollection rows = dt1.Rows;
            if (rows.Count == 1)
            {
                MessageBox.Show("LOGIN SUCCESSFUL WELCOME TO BOOKING PAGE");
                /* I have made a new page called home page. If the user is successfully
authenticated then the form will be moved to the next form */
                Form3 ff = new Form3();
                ff.Show();
                this.Hide();
            }
            else
                MessageBox.Show("Invalid username or password");

        }
        private void button2_Click_2(object sender, EventArgs e)
        {
            SqlConnection conString = new SqlConnection(@"Data Source = DESKTOP-
TAEPE9T\SQLEXPRESS; Initial Catalog=FMS; Integrated Security = True");
            conString.Open();
            if (conString.State == System.Data.ConnectionState.Open)
            {
                Form fff = new Form5();
                fff.Show();
                this.Hide();
            }

        }
            private void Form4_Load(object sender, EventArgs e)
        {

        }
    }
}
```

Form 5:



Source Code:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.OleDb;
using System.Data.SqlClient;
namespace WinFormsApp1
{
    public partial class Form5 : Form
    {
        public Form5()
        {
            InitializeComponent();
        }
        public string conString = @"Data Source = DESKTOP-TAEPE9T\SQLEXPRESS; Initial
Catalog=FMS; Integrated Security = True";
        private void button1_Click(object sender, EventArgs e)
        {
            SqlConnection con = new SqlConnection(conString);
            con.Open();
            if (con.State == System.Data.ConnectionState.Open)
            {
                string qurey = "SELECT * FROM REGISTRATION WHERE NAME='" + fna.Text + "'
AND USERNAME='" + ui.Text + "' AND PASSWORD ='" +pw.Text + "'";
                SqlDataAdapter sda = new SqlDataAdapter(qurey, conString);
                MessageBox.Show("Congratulations You Have Registered Successfully");
            }

        }

        private void Form5_Load(object sender, EventArgs e)
        {

        }
    }
```

}

5. Stages:

The overall code review process is divided into three stages.

Stage	Name	Purpose
1	Code Preparation	We as the preparer ensures that code adheres to code review checklist. We make certain non-functional changes as necessary. We make sure to notice that any functional changes that should be made to adhere to checklist.
2	Off-line Code Review	We individually review the software design document in detail so that any of the point coving the check list of SRS doesn't get ignored.
3	Formal Code Review Meeting	After that we as the reviewers suggested changes from the off-line code reviews, decide actions, approve code.

6. Definitions:

The following definitions apply.

Code Review	The formal review of software units or modules.
A Unit	One function or routine, starting from its comment header block, to the last line of code or comment in the unit
A Module	The logical grouping of a set of units and its data structures. Normally a module will consist of one or more '.C' source files and it's associated '.H' files.
The Presenter	The person who authored the module or unit(s) for code review.
The Reviewers	The two or more people who are reviewing the module or unit(s)
SRS	Software Requirements Specification – document containing the user/system level requirements this piece of software must fulfill
SDS	Software Design Specification - document containing specific information about the high-level design of the Unit or Module being reviewed.