**Department of Computer Science**



**Visual Programming**

**Semester Project Documentation**

**Group Members:**

|  |  |
| --- | --- |
| Vaneeza Aftab | 225159 |
| Syed Sail Abbas | 225183 |
| Ahmed Iqbal | 225157 |
| Hussain Shahzaib | 225168 |

**Submitted To:** Mr. Mustajab Hussain

**Date:** 28th may, 2024

**Flight Reservation System Documentation**

**Table of Contents**

1. Introduction
2. System Overview
3. User Interface Description
   1. Splash Screen
   2. Login Page
   3. Main Menu
   4. Flights
   5. Passengers
   6. Tickets
   7. Cancellations
4. Database Design
5. Features and Functionality
6. Implementation Details
   1. Development Environment
   2. Tools and Technologies Used
   3. Code Snippets
7. Testing
8. Conclusion
9. Future Enhancements
10. Appendix

**1. Introduction**

The Flight Reservation System is a Windows Forms application developed for the Visual Programming course. The system is designed for "3 Friends Airline" and facilitates the management of flight schedules, passenger information, ticket booking, and cancellations.

**2. System Overview**

The application provides a user-friendly interface for airline staff to manage various aspects of flight reservations. It includes functionalities to add, update, and delete records related to flights, passengers, tickets, and cancellations.

**3. User Interface Description**

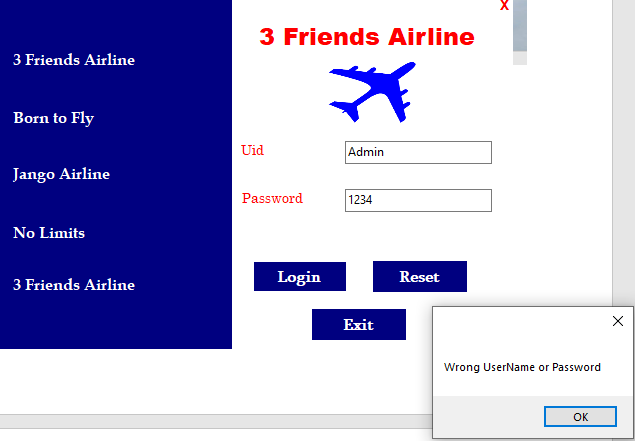
**Splash Screen**

Upon launching the application, a splash screen for "3 Friends Airline" is displayed. This screen serves as an introduction to the application and provides a brief delay before the main interface is loaded.



**Login Page**

The login page requires users to enter their credentials to access the system. Upon successful login, the user is navigated to the main menu.





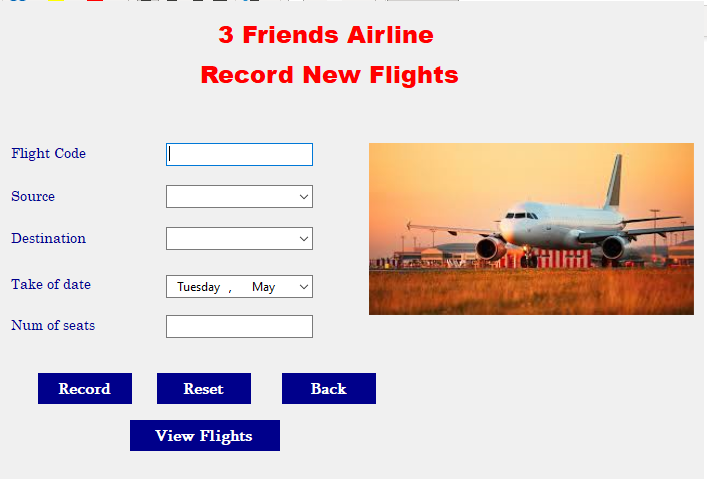
**Main Menu**

The main menu provides four options: Flights, Passengers, Tickets, and Cancellations. Each option opens a different form for managing the respective data.



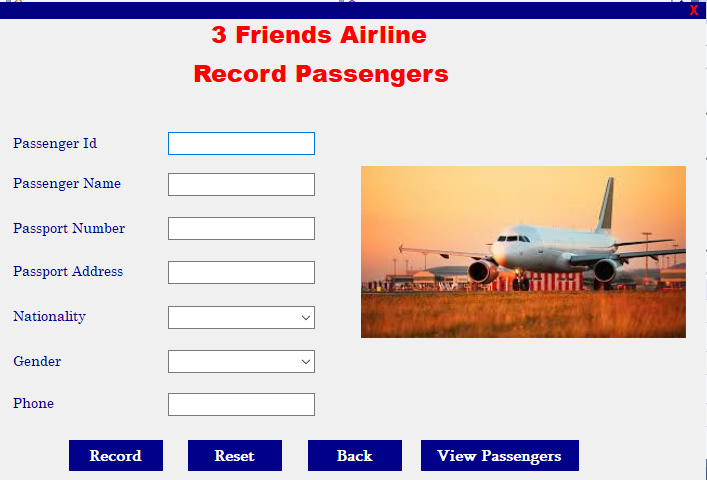
**Flights**

The Flights form allows users to enter and manage flight information. It includes textboxes for entering flight details such as flight number, destination, departure time, etc. Users can add new flights, update existing flight information, or delete flights.



**Passengers**

The Passengers form is used to manage passenger information. Users can enter details such as passenger name, contact information, and other relevant data. The form allows for adding, updating, and deleting passenger records.



**Tickets**

The Tickets form facilitates ticket booking. Users can enter ticket information, including the flight number, passenger details, seat number, and other relevant data. The form supports adding new tickets, updating ticket information, and canceling tickets.

**Cancellations**

The Cancellations form is used to manage flight cancellations. Users can enter details of the cancellations, including the flight number and reason for cancellation. The form allows users to add new cancellations and update or delete existing cancellation records.

**4. Database Design**

Provide details about the database schema, including tables, columns, data types, and relationships between tables. Include a diagram if possible.

**5. Features and Functionality**

- User authentication and login

- Manage flight information

- Manage passenger information

- Book and manage tickets

- Manage cancellations

**6. Implementation Details**

**Development Environment**

- Visual Studio 2022

- .NET Framework

**Tools and Technologies Used**

- Windows Forms

- C#

- SQL Server

**Code Snippets**

Following are key code snippets, such as the login authentication process, database connection setup, and CRUD operations for each form.

**Login**

|  |
| --- |
| private void button1\_Click(object sender, EventArgs e)  {  if (UidTb.Text == "" || PassTb.Text == "")  {  MessageBox.Show("Enter the User Id and Password");  }  else if (UidTb.Text == "Admin" && PassTb.Text == "Admin")  {  Home home = new Home();  home.Show();  this.Hide();  }  else  {  MessageBox.Show("Wrong UserName or Password");  }  } |

**Flights**

|  |
| --- |
| private void button1\_Click(object sender, EventArgs e)  {  if (FcodeTb.Text == "" || Fsrc.Text == "" || FDest.Text == "" || FDate.Text == "" || SeatNum.Text == "")  {  MessageBox.Show("Missing Info");  }  else  {  try  {  Con.Open();  string query = "insert into FlightTb1 values('" + FcodeTb.Text + "','" + Fsrc.SelectedItem.ToString() + "','" + FDest.SelectedItem.ToString() + "','" + FDate.Value.ToString() + "'," + SeatNum.Text + " )";  SqlCommand cmd = new SqlCommand(query, Con);  cmd.ExecuteNonQuery();  MessageBox.Show("Flight Recorded Successfully");  Con.Close();  }  catch (Exception ex)  {  MessageBox.Show(ex.Message);  }  }  } |

**Passengers**

**View Passengers**

|  |
| --- |
| private void button1\_Click(object sender, EventArgs e)  {  if (Passid.Text == "" || PassAd.Text == "" || PassName.Text == "" || PassportTb.Text == "" || PhoneTb.Text == "")  {  MessageBox.Show("Missing Info");  }  else  {  try  {  Con.Open();  string query = "insert into PassengerTb1 values(" + Passid.Text + ",'" + PassName.Text + "','" + PassportTb.Text + "','" + PassAd.Text + "','" + NationalityCb.SelectedItem.ToString() + "','" + GenderCb.SelectedItem.ToString() + "','" + PhoneTb.Text + "')";  SqlCommand cmd = new SqlCommand(query, Con);  cmd.ExecuteNonQuery();  MessageBox.Show("Passenger Recorded Successfully");  Con.Close();  }  catch (Exception ex)  {  MessageBox.Show(ex.Message);  }  }  } |

**Update Passenger**

|  |
| --- |
| private void button1\_Click(object sender, EventArgs e)  {  if(PidTb.Text=="" || PnameTb.Text=="" || PpassTb.Text=="" || PaddTb.Text=="" )  {  MessageBox.Show("Missing Information");  }  else  {  try  {  Con.Open();  string query = "update PassengerTb1 set PassName='" + PnameTb.Text + "',Passport='" + PpassTb.Text + "',PassAd='" + PaddTb.Text + "',PassNat='" + natcb.SelectedItem.ToString() + "',PassGend='" + GendCb.SelectedItem.ToString() + "',PassPhone='" + PphoneTb.Text + "'where PassId=" + PidTb.Text + ";";  SqlCommand cmd = new SqlCommand(query, Con);  cmd.ExecuteNonQuery();  MessageBox.Show("Passenger Updated Sussessfully");  Con.Close();  populate();  }catch(Exception ex)  {  MessageBox.Show("Missing Information");  }  }  } |

**Tickets**

|  |
| --- |
| private void button1\_Click(object sender, EventArgs e)  {  if (Tid.Text == "" || PNatTb.Text == "")  {  MessageBox.Show("Missing Info");  }  else  {  try  {  Con.Open();  string query = "insert into TicketTb1 values(" + Tid.Text + ",'" + FCodeCb.SelectedValue.ToString() + "'," + PIdCb.SelectedValue.ToString() + ",'" + PNameTb.Text + "','" + PPassTb.Text + "','" + PNameTb.Text + "'," + PAmtTb.Text + ")";  SqlCommand cmd = new SqlCommand(query, Con);  cmd.ExecuteNonQuery();  MessageBox.Show("Ticket Booked Successfully");  Con.Close();  populate();  }  catch (Exception ex)  {  MessageBox.Show(ex.Message);  }  }  } |

**Cancellations**

|  |
| --- |
| private void deleteTicket()  {  try  {  Con.Open();  string query = "delete from TicketTb1 where Tid=" + TidCb.SelectedValue.ToString() + ";";  SqlCommand cmd = new SqlCommand(query, Con);  cmd.ExecuteNonQuery();  MessageBox.Show("Flight deleted Successfully");  Con.Close();  populate();  }  catch (Exception ex)  {  MessageBox.Show(ex.Message);  }  }  private void button2\_Click(object sender, EventArgs e)  {  if (CanId.Text == "" || FcodeTb.Text == "")  {  MessageBox.Show("Missing Info");  }  else  {  try  {  Con.Open();  string query = "insert into CancelTb1 values(" + CanId.Text + "," + TidCb.SelectedValue.ToString() + ",'" + FcodeTb.Text + "','" + CancDate.Value.Date + "')";  SqlCommand cmd = new SqlCommand(query, Con);  cmd.ExecuteNonQuery();  MessageBox.Show("Ticket Booked Successfully");  Con.Close();  populate();  deleteTicket();  }  catch (Exception ex)  {  MessageBox.Show(ex.Message);  }  }  } |

**7. Conclusion**

The development of the Flight Reservation System for "3 Friends Airline" has been achieved through collective hard work and dedication. This project showcases a user-friendly interface that simplifies the process of managing flights, passengers, tickets, and cancellations. The collaborative effort of the team has resulted in an application that is both easy to use and efficient in handling the airline's reservation needs.

While the current version of the Flight Reservation System is robust and functional, there are several areas for future enhancements:

**8. Future Enhancements**

**Error Handling:**

Improve the application's error-handling mechanisms to make it less prone to errors and enhance its stability. This includes implementing more comprehensive exception handling to ensure the application can gracefully handle unexpected situations.

**Feature Expansion:**

Add more features to increase the flexibility and functionality of the application. Potential features could include advanced search and filtering options, detailed reporting and analytics, and integration with other airline management systems.

**User Experience Improvements:**

Continuously refine the user interface to ensure it remains intuitive and easy to use. This could involve user feedback sessions to identify areas for improvement and implementing modern UI design principles.

By addressing these areas, the Flight Reservation System can evolve to meet the growing needs of "3 Friends Airline" and continue to provide a seamless user experience.

------------ ----------------------------------- -----------