|  |
| --- |
| МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ |
| федеральное государственное бюджетное образовательное учреждение высшего образования «Российский экономический университет имени Г. В. Плеханова»  Московский приборостроительный техникум |

ОТЧЕТ

по учебной практике

УП.11.01 «Разработка и эксплуатация информационных систем»

Профессионального модуля

ПМ.11 «Разработка, администрирование и защита баз данных»

Специальность 09.02.07 Информационные системы и программирование

Студент \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Дрюпин Андрей Александрович

*подпись фамилия, имя, отчество*

Группа П50-1-18

Руководитель по практической подготовке от техникума

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Митасов Никита Сергеевич

*подпись фамилия, имя, отчество*

«25» января 2021 года

СОДЕРЖАНИЕ

[ПРАКТИЧЕСКАЯ РАБОТА №1 3](#_Toc62385447)

[ПРАКТИЧЕСКАЯ РАБОТА №2 28](#_Toc62385448)

ПРАКТИЧЕСКАЯ РАБОТА №1

Схема данных: схема данных, описывающая базу данных представлена на Рисунке 1.

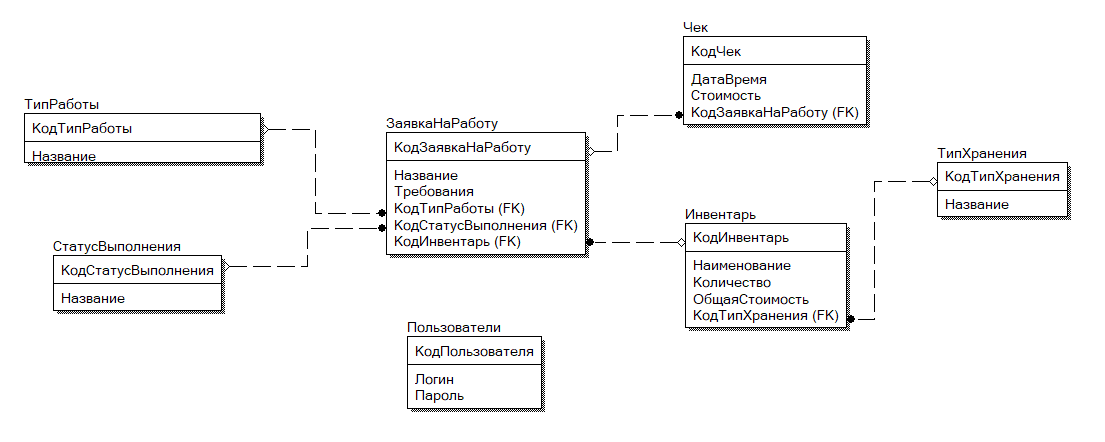


Рисунок - Схема данных

Исходный код:

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.IO;

using System.Linq;

using System.Security.Cryptography;

using System.Text;

using System.Text.RegularExpressions;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Xml;

namespace cyic

{

public partial class Form1 : Form

{

public String conStr = "Data Source=ANDREYNOTE\\FOREMDB;Initial Catalog=practice;Persist Security Info=True;User ID=sa;Password=12345";

public int selectTab = 0;

private SqlConnection connection;

private SqlCommand cmd;

private DataSet ds;

private SqlDataAdapter da;

private SqlDataReader ExecuteReader;

public Form1()

{

InitializeComponent();

GridFill(selectTab);

}

private void GridFill(int SelectedTab)

{

selectTab = SelectedTab;

switch (SelectedTab)

{

case 0:

da = new SqlDataAdapter("SELECT \* FROM Inventory", connection);

break;

case 1:

da = new SqlDataAdapter("SELECT \* FROM Recipes", connection);

break;

case 2:

da = new SqlDataAdapter("SELECT \* FROM Status", connection);

break;

case 3:

da = new SqlDataAdapter("SELECT \* FROM TypeOfStorage", connection);

break;

case 4:

da = new SqlDataAdapter("SELECT \* FROM TypeOfWork", connection);

break;

case 5:

da = new SqlDataAdapter("SELECT \* FROM Works", connection);

break;

case 6:

da = new SqlDataAdapter("SELECT \* FROM Users", connection);

break;

}

connection = new SqlConnection(conStr);

try

{

connection.Open();

ds = new DataSet();

da.Fill(ds);

dataGridView1.DataSource = ds.Tables[0];

switch (SelectedTab)

{

case 0:

DataSet dsStorage = new DataSet();

SqlDataAdapter daStorage = new SqlDataAdapter("SELECT \* FROM TypeOfStorage", connection);

daStorage.Fill(dsStorage);

InventoryStorage.DataSource = dsStorage.Tables[0];

InventoryStorage.DisplayMember = "Name";

InventoryStorage.ValueMember = "Name";

break;

case 1:

DataSet dsWork = new DataSet();

SqlDataAdapter daWork = new SqlDataAdapter("SELECT \* FROM Works", connection);

daWork.Fill(dsWork);

ReceiptNumber.DataSource = dsWork.Tables[0];

ReceiptNumber.DisplayMember = "id";

ReceiptNumber.ValueMember = "id";

break;

case 5:

DataSet dsType = new DataSet();

SqlDataAdapter daType = new SqlDataAdapter("SELECT \* FROM TypeOfWork", connection);

daType.Fill(dsType);

WorkType.DataSource = dsType.Tables[0];

WorkType.DisplayMember = "Name";

WorkType.ValueMember = "Name";

DataSet dsStatus = new DataSet();

SqlDataAdapter daStatus = new SqlDataAdapter("SELECT \* FROM Status", connection);

daStatus.Fill(dsStatus);

WorkStatus.DataSource = dsStatus.Tables[0];

WorkStatus.DisplayMember = "Name";

WorkStatus.ValueMember = "Name";

DataSet dsInventory = new DataSet();

SqlDataAdapter daInventory = new SqlDataAdapter("SELECT \* FROM Inventory", connection);

daInventory.Fill(dsInventory);

WorkInventory.DataSource = dsInventory.Tables[0];

WorkInventory.DisplayMember = "Name";

WorkInventory.ValueMember = "Name";

break;

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

finally

{

connection.Close();

}

}

//---------------------------------Status--------------------------------------

private void AddStatus\_Click(object sender, EventArgs e)

{

connection = new SqlConnection(conStr);

if (StatusName.Text.Length > 0 && StatusName.Text.Length <= 50)

{

try

{

connection.Open();

cmd = new SqlCommand(String.Format("INSERT INTO Status (Name) VALUES ('{0}')", StatusName.Text), connection);

cmd.ExecuteNonQuery();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

finally

{

connection.Close();

}

GridFill(selectTab);

}

else MessageBox.Show("Проверьте правильность заполнения полей");

}

private void DeleteStatus\_Click(object sender, EventArgs e)

{

connection = new SqlConnection(conStr);

try

{

connection.Open();

cmd = new SqlCommand(String.Format("DELETE FROM Status WHERE ID={0}", dataGridView1.SelectedRows[0].Cells[0].Value), connection);

cmd.ExecuteNonQuery();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

finally

{

connection.Close();

}

GridFill(selectTab);

}

private void UpdateStatus\_Click(object sender, EventArgs e)

{

if (StatusName.Text.Length > 0 && StatusName.Text.Length <= 50)

{

connection = new SqlConnection(conStr);

try

{

connection.Open();

cmd = new SqlCommand(String.Format("UPDATE Status SET Name='{0}' WHERE ID={1}", StatusName.Text, dataGridView1.SelectedRows[0].Cells[0].Value), connection);

cmd.ExecuteNonQuery();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

finally

{

connection.Close();

}

GridFill(selectTab);

}

else MessageBox.Show("Проверьте правильность заполнения полей");

}

//---------------------------------Status--------------------------------------

//---------------------------------TypeOfStorage--------------------------------------

private void AddStorage\_Click(object sender, EventArgs e)

{

connection = new SqlConnection(conStr);

if (StorageName.Text.Length > 0 && StorageName.Text.Length <= 50)

{

try

{

connection.Open();

cmd = new SqlCommand(String.Format("INSERT INTO TypeOfStorage (Name) VALUES ('{0}')", StorageName.Text), connection);

cmd.ExecuteNonQuery();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

finally

{

connection.Close();

}

GridFill(selectTab);

}

else MessageBox.Show("Проверьте правильность заполнения полей");

}

private void DeleteStorage\_Click(object sender, EventArgs e)

{

connection = new SqlConnection(conStr);

try

{

connection.Open();

cmd = new SqlCommand(String.Format("DELETE FROM TypeOfStorage WHERE ID={0}", dataGridView1.SelectedRows[0].Cells[0].Value), connection);

cmd.ExecuteNonQuery();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

finally

{

connection.Close();

}

GridFill(selectTab);

}

private void UpdateStorage\_Click(object sender, EventArgs e)

{

if (StorageName.Text.Length > 0 && StorageName.Text.Length <= 50)

{

connection = new SqlConnection(conStr);

try

{

connection.Open();

cmd = new SqlCommand(String.Format("UPDATE TypeOfStorage SET Name='{0}' WHERE ID={1}", StorageName.Text, dataGridView1.SelectedRows[0].Cells[0].Value), connection);

cmd.ExecuteNonQuery();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

finally

{

connection.Close();

}

GridFill(selectTab);

}

else MessageBox.Show("Проверьте правильность заполнения полей");

}

//---------------------------------TypeOfStorage--------------------------------------

//---------------------------------TypeOfWork--------------------------------------

private void AddType\_Click(object sender, EventArgs e)

{

connection = new SqlConnection(conStr);

if (TypeName.Text.Length > 0 && TypeName.Text.Length <= 50)

{

try

{

connection.Open();

cmd = new SqlCommand(String.Format("INSERT INTO TypeOfWork (Name) VALUES ('{0}')", TypeName.Text), connection);

cmd.ExecuteNonQuery();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

finally

{

connection.Close();

}

GridFill(selectTab);

}

else MessageBox.Show("Проверьте правильность заполнения полей");

}

private void DeleteType\_Click(object sender, EventArgs e)

{

connection = new SqlConnection(conStr);

try

{

connection.Open();

cmd = new SqlCommand(String.Format("DELETE FROM TypeOfWork WHERE ID={0}", dataGridView1.SelectedRows[0].Cells[0].Value), connection);

cmd.ExecuteNonQuery();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

finally

{

connection.Close();

}

GridFill(selectTab);

}

private void UpdateType\_Click(object sender, EventArgs e)

{

if (TypeName.Text.Length > 0 && TypeName.Text.Length <= 50)

{

connection = new SqlConnection(conStr);

try

{

connection.Open();

cmd = new SqlCommand(String.Format("UPDATE TypeOfWork SET Name='{0}' WHERE ID={1}", TypeName.Text, dataGridView1.SelectedRows[0].Cells[0].Value), connection);

cmd.ExecuteNonQuery();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

finally

{

connection.Close();

}

GridFill(selectTab);

}

else MessageBox.Show("Проверьте правильность заполнения полей");

}

//---------------------------------TypeOfWork--------------------------------------

//---------------------------------Inventory--------------------------------------

private void AddInventory\_Click(object sender, EventArgs e)

{

if ((InventoryName.Text.Length > 0 && InventoryName.Text.Length <= 50) && InventoryQuantity.Value >= 0 && InventoryCost.Value >= 0)

{

try

{

connection.Open();

cmd = new SqlCommand(String.Format("INSERT INTO Inventory (Name, Quantity, Cost, StorageType) VALUES ('{0}', '{1}', '{2}', '{3}')", InventoryName.Text, InventoryQuantity.Value, InventoryCost.Value, InventoryStorage.SelectedValue), connection);

cmd.ExecuteNonQuery();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

finally

{

connection.Close();

}

GridFill(selectTab);

}

else MessageBox.Show("Проверьте правильность заполнения полей");

}

private void DeleteInventory\_Click(object sender, EventArgs e)

{

connection = new SqlConnection(conStr);

try

{

connection.Open();

cmd = new SqlCommand(String.Format("DELETE FROM Inventory WHERE ID={0}", dataGridView1.SelectedRows[0].Cells[0].Value), connection);

cmd.ExecuteNonQuery();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

finally

{

connection.Close();

}

GridFill(selectTab);

}

private void UpdateInventory\_Click(object sender, EventArgs e)

{

if ((InventoryName.Text.Length > 0 && InventoryName.Text.Length <= 50) && InventoryQuantity.Value >= 0 && InventoryCost.Value >= 0)

{

connection = new SqlConnection(conStr);

try

{

connection.Open();

cmd = new SqlCommand(String.Format("UPDATE Inventory SET Name='{0}', Quantity='{1}', Cost='{2}', StorageType='{3}' WHERE ID={4}", InventoryName.Text, InventoryQuantity.Value, InventoryCost.Value, InventoryStorage.SelectedValue, dataGridView1.SelectedRows[0].Cells[0].Value), connection);

cmd.ExecuteNonQuery();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

finally

{

connection.Close();

}

GridFill(selectTab);

}

else MessageBox.Show("Проверьте правильность заполнения полей");

}

//---------------------------------Inventory--------------------------------------

//---------------------------------Works--------------------------------------

private void AddWork\_Click(object sender, EventArgs e)

{

if ((WorkName.Text.Length > 0 && WorkName.Text.Length <= 50) && WorkTask.Text.Length > 0)

{

try

{

connection.Open();

cmd = new SqlCommand(String.Format("INSERT INTO Works (Name, Task, TypeOfWork, Status, Inventory) VALUES ('{0}', '{1}', '{2}', '{3}', '{4}')", WorkName.Text, WorkTask.Text, WorkType.SelectedValue, WorkStatus.SelectedValue, WorkInventory.SelectedValue), connection);

cmd.ExecuteNonQuery();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

finally

{

connection.Close();

}

GridFill(selectTab);

}

else MessageBox.Show("Проверьте правильность заполнения полей");

}

private void DeleteWork\_Click(object sender, EventArgs e)

{

connection = new SqlConnection(conStr);

try

{

connection.Open();

cmd = new SqlCommand(String.Format("DELETE FROM Works WHERE ID={0}", dataGridView1.SelectedRows[0].Cells[0].Value), connection);

cmd.ExecuteNonQuery();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

finally

{

connection.Close();

}

GridFill(selectTab);

}

private void UpdateWork\_Click(object sender, EventArgs e)

{

if ((WorkName.Text.Length > 0 && WorkName.Text.Length <= 50) && WorkTask.Text.Length > 0)

{

connection = new SqlConnection(conStr);

try

{

connection.Open();

cmd = new SqlCommand(String.Format("UPDATE Works SET Name='{0}', Task='{1}', TypeOfWork='{2}', Status='{3}', Inventory='{4}' WHERE ID={5}", WorkName.Text, WorkTask.Text, WorkType.SelectedValue, WorkStatus.SelectedValue, WorkInventory.SelectedValue, dataGridView1.SelectedRows[0].Cells[0].Value), connection);

cmd.ExecuteNonQuery();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

finally

{

connection.Close();

}

GridFill(selectTab);

}

else MessageBox.Show("Проверьте правильность заполнения полей");

}

//---------------------------------Works--------------------------------------

//---------------------------------Receipts--------------------------------------

private void AddReceipt\_Click(object sender, EventArgs e)

{

if (ReceiptCost.Value >= 0 && ReceiptNumber.SelectedIndex > -1 && ReceiptDate.Value <= DateTime.Now)

{

try

{

connection.Open();

cmd = new SqlCommand(String.Format("INSERT INTO Recipes (Date, Cost, WorkId) VALUES ('{0}', '{1}', '{2}')", ReceiptDate.Value.ToShortDateString(), ReceiptCost.Value, ReceiptNumber.SelectedValue), connection);

cmd.ExecuteNonQuery();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

finally

{

connection.Close();

}

GridFill(selectTab);

}

else MessageBox.Show("Проверьте правильность заполнения полей");

}

private void DeleteReceipt\_Click(object sender, EventArgs e)

{

connection = new SqlConnection(conStr);

try

{

connection.Open();

cmd = new SqlCommand(String.Format("DELETE FROM Recipes WHERE ID={0}", dataGridView1.SelectedRows[0].Cells[0].Value), connection);

cmd.ExecuteNonQuery();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

finally

{

connection.Close();

}

GridFill(selectTab);

}

//---------------------------------Recepts--------------------------------------

//---------------------------------Users--------------------------------------

private void DeleteUser\_Click(object sender, EventArgs e)

{

connection = new SqlConnection(conStr);

try

{

connection.Open();

cmd = new SqlCommand(String.Format("DELETE FROM Users WHERE ID={0}", dataGridView1.SelectedRows[0].Cells[0].Value), connection);

cmd.ExecuteNonQuery();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

finally

{

connection.Close();

}

GridFill(selectTab);

}

private void UpdateUser\_Click(object sender, EventArgs e)

{

if (UserLogin.Text.Length > 0 && UserLogin.Text.Length <= 50)

{

connection = new SqlConnection(conStr);

try

{

connection.Open();

cmd = new SqlCommand(String.Format("UPDATE Users SET login='{0}' WHERE ID={1}", UserLogin.Text, dataGridView1.SelectedRows[0].Cells[0].Value), connection);

cmd.ExecuteNonQuery();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

finally

{

connection.Close();

}

GridFill(selectTab);

}

else MessageBox.Show("Проверьте правильность заполнения полей");

}

//---------------------------------Users--------------------------------------

//---------------------------------Auth--------------------------------------

private void UserReg\_Click(object sender, EventArgs e)

{

string Login = UserLogBox.Text;

string Password = UserPassBox.Text;

string SecurePass = "";

bool ok = true;

Regex regex = new Regex(@"^(?=.\*[a-z])(?=.\*[A-Z])(?=.\*\d)(?=.\*[$@$!%\*?&])[A-Za-z\d$@$!%\*?&]{8,}");

if (regex.IsMatch(Password) && Login.Length >= 3)

{

byte[] hash = Encoding.ASCII.GetBytes(Password);

MD5 md5 = new MD5CryptoServiceProvider();

byte[] hashenc = md5.ComputeHash(hash);

string halfresult = "";

foreach (var b in hashenc)

{

halfresult += b.ToString("x2");

}

byte[] hashRev = Encoding.ASCII.GetBytes(Reverce(halfresult+"hY$T37@84#hJF!2"));

MD5 md5Rev = new MD5CryptoServiceProvider();

byte[] hashencRev = md5Rev.ComputeHash(hashRev);

foreach (var b in hashencRev)

{

SecurePass += b.ToString("x2");

}

connection = new SqlConnection(conStr);

try

{

connection.Open();

cmd = new SqlCommand(String.Format("SELECT \* FROM Users WHERE login='{0}'", Login), connection);

ExecuteReader = cmd.ExecuteReader();

while (ExecuteReader.Read()) if (ExecuteReader["id"].ToString().Length > 0) ok = false;

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

finally

{

connection.Close();

}

connection = new SqlConnection(conStr);

if (ok)

{

try

{

connection.Open();

cmd = new SqlCommand(String.Format("INSERT INTO Users (login, password) VALUES ('{0}','{1}')", Login, SecurePass), connection);

cmd.ExecuteNonQuery();

MessageBox.Show("Упех!");

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

finally

{

connection.Close();

}

GridFill(selectTab);

}

else MessageBox.Show("Такой логин уже существует");

}

else MessageBox.Show("Пароль/логин не соответствует условиям");

}

private void UserLog\_Click(object sender, EventArgs e)

{

string Login = UserLogBox.Text;

string Password = UserPassBox.Text;

string SecurePass = "";

Regex = new Regex(@"^(?=.\*[a-z])(?=.\*[A-Z])(?=.\*\d)(?=.\*[$@$!%\*?&])[A-Za-z\d$@$!%\*?&]{8,}");

if (regex.IsMatch(Password) && Login.Length >= 3)

{

byte[] hash = Encoding.ASCII.GetBytes(Password);

MD5 = new MD5CryptoServiceProvider();

byte[] hashenc = md5.ComputeHash(hash);

string halfresult = "";

foreach (var b in hashenc)

{

halfresult += b.ToString("x2");

}

byte[] hashRev = Encoding.ASCII.GetBytes(Reverce(halfresult + "hY$T37@84#hJF!2"));

MD5 md5Rev = new MD5CryptoServiceProvider();

byte[] hashencRev = md5Rev.ComputeHash(hashRev);

foreach (var b in hashencRev)

{

SecurePass += b.ToString("x2");

}

connection = new SqlConnection(conStr);

try

{

connection.Open();

cmd = new SqlCommand(String.Format("SELECT \* FROM Users WHERE login='{0}'", Login), connection);

ExecuteReader = cmd.ExecuteReader();

while (ExecuteReader.Read())

{

if (ExecuteReader["password"].ToString() == SecurePass) MessageBox.Show("Добро пожаловать, " + Login + "!");

else MessageBox.Show("Неправильный логин или пароль");

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

finally

{

connection.Close();

}

}

else MessageBox.Show("Пароль/логин не соответствует условиям");

}

//---------------------------------Auth--------------------------------------

private void tabControl1\_Selected(object sender, TabControlEventArgs e)

{

//MessageBox.Show(e.TabPageIndex.ToString());

GridFill(e.TabPageIndex);

}

private void dataGridView1\_CellContentClick(object sender, DataGridViewCellEventArgs e)

{

try

{

switch (selectTab)

{

case 0:

InventoryName.Text = dataGridView1.Rows[e.RowIndex].Cells[1].Value.ToString();

InventoryQuantity.Value = Convert.ToDecimal(dataGridView1.Rows[e.RowIndex].Cells[2].Value);

InventoryCost.Value = Convert.ToDecimal(dataGridView1.Rows[e.RowIndex].Cells[3].Value);

InventoryStorage.SelectedValue = dataGridView1.Rows[e.RowIndex].Cells[4].Value.ToString();

break;

case 1:

ReceiptDate.Value = DateTime.Parse(dataGridView1.Rows[e.RowIndex].Cells[1].Value.ToString());

ReceiptCost.Value = Convert.ToDecimal(dataGridView1.Rows[e.RowIndex].Cells[2].Value);

ReceiptNumber.SelectedValue = dataGridView1.Rows[e.RowIndex].Cells[3].Value.ToString();

break;

case 2:

StatusName.Text = dataGridView1.Rows[e.RowIndex].Cells[1].Value.ToString();

break;

case 3:

StorageName.Text = dataGridView1.Rows[e.RowIndex].Cells[1].Value.ToString();

break;

case 4:

TypeName.Text = dataGridView1.Rows[e.RowIndex].Cells[1].Value.ToString();

break;

case 5:

WorkName.Text = dataGridView1.Rows[e.RowIndex].Cells[1].Value.ToString();

WorkTask.Text = dataGridView1.Rows[e.RowIndex].Cells[2].Value.ToString();

WorkType.SelectedValue = dataGridView1.Rows[e.RowIndex].Cells[3].Value.ToString();

WorkStatus.SelectedValue = dataGridView1.Rows[e.RowIndex].Cells[4].Value.ToString();

WorkInventory.SelectedValue = dataGridView1.Rows[e.RowIndex].Cells[5].Value.ToString();

break;

case 6:

UserLogin.Text = dataGridView1.Rows[e.RowIndex].Cells[1].Value.ToString();

break;

}

}

catch { }

}

public static string Reverce(string str)

{

return new string(str.ToCharArray().Reverse().ToArray());

}

private void ExportExel\_Click(object sender, EventArgs e)

{

SaveFileDialog saveFileDialog1 = new SaveFileDialog();

saveFileDialog1.Filter = "Excel files (\*.xls)|\*.xls";

saveFileDialog1.FilterIndex = 2;

saveFileDialog1.RestoreDirectory = true;

if (saveFileDialog1.ShowDialog() == DialogResult.OK)

{

try

{

using (ExcelWriter writer = new ExcelWriter(saveFileDialog1.FileName))

{

writer.WriteStartDocument();

// Write the worksheet contents

writer.WriteStartWorksheet("Чек");

//Write header row

writer.WriteStartRow();

writer.WriteExcelUnstyledCell("Id");

writer.WriteExcelUnstyledCell("Дата");

writer.WriteExcelUnstyledCell("Стоимость");

writer.WriteExcelUnstyledCell("Номер заказа");

writer.WriteEndRow();

//write data

foreach (DataRow row in ds.Tables[0].Rows)

{

writer.WriteStartRow();

foreach (object o in row.ItemArray)

{

writer.WriteExcelAutoStyledCell(o);

}

writer.WriteEndRow();

}

// Close up the document

writer.WriteEndWorksheet();

GridFill(5);

writer.WriteStartWorksheet("Заказы");

//Write header row

writer.WriteStartRow();

writer.WriteExcelUnstyledCell("Id");

writer.WriteExcelUnstyledCell("Название");

writer.WriteExcelUnstyledCell("ТЗ");

writer.WriteExcelUnstyledCell("Тип работ");

writer.WriteExcelUnstyledCell("Статус");

writer.WriteExcelUnstyledCell("Инвентарь");

writer.WriteEndRow();

//write data

foreach (DataRow row in ds.Tables[0].Rows)

{

writer.WriteStartRow();

foreach (object o in row.ItemArray)

{

writer.WriteExcelAutoStyledCell(o);

}

writer.WriteEndRow();

}

// Close up the document

writer.WriteEndWorksheet();

writer.WriteEndDocument();

writer.Close();

GridFill(1);

}

}

catch (Exception myException)

{

}

}

}

}

public class ExcelWriter : IDisposable

{

private XmlWriter \_writer;

public enum CellStyle { General, Number, Currency, DateTime, ShortDate };

public void WriteStartDocument()

{

if (\_writer == null) throw new InvalidOperationException("Cannot write after closing.");

\_writer.WriteProcessingInstruction("mso-application", "progid=\"Excel.Sheet\"");

\_writer.WriteStartElement("ss", "Workbook", "urn:schemas-microsoft-com:office:spreadsheet");

WriteExcelStyles();

}

public void WriteEndDocument()

{

if (\_writer == null) throw new InvalidOperationException("Cannot write after closing.");

\_writer.WriteEndElement();

}

private void WriteExcelStyleElement(CellStyle style)

{

\_writer.WriteStartElement("Style", "urn:schemas-microsoft-com:office:spreadsheet");

\_writer.WriteAttributeString("ID", "urn:schemas-microsoft-com:office:spreadsheet", style.ToString());

\_writer.WriteEndElement();

}

private void WriteExcelStyleElement(CellStyle style, string NumberFormat)

{

\_writer.WriteStartElement("Style", "urn:schemas-microsoft-com:office:spreadsheet");

\_writer.WriteAttributeString("ID", "urn:schemas-microsoft-com:office:spreadsheet", style.ToString());

\_writer.WriteStartElement("NumberFormat", "urn:schemas-microsoft-com:office:spreadsheet");

\_writer.WriteAttributeString("Format", "urn:schemas-microsoft-com:office:spreadsheet", NumberFormat);

\_writer.WriteEndElement();

\_writer.WriteEndElement();

}

private void WriteExcelStyles()

{

\_writer.WriteStartElement("Styles", "urn:schemas-microsoft-com:office:spreadsheet");

WriteExcelStyleElement(CellStyle.General);

WriteExcelStyleElement(CellStyle.Number, "General Number");

WriteExcelStyleElement(CellStyle.DateTime, "General Date");

WriteExcelStyleElement(CellStyle.Currency, "Currency");

WriteExcelStyleElement(CellStyle.ShortDate, "Short Date");

\_writer.WriteEndElement();

}

public void WriteStartWorksheet(string name)

{

if (\_writer == null) throw new InvalidOperationException("Cannot write after closing.");

\_writer.WriteStartElement("Worksheet", "urn:schemas-microsoft-com:office:spreadsheet");

\_writer.WriteAttributeString("Name", "urn:schemas-microsoft-com:office:spreadsheet", name);

\_writer.WriteStartElement("Table", "urn:schemas-microsoft-com:office:spreadsheet");

}

public void WriteEndWorksheet()

{

if (\_writer == null) throw new InvalidOperationException("Cannot write after closing.");

\_writer.WriteEndElement();

\_writer.WriteEndElement();

}

public ExcelWriter(string outputFileName)

{

XmlWriterSettings settings = new XmlWriterSettings();

settings.Indent = true;

\_writer = XmlWriter.Create(outputFileName, settings);

}

public void Close()

{

if (\_writer == null) throw new InvalidOperationException("Already closed.");

\_writer.Close();

\_writer = null;

}

public void WriteExcelColumnDefinition(int columnWidth)

{

if (\_writer == null) throw new InvalidOperationException("Cannot write after closing.");

\_writer.WriteStartElement("Column", "urn:schemas-microsoft-com:office:spreadsheet");

\_writer.WriteStartAttribute("Width", "urn:schemas-microsoft-com:office:spreadsheet");

\_writer.WriteValue(columnWidth);

\_writer.WriteEndAttribute();

\_writer.WriteEndElement();

}

public void WriteExcelUnstyledCell(string value)

{

if (\_writer == null) throw new InvalidOperationException("Cannot write after closing.");

\_writer.WriteStartElement("Cell", "urn:schemas-microsoft-com:office:spreadsheet");

\_writer.WriteStartElement("Data", "urn:schemas-microsoft-com:office:spreadsheet");

\_writer.WriteAttributeString("Type", "urn:schemas-microsoft-com:office:spreadsheet", "String");

\_writer.WriteValue(value);

\_writer.WriteEndElement();

\_writer.WriteEndElement();

}

public void WriteStartRow()

{

if (\_writer == null) throw new InvalidOperationException("Cannot write after closing.");

\_writer.WriteStartElement("Row", "urn:schemas-microsoft-com:office:spreadsheet");

}

public void WriteEndRow()

{

if (\_writer == null) throw new InvalidOperationException("Cannot write after closing.");

\_writer.WriteEndElement();

}

public void WriteExcelStyledCell(object value, CellStyle style)

{

if (\_writer == null) throw new InvalidOperationException("Cannot write after closing.");

\_writer.WriteStartElement("Cell", "urn:schemas-microsoft-com:office:spreadsheet");

\_writer.WriteAttributeString("StyleID", "urn:schemas-microsoft-com:office:spreadsheet", style.ToString());

\_writer.WriteStartElement("Data", "urn:schemas-microsoft-com:office:spreadsheet");

switch (style)

{

case CellStyle.General:

\_writer.WriteAttributeString("Type", "urn:schemas-microsoft-com:office:spreadsheet", "String");

break;

case CellStyle.Number:

case CellStyle.Currency:

\_writer.WriteAttributeString("Type", "urn:schemas-microsoft-com:office:spreadsheet", "Number");

break;

case CellStyle.ShortDate:

case CellStyle.DateTime:

\_writer.WriteAttributeString("Type", "urn:schemas-microsoft-com:office:spreadsheet", "DateTime");

break;

}

\_writer.WriteValue(value);

// tag += String.Format("{1}\"><ss:Data ss:Type=\"DateTime\">{0:yyyy\\-MM\\-dd\\THH\\:mm\\:ss\\.fff}</ss:Data>", value,

\_writer.WriteEndElement();

\_writer.WriteEndElement();

}

public void WriteExcelAutoStyledCell(object value)

{

if (\_writer == null) throw new InvalidOperationException("Cannot write after closing.");

//write the <ss:Cell> and <ss:Data> tags for something

if (value is Int16 || value is Int32 || value is Int64 || value is SByte ||

value is UInt16 || value is UInt32 || value is UInt64 || value is Byte)

{

WriteExcelStyledCell(value, CellStyle.Number);

}

else if (value is Single || value is Double || value is Decimal) //we'll assume it's a currency

{

WriteExcelStyledCell(value, CellStyle.Currency);

}

else if (value is DateTime)

{

//check if there's no time information and use the appropriate style

WriteExcelStyledCell(value, ((DateTime)value).TimeOfDay.CompareTo(new TimeSpan(0, 0, 0, 0, 0)) == 0 ? CellStyle.ShortDate : CellStyle.DateTime);

}

else

{

WriteExcelStyledCell(value, CellStyle.General);

}

}

#region IDisposable Members

public void Dispose()

{

if (\_writer == null)

return;

\_writer.Close();

\_writer = null;

}

#endregion

}

}

Результат работы программы:

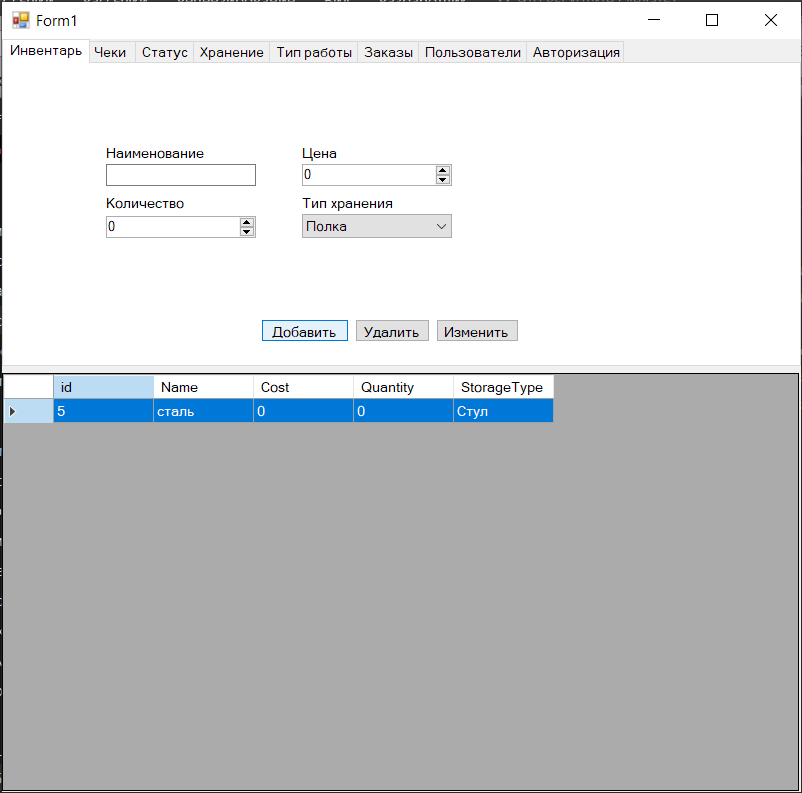


Рисунок – Вкладка 1

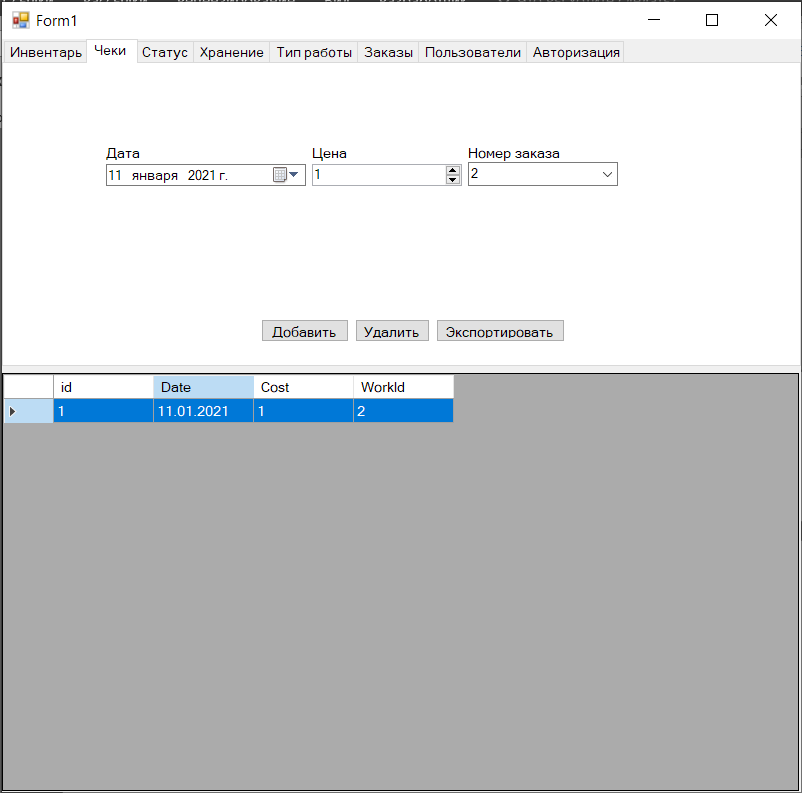


Рисунок – Вкладка 2

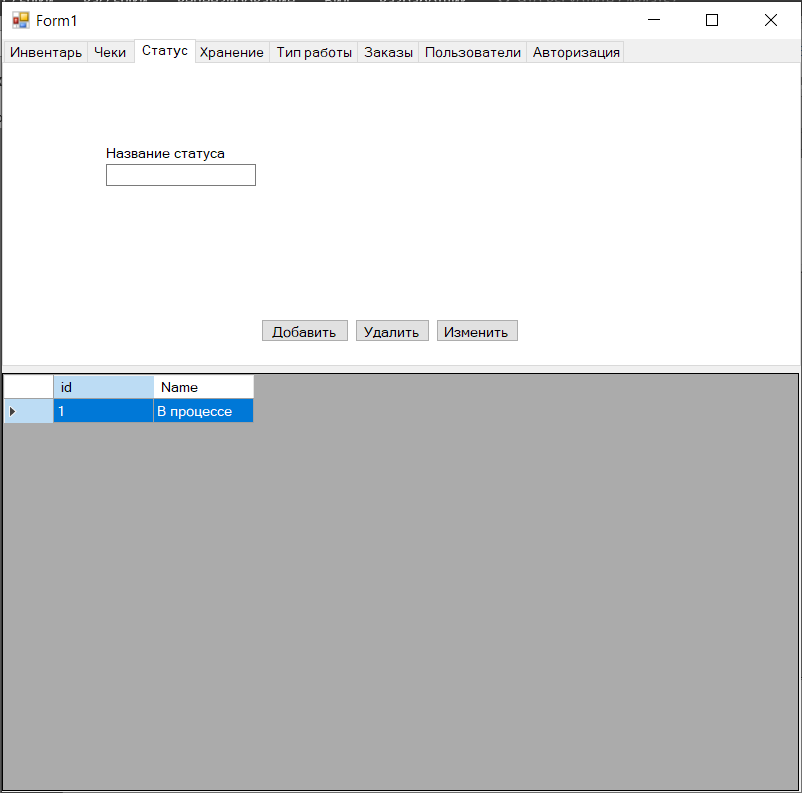


Рисунок – Вкладка 3

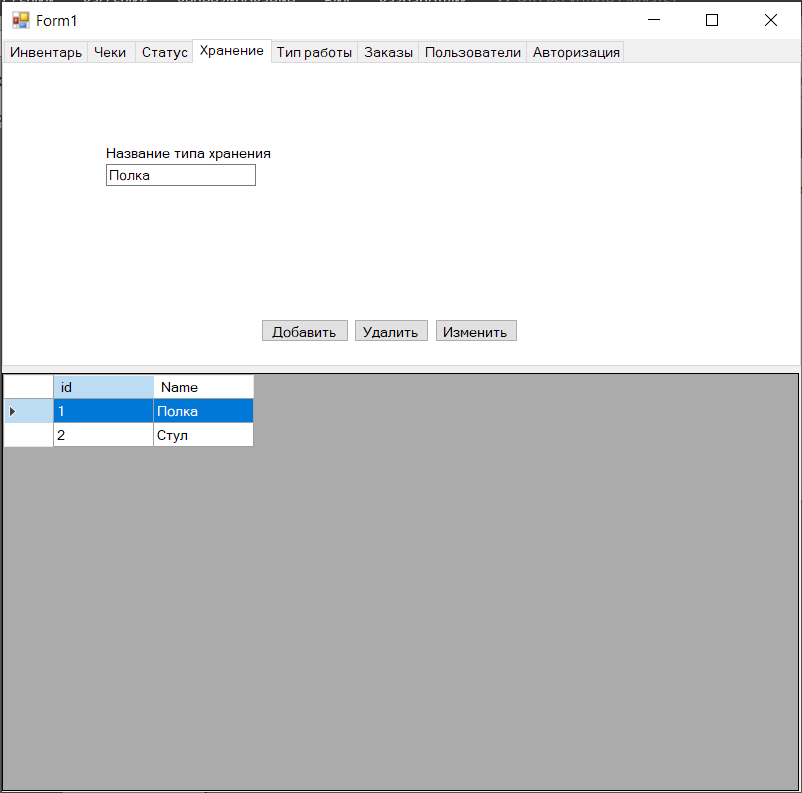


Рисунок – Вкладка 4

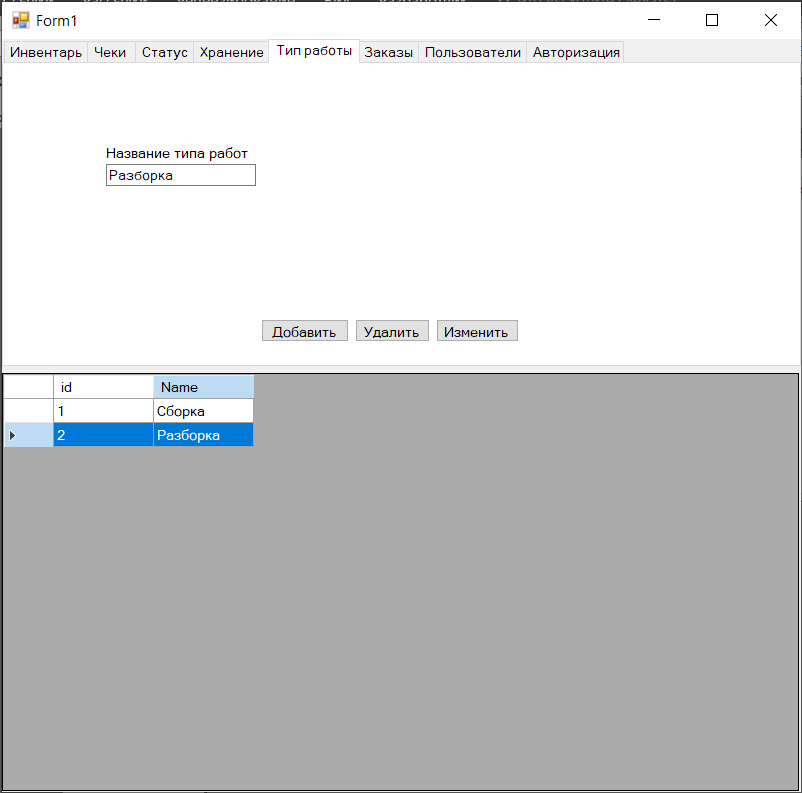


Рисунок - Вкладка 5

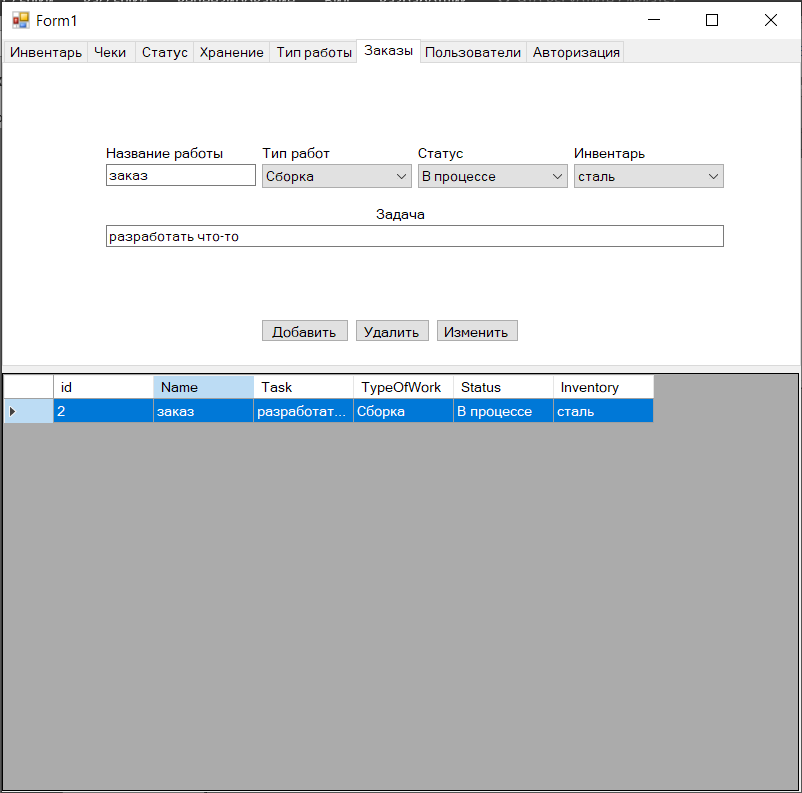


Рисунок - Вкладка 6

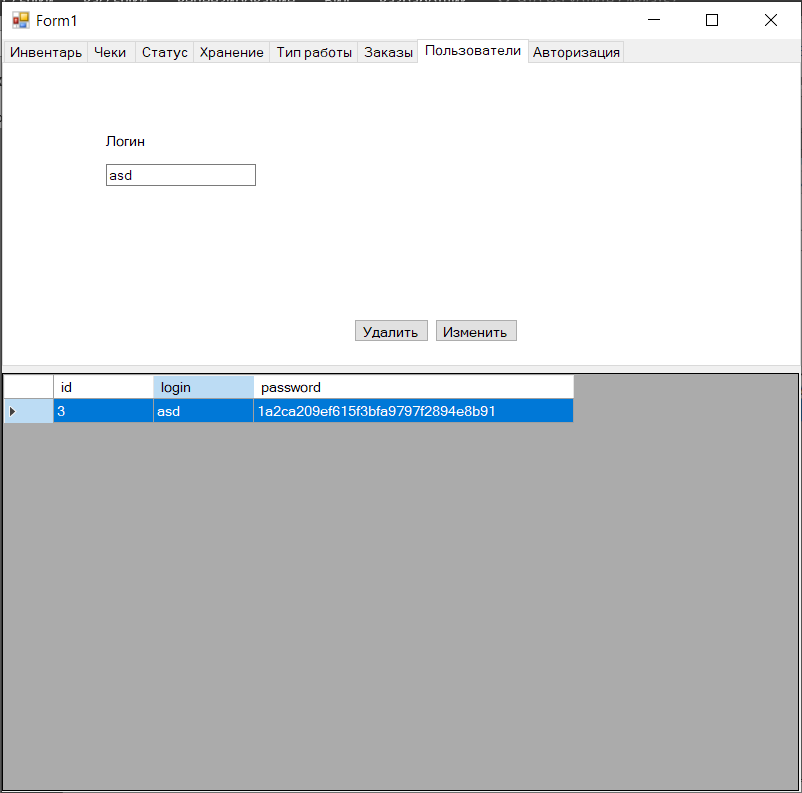


Рисунок - Вкладка 7

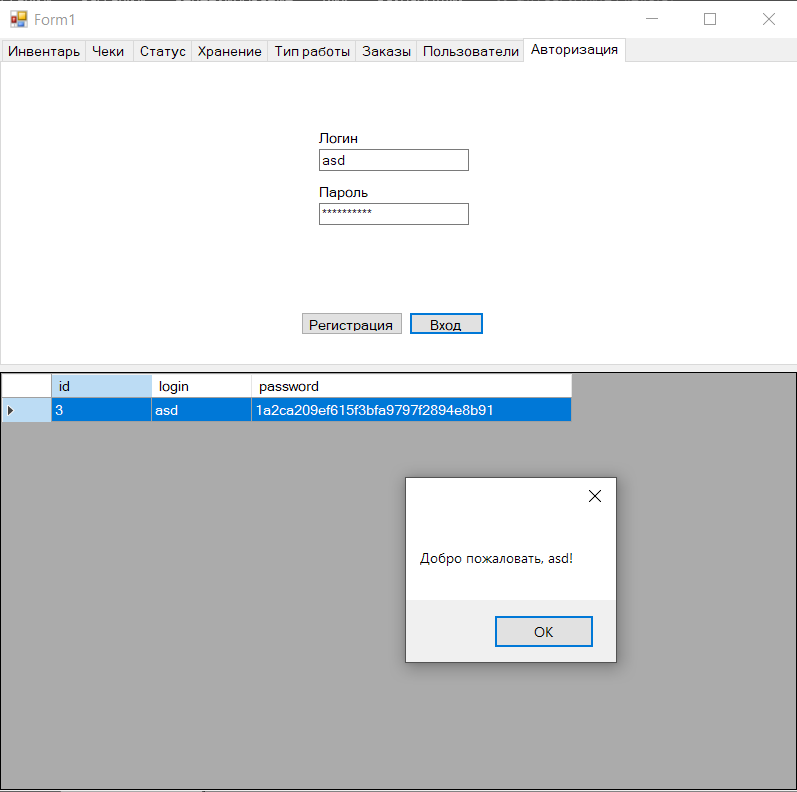


Рисунок - Вкладка 8

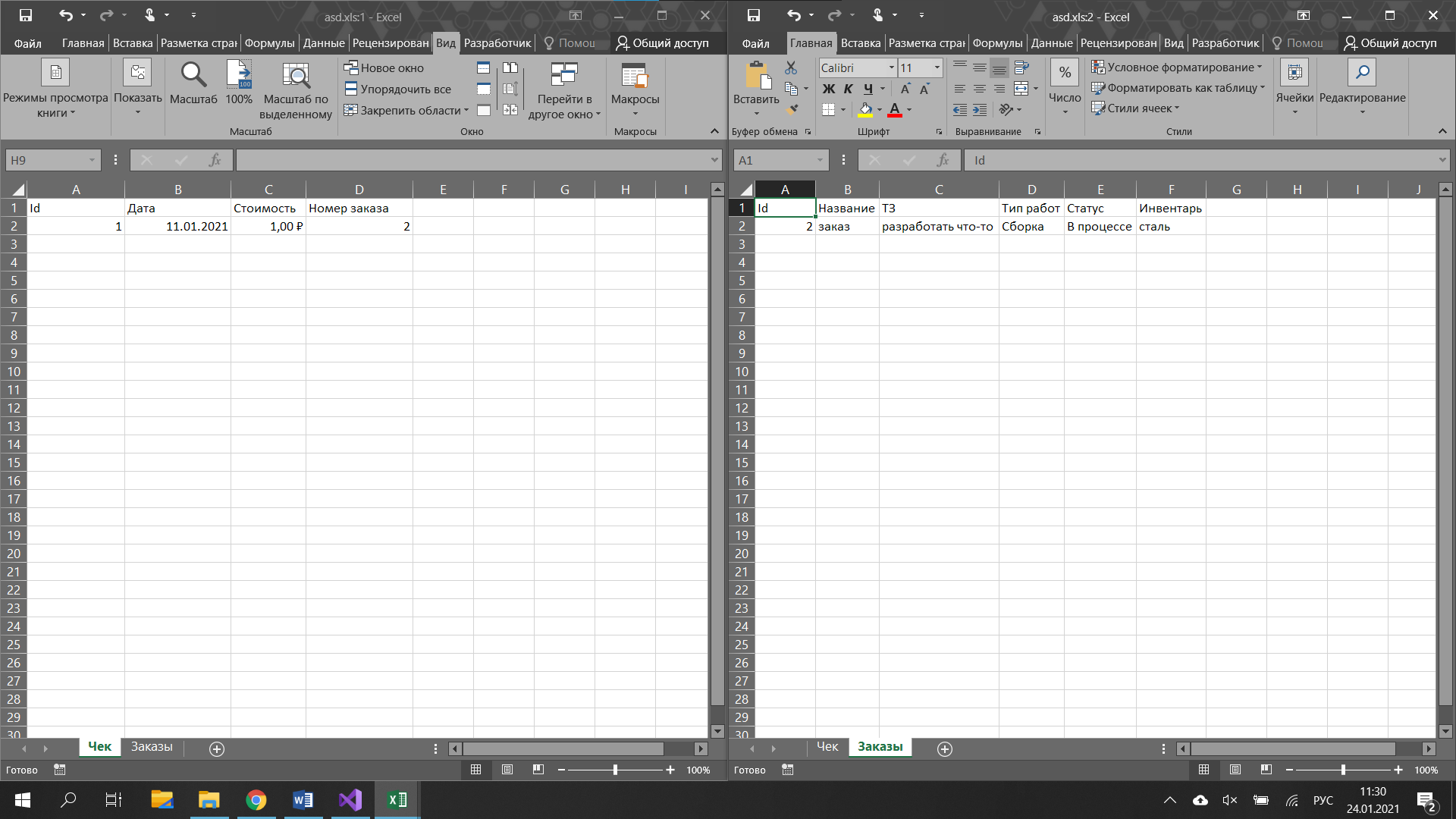


Рисунок – Результат экспорта

ПРАКТИЧЕСКАЯ РАБОТА №2

Схема данных: схема данных, описывающая базу данных представлена на Рисунке 2.

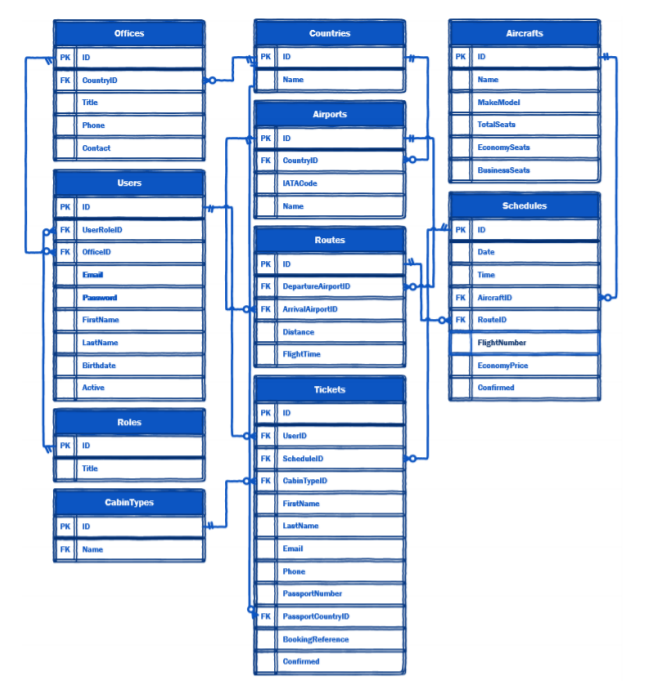


Рисунок - Схема данных

Исходный код:

Листинг файла UserPanel.cs

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Avia

{

public partial class UserPanel : Form

{

public String conStr = "Data Source=ANDREYNOTE\\FOREMDB;Initial Catalog=session;Persist Security Info=True;User ID=sa;Password=12345";

private SqlConnection connection;

private SqlCommand cmd;

private DataSet DS;

private SqlDataAdapter da;

private SqlDataReader ExecuteReader;

bool ExitOK = false;

int UserID = 0;

string UserName = "";

public UserPanel(int id, string name)

{

InitializeComponent();

UserID = id;

UserName = name;

UpdateDataGridUsers();

SaveExit();

UserPanelWelcome\_Label.Text = "Hi " + UserName + ", Welcome to AMONIC Airlines.";

}

public void UpdateDataGridUsers()

{

connection = new SqlConnection(conStr);

try

{

connection.Open();

DS = new DataSet();

SqlDataAdapter DAUser = new SqlDataAdapter(String.Format("SELECT Tracks.Date AS Date, Tracks.Login AS 'Login time', Tracks.Logout AS 'Logout time', DATEDIFF (minute, Tracks.Login, Tracks.Logout) AS 'Time spent on system',Tracks.Reason AS 'Unsuccessful logout reason' FROM Tracks WHERE UserID = {0} AND ExitOK = 'True'", UserID), connection);

DAUser.Fill(DS);

UserGridUser\_View.DataSource = DS.Tables[0];

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

finally

{

connection.Close();

}

int ErrorCounter = 0;

int SpendTime = 0;

foreach (DataGridViewRow row in UserGridUser\_View.Rows)

{

if (row.Cells[4].Value.ToString().Length > 0) ErrorCounter++;

if (row.Cells[3].Value.ToString().Length > 0) SpendTime += Convert.ToInt32(row.Cells[3].Value);

}

UserPanelErrors\_Label.Text = "Number of crashes: " + ErrorCounter.ToString();

UserPanelTime\_Label.Text = "Time spent on system: " + SpendTime.ToString();

}

private void exitToolStripMenuItem\_Click(object sender, EventArgs e)

{

this.Close();

}

private void UserPanel\_FormClosed(object sender, FormClosedEventArgs e)

{

SaveExit();

}

private void SaveExit()

{

connection = new SqlConnection(conStr);

try

{

connection.Open();

cmd = new SqlCommand(String.Format("SELECT \* FROM Tracks WHERE UserID={0}", UserID), connection);

ExecuteReader = cmd.ExecuteReader();

while (ExecuteReader.Read())

{

if (ExecuteReader["ExitOK"].ToString() == "False" && !ExitOK)

{

ExitReason ExitReasonForm = new ExitReason(ExecuteReader["ID"].ToString(), ExecuteReader["Date"].ToString(), ExecuteReader["Login"].ToString());

ExitReasonForm.Show();

}

}

ExecuteReader.Close();

if (!ExitOK)

{

cmd = new SqlCommand(String.Format("INSERT INTO Tracks (UserID, Date, Login, Logout, ExitOK, Reason) VALUES ('{0}', '{1}', '{2}', '{3}', '{4}', '{5}')", UserID, DateTime.Now.ToShortDateString(), DateTime.Now.ToString("HH:mm:ss"), null, "False", null), connection);

cmd.ExecuteNonQuery();

ExitOK = true;

}

else

{

cmd = new SqlCommand(String.Format("UPDATE Tracks SET Logout='{0}', ExitOK='{1}' WHERE UserID={2} AND ID = (SELECT MAX(ID) FROM Users)", DateTime.Now.ToString("HH:mm:ss"), "True", UserID), connection);

cmd.ExecuteNonQuery();

Application.Exit();

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

finally

{

connection.Close();

}

}

private void searchFlightToolStripMenuItem\_Click(object sender, EventArgs e)

{

SearchFlight SearchFlightForm = new SearchFlight(UserID);

SearchFlightForm.Show();

}

private void userFeedbackToolStripMenuItem\_Click(object sender, EventArgs e)

{

QuestionsSummary QuestionsSummaryForm = new QuestionsSummary();

QuestionsSummaryForm.Show();

}

private void amenitiesToolStripMenuItem\_Click(object sender, EventArgs e)

{

AmenitiesPurchase AmenitiesPurchaseForm = new AmenitiesPurchase();

AmenitiesPurchaseForm.Show();

}

private void reportToolStripMenuItem\_Click(object sender, EventArgs e)

{

AmenitiesReport AmenitiesReportForm = new AmenitiesReport();

AmenitiesReportForm.Show();

}

private void summaryToolStripMenuItem\_Click(object sender, EventArgs e)

{

Summary SummaryForm = new Summary();

SummaryForm.Show();

}

}

}

Листинг файла AdminPanel.cs

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Avia

{

public partial class AdminPanel : Form

{

public String conStr = "Data Source=ANDREYNOTE\\FOREMDB;Initial Catalog=session;Persist Security Info=True;User ID=sa;Password=12345";

private SqlConnection connection;

private SqlCommand cmd;

private DataSet DS;

private SqlDataAdapter da;

private SqlDataReader ExecuteReader;

int StartupCounter = 0;

bool ExitOK = false;

int UserID = 0;

public AdminPanel(int id)

{

UserID = id;

InitializeComponent();

UpdateDataGridUsers();

SaveExit();

}

public void UpdateDataGridUsers()

{

connection = new SqlConnection(conStr);

try

{

connection.Open();

DS = new DataSet();

SqlDataAdapter DAUserAdmin = new SqlDataAdapter("SELECT Users.FirstName AS Name, Users.LastName, DATEDIFF( year, Users.Birthdate, getDate() ) AS Age, Roles.Title AS 'User Role', Users.Email AS Email, Offices.Title AS Office, Users.Active, Users.ID FROM Users INNER JOIN Offices ON Users.OfficeID = Offices.ID INNER JOIN Roles ON Users.RoleID = Roles.ID", connection);

DAUserAdmin.Fill(DS);

UserGridAdmin\_View.DataSource = DS.Tables[0];

foreach (DataGridViewRow row in UserGridAdmin\_View.Rows)

{

if (row.Cells[6].Value.ToString() == "False") row.DefaultCellStyle.BackColor = Color.Red;

else row.DefaultCellStyle.BackColor = Color.White;

}

DataSet DSOffices = new DataSet();

SqlDataAdapter DAOffices = new SqlDataAdapter("SELECT DISTINCT Users.OfficeID, Offices.Title, Offices.ID FROM Users INNER JOIN Offices ON Users.OfficeID = Offices.ID", connection);

DAOffices.Fill(DSOffices);

DSOffices.Tables[0].Rows.Add(new object[] { null, "All offices" });

SelectOfficesAdmin\_ComboBox.DataSource = DSOffices.Tables[0];

SelectOfficesAdmin\_ComboBox.DisplayMember = "Title";

SelectOfficesAdmin\_ComboBox.ValueMember = "Title";

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

finally

{

connection.Close();

}

SelectOfficesAdmin\_ComboBox.SelectedIndex = SelectOfficesAdmin\_ComboBox.Items.Count - 1;

}

private void SelectOfficesAdmin\_ComboBox\_SelectedValueChanged(object sender, EventArgs e)

{

try

{

if (SelectOfficesAdmin\_ComboBox.SelectedValue.ToString() == "All offices") (UserGridAdmin\_View.DataSource as DataTable).DefaultView.RowFilter = null;

else (UserGridAdmin\_View.DataSource as DataTable).DefaultView.RowFilter = string.Format("Office = '{0}'", SelectOfficesAdmin\_ComboBox.SelectedValue.ToString());

if (StartupCounter < 2)

{

StartupCounter++;

(UserGridAdmin\_View.DataSource as DataTable).DefaultView.RowFilter = null;

SelectOfficesAdmin\_ComboBox.SelectedIndex = SelectOfficesAdmin\_ComboBox.Items.Count - 1;

}

}

catch { }

}

private void ChangeActivateAdmin\_Btn\_Click(object sender, EventArgs e)

{

string TempUserActive = "";

if (UserGridAdmin\_View.SelectedRows[0].Cells[6].Value.ToString() == "True") TempUserActive = "False";

else TempUserActive = "True";

connection = new SqlConnection(conStr);

try

{

connection.Open();

cmd = new SqlCommand(String.Format("UPDATE Users SET Active='{0}' WHERE Email='{1}'", TempUserActive, UserGridAdmin\_View.SelectedRows[0].Cells[4].Value), connection);

cmd.ExecuteNonQuery();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

finally

{

connection.Close();

}

UpdateDataGridUsers();

}

private void addUserToolStripMenuItem\_Click(object sender, EventArgs e)

{

AddUser AddUserForm = new AddUser();

AddUserForm.Show();

}

private void ChangeRoleAdmin\_Btn\_Click(object sender, EventArgs e)

{

ChangeUserRole ChangeUserRoleForm = new ChangeUserRole(UserGridAdmin\_View.CurrentRow.Cells[4].Value.ToString(), UserGridAdmin\_View.CurrentRow.Cells[0].Value.ToString(), UserGridAdmin\_View.CurrentRow.Cells[1].Value.ToString(), UserGridAdmin\_View.CurrentRow.Cells[5].Value.ToString(), UserGridAdmin\_View.CurrentRow.Cells[3].Value.ToString(), UserGridAdmin\_View.CurrentRow.Cells[7].Value.ToString());

ChangeUserRoleForm.Show();

}

private void UserGridAdmin\_View\_CellEnter(object sender, DataGridViewCellEventArgs e)

{

ChangeRoleAdmin\_Btn.Enabled = true;

}

private void SaveExit()

{

connection = new SqlConnection(conStr);

try

{

connection.Open();

cmd = new SqlCommand(String.Format("SELECT \* FROM Tracks WHERE UserID={0}", UserID), connection);

ExecuteReader = cmd.ExecuteReader();

while (ExecuteReader.Read())

{

if (ExecuteReader["ExitOK"].ToString() == "False" && !ExitOK)

{

ExitReason ExitReasonForm = new ExitReason(ExecuteReader["ID"].ToString(), ExecuteReader["Date"].ToString(), ExecuteReader["Login"].ToString());

ExitReasonForm.Show();

}

}

ExecuteReader.Close();

if (!ExitOK)

{

cmd = new SqlCommand(String.Format("INSERT INTO Tracks (UserID, Date, Login, Logout, ExitOK, Reason) VALUES ('{0}', '{1}', '{2}', '{3}', '{4}', '{5}')", UserID, DateTime.Now.ToShortDateString(), DateTime.Now.ToString("HH:mm:ss"), null, "False", null), connection);

cmd.ExecuteNonQuery();

ExitOK = true;

}

else

{

cmd = new SqlCommand(String.Format("UPDATE Tracks SET Logout='{0}', ExitOK='{1}' WHERE UserID={2} AND ID = (SELECT MAX(ID) FROM Users)", DateTime.Now.ToString("HH:mm:ss"), "True", UserID), connection);

cmd.ExecuteNonQuery();

Application.Exit();

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

finally

{

connection.Close();

}

}

private void exitToolStripMenuItem\_Click(object sender, EventArgs e)

{

this.Close();

}

private void scheduleToolStripMenuItem\_Click(object sender, EventArgs e)

{

SchedulePanel ShedulePanelForm = new SchedulePanel();

ShedulePanelForm.Show();

}

private void AdminPanel\_FormClosed(object sender, FormClosedEventArgs e)

{

SaveExit();

}

}

}

Листинг файла AddUser.cs

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.Linq;

using System.Security.Cryptography;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Avia

{

public partial class AddUser : Form

{

public String conStr = "Data Source=ANDREYNOTE\\FOREMDB;Initial Catalog=session;Persist Security Info=True;User ID=sa;Password=12345";

private SqlConnection connection;

private SqlCommand cmd;

public AddUser()

{

InitializeComponent();

connection = new SqlConnection(conStr);

try

{

connection.Open();

DataSet DSOffices = new DataSet();

SqlDataAdapter DAOffices = new SqlDataAdapter("SELECT \* FROM Offices", connection);

DAOffices.Fill(DSOffices);

AddUserOffice\_Combobox.DataSource = DSOffices.Tables[0];

AddUserOffice\_Combobox.DisplayMember = "Title";

AddUserOffice\_Combobox.ValueMember = "Title";

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

finally

{

connection.Close();

}

}

private void AddUserSave\_Btn\_Click(object sender, EventArgs e)

{

if(AddUserEmail\_Box.Text.Length > 0 && AddUserName\_Box.Text.Length > 0 && AddUserLastname\_Box.Text.Length > 0 && AddUserPassword\_Box.Text.Length > 0 && AddUserBirthdate\_Picker.Value < DateTime.Now)

{

byte[] hash = Encoding.ASCII.GetBytes(AddUserPassword\_Box.Text);

MD5 md5 = new MD5CryptoServiceProvider();

byte[] hashenc = md5.ComputeHash(hash);

string SecurePass = "";

foreach (var b in hashenc)

{

SecurePass += b.ToString("x2");

}

try

{

connection.Open();

cmd = new SqlCommand(String.Format("INSERT INTO Users (RoleID, Email, Password, FirstName, LastName, OfficeID, Birthdate, Active) VALUES ('{0}', '{1}', '{2}', '{3}', '{4}', '{5}', '{6}', '{7}')", 1, AddUserEmail\_Box.Text, SecurePass, AddUserName\_Box.Text, AddUserLastname\_Box.Text, 1, AddUserBirthdate\_Picker.Value.ToShortDateString(), "True"), connection);

cmd.ExecuteNonQuery();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

finally

{

connection.Close();

}

(Application.OpenForms["AdminPanel"] as AdminPanel).UpdateDataGridUsers();

this.Close();

}

else MessageBox.Show("Not all fields are filled");

}

private void AddUserCancel\_Btn\_Click(object sender, EventArgs e)

{

this.Close();

}

}

}

Листинг файла AmenitiesReport.cs

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Avia

{

public partial class AmenitiesReport : Form

{

public String conStr = "Data Source=ANDREYNOTE\\FOREMDB;Initial Catalog=session;Persist Security Info=True;User ID=sa;Password=12345";

private SqlConnection connection;

private SqlCommand cmd;

private DataSet DS;

private SqlDataAdapter da;

private SqlDataReader ExecuteReader;

public AmenitiesReport()

{

InitializeComponent();

UpdateGrid();

}

public void UpdateGrid()

{

connection = new SqlConnection(conStr);

try

{

connection.Open();

DataSet DSAmenities = new DataSet();

SqlDataAdapter DAAmenities = new SqlDataAdapter("SELECT DISTINCT Service AS 'Amenity', (SELECT COUNT(Service) FROM Amenities WHERE TicketID = ID AND CabinTypeID = 1) AS 'Economy', (SELECT COUNT(Service) FROM Amenities WHERE TicketID = ID AND CabinTypeID = 2) AS 'Business', (SELECT COUNT(Service) FROM Amenities WHERE TicketID = ID AND CabinTypeID = 3) AS 'First' FROM AmenitiesTickets INNER JOIN Amenities ON AmenitiesTickets.AmenityID = Amenities.ID INNER JOIN Tickets ON AmenitiesTickets.TicketID = Tickets.ID", connection);

DAAmenities.Fill(DSAmenities);

AmenitiesReportGrid\_View.DataSource = DSAmenities.Tables[0];

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

finally

{

connection.Close();

}

}

}

}

Листинг файла AmenitiesPurchase.cs

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Avia

{

public partial class AmenitiesPurchase : Form

{

public String conStr = "Data Source=ANDREYNOTE\\FOREMDB;Initial Catalog=session;Persist Security Info=True;User ID=sa;Password=12345";

private SqlConnection connection;

private SqlCommand cmd;

private SqlDataReader ExecuteReader;

private int TicketID = 0;

public AmenitiesPurchase()

{

InitializeComponent();

connection = new SqlConnection(conStr);

try

{

connection.Open();

cmd = new SqlCommand(String.Format("SELECT Service, convert(int, floor(Price)) AS Cost FROM Amenities"), connection);

ExecuteReader = cmd.ExecuteReader();

int TempAmenityCounter = 1;

while (ExecuteReader.Read())

{

switch (TempAmenityCounter)

{

case 1:

Amenities1\_Check.Text = ExecuteReader["Service"] + " (" + ExecuteReader["Cost"] + ")";

break;

case 2:

Amenities2\_Check.Text = ExecuteReader["Service"] + " (" + ExecuteReader["Cost"] + ")";

break;

case 3:

Amenities3\_Check.Text = ExecuteReader["Service"] + " (" + ExecuteReader["Cost"] + ")";

break;

case 4:

Amenities4\_Check.Text = ExecuteReader["Service"] + " (" + ExecuteReader["Cost"] + ")";

break;

case 5:

Amenities5\_Check.Text = ExecuteReader["Service"] + " (" + ExecuteReader["Cost"] + ")";

break;

case 6:

Amenities6\_Check.Text = ExecuteReader["Service"] + " (" + ExecuteReader["Cost"] + ")";

break;

case 7:

Amenities7\_Check.Text = ExecuteReader["Service"] + " (" + ExecuteReader["Cost"] + ")";

break;

case 8:

Amenities8\_Check.Text = ExecuteReader["Service"] + " (" + ExecuteReader["Cost"] + ")";

break;

case 9:

Amenities9\_Check.Text = ExecuteReader["Service"] + " (" + ExecuteReader["Cost"] + ")";

break;

case 10:

Amenities10\_Check.Text = ExecuteReader["Service"] + " (" + ExecuteReader["Cost"] + ")";

break;

case 11:

Amenities11\_Check.Text = ExecuteReader["Service"] + " (" + ExecuteReader["Cost"] + ")";

break;

case 12:

Amenities12\_Check.Text = ExecuteReader["Service"] + " (" + ExecuteReader["Cost"] + ")";

break;

}

TempAmenityCounter++;

}

ExecuteReader.Close();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

finally

{

connection.Close();

}

}

private void AmenitiesExit\_Btn\_Click(object sender, EventArgs e)

{

this.Close();

}

private void AmenitiesOK\_Btn\_Click(object sender, EventArgs e)

{

if (AmenitiesReference\_Box.Text.Length > 0)

{

connection = new SqlConnection(conStr);

try

{

connection.Open();

DataSet DSTick = new DataSet();

SqlDataAdapter DATick = new SqlDataAdapter(string.Format("SELECT \* FROM Tickets INNER JOIN Schedules ON Tickets.ScheduleID = Schedules.ID WHERE BookingReference = '{0}' AND Date < '{1}'", AmenitiesReference\_Box.Text, DateTime.Now.AddDays(-1).ToString("yyyyMMdd")), connection); //Изменить дату

DATick.Fill(DSTick);

AmenitiesFlight\_ComboBox.DataSource = DSTick.Tables[0];

AmenitiesFlight\_ComboBox.DisplayMember = "Firstname";

AmenitiesFlight\_ComboBox.ValueMember = "ID";

//AmenitiesFlight\_ComboBox.Items.Add(ExecuteReader["FlightNumber"]+", "+ExecuteReader["BookingReference"] + ", " + ExecuteReader["Date"] + ", " + ExecuteReader["Time"], ExecuteReader["Time"]);

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

finally

{

connection.Close();

}

}

else MessageBox.Show("Enter reference code!");

}

private void AmenitiesShow\_Btn\_Click(object sender, EventArgs e)

{

Amenities1\_Check.Visible = false; Amenities2\_Check.Visible = false; Amenities3\_Check.Visible = false; Amenities4\_Check.Visible = false; Amenities5\_Check.Visible = false; Amenities6\_Check.Visible = false; Amenities7\_Check.Visible = false; Amenities8\_Check.Visible = false; Amenities9\_Check.Visible = false; Amenities10\_Check.Visible = false; Amenities11\_Check.Visible = false; Amenities12\_Check.Visible = false;

Amenities1\_Check.Checked = false; Amenities2\_Check.Checked = false; Amenities3\_Check.Checked = false; Amenities4\_Check.Checked = false; Amenities5\_Check.Checked = false; Amenities6\_Check.Checked = false; Amenities7\_Check.Checked = false; Amenities8\_Check.Checked = false; Amenities9\_Check.Checked = false; Amenities10\_Check.Checked = false; Amenities11\_Check.Checked = false; Amenities12\_Check.Checked = false;

string CabinTypeID = "";

if (AmenitiesFlight\_ComboBox.SelectedValue.ToString().Length > 0)

{

TicketID = Convert.ToInt32(AmenitiesFlight\_ComboBox.SelectedValue);

connection = new SqlConnection(conStr);

try

{

connection.Open();

cmd = new SqlCommand(String.Format("SELECT \* FROM Tickets INNER JOIN CabinTypes ON Tickets.CabinTypeID = CabinTypes.ID WHERE Tickets.ID={0}", AmenitiesFlight\_ComboBox.SelectedValue.ToString()), connection);

ExecuteReader = cmd.ExecuteReader();

while (ExecuteReader.Read())

{

AmenitiesFlightName\_Label.Text = "Full name: " + ExecuteReader["Firstname"] + " " + ExecuteReader["Lastname"];

AmenitiesFlightCabin\_Label.Text = "Your cabin class is: " + ExecuteReader["Name"];

AmenitiesFlightPassport\_Label.Text = "Passport number: " + ExecuteReader["PassportNumber"];

CabinTypeID = ExecuteReader["CabinTypeID"].ToString();

}

ExecuteReader.Close();

cmd = new SqlCommand(String.Format("SELECT \* FROM AmenitiesCabinType WHERE CabinTypeID={0}", CabinTypeID), connection);

ExecuteReader = cmd.ExecuteReader();

while (ExecuteReader.Read())

{

switch(ExecuteReader["AmenityID"])

{

case 1:

Amenities1\_Check.Visible = true;

break;

case 2:

Amenities2\_Check.Visible = true;

break;

case 3:

Amenities3\_Check.Visible = true;

break;

case 4:

Amenities4\_Check.Visible = true;

break;

case 5:

Amenities5\_Check.Visible = true;

break;

case 6:

Amenities6\_Check.Visible = true;

break;

case 7:

Amenities7\_Check.Visible = true;

break;

case 8:

Amenities8\_Check.Visible = true;

break;

case 9:

Amenities9\_Check.Visible = true;

break;

case 10:

Amenities10\_Check.Visible = true;

break;

case 11:

Amenities11\_Check.Visible = true;

break;

case 12:

Amenities12\_Check.Visible = true;

break;

}

}

ExecuteReader.Close();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

finally

{

connection.Close();

}

}

}

public int FullCost = 0;

private void Amenities\_Check\_CheckedChanged(object sender, EventArgs e)

{

CheckBox СheckBoxSelected = (CheckBox)sender;

int TempValue = 0;

if (СheckBoxSelected.Checked == true)

{

int.TryParse(string.Join("", СheckBoxSelected.Text.Remove(0, 9).Where(c => char.IsDigit(c))), out TempValue);

FullCost += TempValue;

}

else

{

int.TryParse(string.Join("", СheckBoxSelected.Text.Remove(0, 9).Where(c => char.IsDigit(c))), out TempValue);

FullCost -= TempValue;

}

AmenitiesSelected\_Label.Text = "Items selected: " + FullCost;

AmenitiesDuty\_Label.Text = "Duties and taxes: " + FullCost\*0.05;

AmenitiesFull\_Label.Text = "Total payable: " + (FullCost+FullCost \* 0.05);

}

private void AmenitiesSave\_Btn\_Click(object sender, EventArgs e)

{

try

{

connection.Open();

int TempValue = 0;

if (Amenities1\_Check.Checked) {

TempValue = 0; int.TryParse(string.Join("", Amenities1\_Check.Text.Remove(0, 9).Where(c => char.IsDigit(c))), out TempValue); cmd = new SqlCommand(String.Format("INSERT INTO AmenitiesTickets (AmenityID, TicketID, Price) VALUES ((SELECT ID FROM Amenities WHERE Service LIKE '%' + '{0}' + '%'), '{1}', '{2}')", Amenities1\_Check.Text.Remove(8, Amenities1\_Check.Text.Length - 8), TicketID, TempValue), connection); cmd.ExecuteNonQuery();

}

if (Amenities2\_Check.Checked) {

TempValue = 0; int.TryParse(string.Join("", Amenities2\_Check.Text.Remove(0, 9).Where(c => char.IsDigit(c))), out TempValue); cmd = new SqlCommand(String.Format("INSERT INTO AmenitiesTickets (AmenityID, TicketID, Price) VALUES ((SELECT ID FROM Amenities WHERE Service LIKE '%' + '{0}' + '%'), '{1}', '{2}')", Amenities2\_Check.Text.Remove(8, Amenities2\_Check.Text.Length - 8), TicketID, TempValue), connection); cmd.ExecuteNonQuery();

}

if (Amenities3\_Check.Checked) {

TempValue = 0; int.TryParse(string.Join("", Amenities3\_Check.Text.Remove(0, 9).Where(c => char.IsDigit(c))), out TempValue); cmd = new SqlCommand(String.Format("INSERT INTO AmenitiesTickets (AmenityID, TicketID, Price) VALUES ((SELECT ID FROM Amenities WHERE Service LIKE '%' + '{0}' + '%'), '{1}', '{2}')", Amenities3\_Check.Text.Remove(8, Amenities3\_Check.Text.Length - 8), TicketID, TempValue), connection); cmd.ExecuteNonQuery();

}

if (Amenities4\_Check.Checked) {

TempValue = 0; int.TryParse(string.Join("", Amenities4\_Check.Text.Remove(0, 9).Where(c => char.IsDigit(c))), out TempValue); cmd = new SqlCommand(String.Format("INSERT INTO AmenitiesTickets (AmenityID, TicketID, Price) VALUES ((SELECT ID FROM Amenities WHERE Service LIKE '%' + '{0}' + '%'), '{1}', '{2}')", Amenities4\_Check.Text.Remove(8, Amenities4\_Check.Text.Length - 8), TicketID, TempValue), connection); cmd.ExecuteNonQuery();

}

if (Amenities5\_Check.Checked) {

TempValue = 0; int.TryParse(string.Join("", Amenities5\_Check.Text.Remove(0, 9).Where(c => char.IsDigit(c))), out TempValue); cmd = new SqlCommand(String.Format("INSERT INTO AmenitiesTickets (AmenityID, TicketID, Price) VALUES ((SELECT ID FROM Amenities WHERE Service LIKE '%' + '{0}' + '%'), '{1}', '{2}')", Amenities5\_Check.Text.Remove(8, Amenities5\_Check.Text.Length - 8), TicketID, TempValue), connection); cmd.ExecuteNonQuery();

}

if (Amenities6\_Check.Checked) {

TempValue = 0; int.TryParse(string.Join("", Amenities6\_Check.Text.Remove(0, 9).Where(c => char.IsDigit(c))), out TempValue); cmd = new SqlCommand(String.Format("INSERT INTO AmenitiesTickets (AmenityID, TicketID, Price) VALUES ((SELECT ID FROM Amenities WHERE Service LIKE '%' + '{0}' + '%'), '{1}', '{2}')", Amenities6\_Check.Text.Remove(8, Amenities6\_Check.Text.Length - 8), TicketID, TempValue), connection); cmd.ExecuteNonQuery();

}

if (Amenities7\_Check.Checked) {

TempValue = 0; int.TryParse(string.Join("", Amenities7\_Check.Text.Remove(0, 9).Where(c => char.IsDigit(c))), out TempValue); cmd = new SqlCommand(String.Format("INSERT INTO AmenitiesTickets (AmenityID, TicketID, Price) VALUES ((SELECT ID FROM Amenities WHERE Service LIKE '%' + '{0}' + '%'), '{1}', '{2}')", Amenities7\_Check.Text.Remove(8, Amenities7\_Check.Text.Length - 8), TicketID, TempValue), connection); cmd.ExecuteNonQuery();

}

if (Amenities8\_Check.Checked) {

TempValue = 0; int.TryParse(string.Join("", Amenities8\_Check.Text.Remove(0, 9).Where(c => char.IsDigit(c))), out TempValue); cmd = new SqlCommand(String.Format("INSERT INTO AmenitiesTickets (AmenityID, TicketID, Price) VALUES ((SELECT ID FROM Amenities WHERE Service LIKE '%' + '{0}' + '%'), '{1}', '{2}')", Amenities8\_Check.Text.Remove(8, Amenities8\_Check.Text.Length - 8), TicketID, TempValue), connection); cmd.ExecuteNonQuery();

}

if (Amenities9\_Check.Checked) {

TempValue = 0; int.TryParse(string.Join("", Amenities9\_Check.Text.Remove(0, 9).Where(c => char.IsDigit(c))), out TempValue); cmd = new SqlCommand(String.Format("INSERT INTO AmenitiesTickets (AmenityID, TicketID, Price) VALUES ((SELECT ID FROM Amenities WHERE Service LIKE '%' + '{0}' + '%'), '{1}', '{2}')", Amenities9\_Check.Text.Remove(8, Amenities9\_Check.Text.Length - 8), TicketID, TempValue), connection); cmd.ExecuteNonQuery();

}

if (Amenities10\_Check.Checked) {

TempValue = 0; int.TryParse(string.Join("", Amenities10\_Check.Text.Remove(0, 9).Where(c => char.IsDigit(c))), out TempValue); cmd = new SqlCommand(String.Format("INSERT INTO AmenitiesTickets (AmenityID, TicketID, Price) VALUES ((SELECT ID FROM Amenities WHERE Service LIKE '%' + '{0}' + '%'), '{1}', '{2}')", Amenities10\_Check.Text.Remove(8, Amenities10\_Check.Text.Length - 8), TicketID, TempValue), connection); cmd.ExecuteNonQuery();

}

if (Amenities11\_Check.Checked) {

TempValue = 0; int.TryParse(string.Join("", Amenities11\_Check.Text.Remove(0, 9).Where(c => char.IsDigit(c))), out TempValue); cmd = new SqlCommand(String.Format("INSERT INTO AmenitiesTickets (AmenityID, TicketID, Price) VALUES ((SELECT ID FROM Amenities WHERE Service LIKE '%' + '{0}' + '%'), '{1}', '{2}')", Amenities11\_Check.Text.Remove(8, Amenities11\_Check.Text.Length - 8), TicketID, TempValue), connection); cmd.ExecuteNonQuery();

}

if (Amenities12\_Check.Checked) {

TempValue = 0; int.TryParse(string.Join("", Amenities12\_Check.Text.Remove(0, 9).Where(c => char.IsDigit(c))), out TempValue); cmd = new SqlCommand(String.Format("INSERT INTO AmenitiesTickets (AmenityID, TicketID, Price) VALUES ((SELECT ID FROM Amenities WHERE Service LIKE '%' + '{0}' + '%'), '{1}', '{2}')", Amenities11\_Check.Text.Remove(8, Amenities12\_Check.Text.Length - 8), TicketID, TempValue), connection); cmd.ExecuteNonQuery();

}

MessageBox.Show("Success!");

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

finally

{

connection.Close();

}

}

}

}

Листинг файла BillConfirm.cs

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 Avia

{

public partial class BillConfirm : Form

{

public BillConfirm(int FullCost)

{

InitializeComponent();

BillingTotal\_Label.Text = "Total amount: " + FullCost.ToString();

}

private void BillConfirm\_Btn\_Click(object sender, EventArgs e)

{

(Application.OpenForms["BookingFlight"] as BookingFlight).ConfirmBill();

this.Close();

}

private void BillCancel\_Btn\_Click(object sender, EventArgs e)

{

this.Close();

}

}

}

Листинг файла BookingFlight.cs

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Avia

{

public partial class BookingFlight : Form

{

public String conStr = "Data Source=ANDREYNOTE\\FOREMDB;Initial Catalog=session;Persist Security Info=True;User ID=sa;Password=12345";

private SqlConnection connection;

private SqlCommand cmd;

private string FromPass = "";

private string ToPass = "";

private int CabinPass = 1;

private bool ReturningPass = false;

private int OutboundCostPass = 0;

private int ReturnCostPass = 0;

private string ScheduleOutboundID = "";

private string ScheduleReturnID = "";

private int UserID = 0;

public BookingFlight(string From, string To, int Cabin, string OutboundDate, string ReturnDate, string OutboundNumber, string ReturnNumber, bool Returning, int OutboundCost, int ReturnCost, string SchID, string ReturnSchID, int UID)

{

InitializeComponent();

FromPass = From;

ToPass = To;

CabinPass = Cabin;

OutboundCostPass = OutboundCost;

ReturnCostPass = ReturnCost;

ScheduleOutboundID = SchID;

ScheduleReturnID = ReturnSchID;

UserID = UID;

connection = new SqlConnection(conStr);

try

{

connection.Open();

DataSet DSCountry = new DataSet();

SqlDataAdapter DACountry = new SqlDataAdapter("SELECT \* FROM Countries", connection);

DACountry.Fill(DSCountry);

BookingPassengerPassportCountry\_Combobox.DataSource = DSCountry.Tables[0];

BookingPassengerPassportCountry\_Combobox.DisplayMember = "Name";

BookingPassengerPassportCountry\_Combobox.ValueMember = "ID";

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

finally

{

connection.Close();

}

string CabinType = "";

switch(Cabin)

{

case 1:

CabinType = "Economy";

break;

case 2:

CabinType = "Business";

break;

case 3:

CabinType = "First Class";

break;

}

BookingOutboundFrom\_Label.Text = "From: "+From;

BookingOutboundTo\_Label.Text = "To: "+To;

BookingOutboundCabin\_Label.Text = "Cabin Type: "+CabinType;

BookingOutboundDate\_Label.Text = "Date: "+OutboundDate;

BookingOutboundFlight\_Label.Text = "Flight number: "+OutboundNumber;

if(Returning && ReturnDate.Length > 0)

{

ReturningPass = true;

BookingReturnFrom\_Label.Text = "From: " + To;

BookingReturnTo\_Label.Text = "To: " + From;

BookingReturnCabin\_Label.Text = "Cabin Type: " + CabinType;

BookingReturnDate\_Label.Text = "Date: " + ReturnDate;

BookingReturnFlight\_Label.Text = "Flight number: "+ ReturnNumber;

}

}

private void BookingCancel\_Btn\_Click(object sender, EventArgs e)

{

this.Close();

}

private void BookingPassengerAdd\_Btn\_Click(object sender, EventArgs e)

{

if (BookingPassengerFirstname\_Box.Text.Length > 0 && BookingPassengerLastname\_Box.Text.Length > 0 && BookingPassengerPassportNum\_Box.Text.Length > 0 && BookingPassengerPhone\_Box.Text.Length > 0)

{

BookingPassengersGrid\_View.Rows.Add(BookingPassengerFirstname\_Box.Text, BookingPassengerLastname\_Box.Text, BookingPassengerBirthdate\_Picker.Value.ToShortDateString(), BookingPassengerPassportNum\_Box.Text, BookingPassengerPassportCountry\_Combobox.SelectedValue.ToString(), BookingPassengerPhone\_Box.Text, ScheduleOutboundID, FromPass+ToPass);

if(ReturningPass) BookingPassengersGrid\_View.Rows.Add(BookingPassengerFirstname\_Box.Text, BookingPassengerLastname\_Box.Text, BookingPassengerBirthdate\_Picker.Value.ToShortDateString(), BookingPassengerPassportNum\_Box.Text, BookingPassengerPassportCountry\_Combobox.SelectedValue.ToString(), BookingPassengerPhone\_Box.Text, ScheduleReturnID, ToPass + FromPass);

}

else MessageBox.Show("Fill all fields");

}

private void BookingRemovePassenger\_Btn\_Click(object sender, EventArgs e)

{

if (BookingPassengersGrid\_View.CurrentRow != null)

{

BookingPassengersGrid\_View.Rows.Remove(BookingPassengersGrid\_View.CurrentRow);

}

}

private void BookingConfirm\_Btn\_Click(object sender, EventArgs e)

{

int FullCost = 0;

if (ReturningPass) FullCost = (BookingPassengersGrid\_View.Rows.Count / 2 \* OutboundCostPass) + (BookingPassengersGrid\_View.Rows.Count / 2 \* ReturnCostPass);

else FullCost = BookingPassengersGrid\_View.Rows.Count / 2 \* OutboundCostPass;

BillConfirm BillConfirmForm = new BillConfirm(FullCost);

BillConfirmForm.Show();

}

public void ConfirmBill()

{

connection = new SqlConnection(conStr);

try

{

connection.Open();

foreach (DataGridViewRow Row in BookingPassengersGrid\_View.Rows)

{

cmd = new SqlCommand(String.Format("INSERT INTO Tickets (UserID, ScheduleID, CabinTypeID, Firstname, Lastname, Email, Phone, PassportNumber, PassportCountryID, BookingReference, Confirmed) VALUES ('{0}', '{1}', '{2}', '{3}', '{4}', '{5}', '{6}', '{7}','{8}', '{9}', '{10}')", UserID, Row.Cells[6].Value.ToString(), CabinPass, Row.Cells[0].Value.ToString(), Row.Cells[1].Value.ToString(), null, Row.Cells[5].Value.ToString(), Row.Cells[3].Value.ToString(), Row.Cells[4].Value.ToString(), Row.Cells[7].Value.ToString(), "True"), connection);

cmd.ExecuteNonQuery();

}

MessageBox.Show("Success");

this.Close();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

finally

{

connection.Close();

}

}

}

}

Листинг файла ChangeUserRole.cs

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Avia

{

public partial class ChangeUserRole : Form

{

public String conStr = "Data Source=ANDREYNOTE\\FOREMDB;Initial Catalog=session;Persist Security Info=True;User ID=sa;Password=12345";

private SqlConnection connection;

private SqlCommand cmd;

int TempRole = 0;

int UserID = 0;

public ChangeUserRole(string Email, string FirstName, string LastName, string Office, string Role, string id)

{

InitializeComponent();

connection = new SqlConnection(conStr);

try

{

connection.Open();

DataSet DSOffices = new DataSet();

SqlDataAdapter DAOffices = new SqlDataAdapter("SELECT \* FROM Offices", connection);

DAOffices.Fill(DSOffices);

UpdateUserOffice\_Combobox.DataSource = DSOffices.Tables[0];

UpdateUserOffice\_Combobox.DisplayMember = "Title";

UpdateUserOffice\_Combobox.ValueMember = "ID";

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

finally

{

connection.Close();

}

UserID = Convert.ToInt32(id);

UpdateUserEmail\_Box.Text = Email;

UpdateUserName\_Box.Text = FirstName;

UpdateUserLastname\_Box.Text = LastName;

UpdateUserOffice\_Combobox.SelectedIndex = UpdateUserOffice\_Combobox.FindString(Office); //Доделать

if (Role == "Administrator") UpdateUserAdm\_Radio.Checked = true;

else UpdateUserUsr\_Radio.Checked = true;

}

private void UpdateUserApply\_Btn\_Click(object sender, EventArgs e)

{

if (UpdateUserEmail\_Box.Text.Length > 0 && UpdateUserName\_Box.Text.Length > 0 && UpdateUserLastname\_Box.Text.Length > 0)

{

try

{

connection.Open();

cmd = new SqlCommand(String.Format("UPDATE Users SET Email='{0}', FirstName='{1}', LastName='{2}', OfficeID='{3}', RoleID={4} WHERE ID={5}", UpdateUserEmail\_Box.Text, UpdateUserName\_Box.Text, UpdateUserLastname\_Box.Text, UpdateUserOffice\_Combobox.SelectedValue, TempRole, UserID), connection);

cmd.ExecuteNonQuery();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

finally

{

connection.Close();

}

(Application.OpenForms["AdminPanel"] as AdminPanel).UpdateDataGridUsers();

this.Close();

}

else MessageBox.Show("Not all fields are filled");

}

private void UpdateUserCancel\_Btn\_Click(object sender, EventArgs e)

{

this.Close();

}

private void UpdateUserUsr\_Radio\_CheckedChanged(object sender, EventArgs e)

{

RadioButton radioButton = (RadioButton)sender;

if (radioButton.Checked) TempRole = 2;

}

private void UpdateUserAdm\_Radio\_CheckedChanged(object sender, EventArgs e)

{

RadioButton radioButton = (RadioButton)sender;

if (radioButton.Checked) TempRole = 1;

}

}

}

Листинг файла ExitReason.cs

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Avia

{

public partial class ExitReason : Form

{

public String conStr = "Data Source=ANDREYNOTE\\FOREMDB;Initial Catalog=session;Persist Security Info=True;User ID=sa;Password=12345";

private SqlConnection connection;

private SqlCommand cmd;

int ProblemID = 0;

string TempReason = "";

const string MainText = "No logout detected for you last login on: ";

public ExitReason(string ID, string Date, string Login)

{

ProblemID = Convert.ToInt32(ID);

InitializeComponent();

NoLogoutText\_Label.Text = MainText + Date + Login;

}

private void NoLogoutConfirm\_Btn\_Click(object sender, EventArgs e)

{

if (NoLogoutReason\_Box.Text.Length > 0)

{

connection = new SqlConnection(conStr);

try

{

connection.Open();

cmd = new SqlCommand(String.Format("UPDATE Tracks SET ExitOK='{0}', Reason='{1}' WHERE ID={2}", "True", TempReason + NoLogoutReason\_Box.Text, ProblemID), connection);

cmd.ExecuteNonQuery();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

finally

{

connection.Close();

}

this.Close();

}

else MessageBox.Show("Not all fields are filled");

}

private void NoLogoutSoft\_Radio\_CheckedChanged(object sender, EventArgs e)

{

TempReason = "Software : ";

}

private void NoLogoutHard\_Radio\_CheckedChanged(object sender, EventArgs e)

{

TempReason = "Hardware : ";

}

}

}

Листинг файла ImportSchedule.cs

using CsvHelper;

using Microsoft.VisualBasic.FileIO;

using System;

using System.Collections;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.IO;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Avia

{

public partial class ImportSchedule : Form

{

public String conStr = "Data Source=ANDREYNOTE\\FOREMDB;Initial Catalog=session;Persist Security Info=True;User ID=sa;Password=12345";

private SqlConnection connection;

private SqlCommand cmd;

public ImportSchedule()

{

InitializeComponent();

}

private void ImportScheduleImport\_Btn\_Click(object sender, EventArgs e)

{

var dialog = new OpenFileDialog();

dialog.Filter = "schedule Files|\*.csv";

dialog.Title = "Select schedule files";

if (dialog.ShowDialog() == DialogResult.OK) {

if (dialog.FileName != null)

{

ImportSchedulePath\_Box.Text = dialog.FileName;

using (TextFieldParser parser = new TextFieldParser(dialog.FileName))

{

parser.TextFieldType = FieldType.Delimited;

parser.SetDelimiters(",");

while (!parser.EndOfData)

{

//Processing row

string[] fields = parser.ReadFields();

foreach (string field in fields)

{

MessageBox.Show(field);

}

}

}

}

}

}

}

}

Листинг файла QuestionsDetails.cs

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Avia

{

public partial class QuestionsDetails : Form

{

public String conStr = "Data Source=ANDREYNOTE\\FOREMDB;Initial Catalog=session;Persist Security Info=True;User ID=sa;Password=12345";

private SqlConnection connection;

private SqlCommand cmd;

private SqlDataReader ExecuteReader;

private string[] QuestionList = { "Please rate our aircraft flown on AMONIC Airlines", "How would you rate our flight attendants", "How would you rate our inflight entertainment", "Please rate the ticket price for the trip you are taking" };

private string[] MarkList = { "Outstanding", "Very Good", "Good", "Adequate", "Needs Improvement", "Poor", "Don’t know" };

private string[] QueryList = {

"Gender = 'M'",

"Gender = 'F'",

"Age > 18 AND Age < 24",

"Age > 25 AND Age < 39",

"Age > 40 AND Age < 59",

"Age < 60",

"CabinType = 'Economy'",

"CabinType = 'Business'",

"CabinType = 'First'",

"Arrival = 'AUH'",

"Arrival = 'BAH'",

"Arrival = 'DOH'",

"Arrival = 'RUH'",

"Arrival = 'CAI'"

};

private string QuerySearchDate = "";

public QuestionsDetails()

{

InitializeComponent();

connection = new SqlConnection(conStr);

try

{

connection.Open();

DataSet DSDate = new DataSet();

SqlDataAdapter DADate = new SqlDataAdapter("SELECT DISTINCT Date FROM Questions ORDER BY Date DESC", connection);

DADate.Fill(DSDate);

DSDate.Tables[0].Rows.Add(new DateTime());

QuestionsDetailsDate\_Combo.DataSource = DSDate.Tables[0];

QuestionsDetailsDate\_Combo.DisplayMember = "Date";

QuestionsDetailsDate\_Combo.ValueMember = "Date";

QuestionsDetailsDate\_Combo.SelectedIndex = QuestionsDetailsDate\_Combo.Items.Count - 1;

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

finally

{

connection.Close();

}

ReCountGrid();

}

public void ReCountGrid()

{

int NumberOfQuestion = 1;

int RowsCounter = 0;

QuestionsGrid\_View.Rows.Clear();

connection = new SqlConnection(conStr);

try

{

connection.Open();

for (int i = 0; i < 4; i++)

{

QuestionsGrid\_View.Rows.Add(new object[] { QuestionList[i] });

QuestionsGrid\_View.Rows[RowsCounter].DefaultCellStyle.BackColor = Color.Yellow;

RowsCounter++;

for (int j = 0; j < 7; j++)

{

QuestionsGrid\_View.Rows.Add(MarkList[j]);

for (int k = 0; k < 14; k++)

{

cmd = new SqlCommand(string.Format("SELECT Count(\*) FROM Questions WHERE Q{0} = {1} AND {2} {3}", NumberOfQuestion, j + 1, QueryList[k], QuerySearchDate), connection);

ExecuteReader = cmd.ExecuteReader();

while (ExecuteReader.Read())

{

QuestionsGrid\_View.Rows[RowsCounter].Cells[k + 2].Value = ExecuteReader[""].ToString();

}

ExecuteReader.Close();

}

QuestionsGrid\_View.Rows[RowsCounter].Cells[1].Value = Convert.ToInt32(QuestionsGrid\_View.Rows[RowsCounter].Cells[2].Value) + Convert.ToInt32(QuestionsGrid\_View.Rows[RowsCounter].Cells[3].Value) + Convert.ToInt32(QuestionsGrid\_View.Rows[RowsCounter].Cells[4].Value) + Convert.ToInt32(QuestionsGrid\_View.Rows[RowsCounter].Cells[5].Value) + Convert.ToInt32(QuestionsGrid\_View.Rows[RowsCounter].Cells[6].Value) + Convert.ToInt32(QuestionsGrid\_View.Rows[RowsCounter].Cells[7].Value) + Convert.ToInt32(QuestionsGrid\_View.Rows[RowsCounter].Cells[8].Value) + Convert.ToInt32(QuestionsGrid\_View.Rows[RowsCounter].Cells[9].Value) + Convert.ToInt32(QuestionsGrid\_View.Rows[RowsCounter].Cells[10].Value) + Convert.ToInt32(QuestionsGrid\_View.Rows[RowsCounter].Cells[11].Value) + Convert.ToInt32(QuestionsGrid\_View.Rows[RowsCounter].Cells[12].Value) + Convert.ToInt32(QuestionsGrid\_View.Rows[RowsCounter].Cells[13].Value) + Convert.ToInt32(QuestionsGrid\_View.Rows[RowsCounter].Cells[14].Value) + Convert.ToInt32(QuestionsGrid\_View.Rows[RowsCounter].Cells[15].Value);

RowsCounter++;

}

NumberOfQuestion++;

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

finally

{

connection.Close();

}

}

private void exitToolStripMenuItem\_Click(object sender, EventArgs e)

{

this.Close();

}

private void QuestionsDetailsApply\_Btn\_Click(object sender, EventArgs e)

{

if (QuestionsDetailsDate\_Combo.SelectedIndex != QuestionsDetailsDate\_Combo.Items.Count - 1)

{

DateTime LowEdge = Convert.ToDateTime(QuestionsDetailsDate\_Combo.SelectedValue);

DateTime HighEdge = Convert.ToDateTime(QuestionsDetailsDate\_Combo.SelectedValue).AddMonths(1);

QuerySearchDate = "AND DATE >= '" + LowEdge.AddDays(-LowEdge.Day + 1).ToShortDateString() + "' AND DATE < '" + HighEdge.AddDays(-LowEdge.Day + 1).ToShortDateString()+"'";

//MessageBox.Show(QuerySearchDate);

}

else QuerySearchDate = "";

ReCountGrid();

}

}

}

Листинг файла Login.cs

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.Linq;

using System.Security.Cryptography;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Avia

{

public partial class Login : Form

{

public String conStr = "Data Source=ANDREYNOTE\\FOREMDB;Initial Catalog=session;Persist Security Info=True;User ID=sa;Password=12345";

private SqlConnection connection;

private SqlCommand cmd;

private SqlDataReader ExecuteReader;

public int EnterCounter = 0;

Timer Tm = null;

public int StartValue = 0;

public Login()

{

InitializeComponent();

Tm = new Timer();

Tm.Tick += new EventHandler(Tm\_Tick);

Tm.Interval = 1000;

//AdminPanel AdminForm = new AdminPanel(1);

//AdminForm.Show(); //test

//UserPanel UserForm = new UserPanel(1, "Henry");

//UserForm.Show(); //test

//SchedulePanel ShedulePanelForm = new SchedulePanel();

//ShedulePanelForm.Show(); //test

//SearchFlight SearchFlightForm = new SearchFlight(1);

//SearchFlightForm.Show(); //test

//BookingFlight BookingFlight = new BookingFlight("ASD","DSA",1,"1","2","3","4", true, 100,200,"5","6",1);

//BookingFlight.Show(); //test

//QuestionsSummary QuestionsSummary = new QuestionsSummary();

//QuestionsSummary.Show(); //test

//QuestionsDetails QuestionsDetailsForm = new QuestionsDetails();

//QuestionsDetailsForm.Show();

//AmenitiesPurchase AmenitiesPurchase = new AmenitiesPurchase();

//AmenitiesPurchase.Show();

//AmenitiesReport AmenitiesReportForm = new AmenitiesReport();

//AmenitiesReportForm.Show();

//Summary Summary = new Summary();

//Summary.Show();

}

private void Login\_Btn\_Click(object sender, EventArgs e)

{

if (PasswordEnter\_Box.Text.Length > 0 && LoginEnter\_Box.Text.Length > 0)

{

byte[] hash = Encoding.ASCII.GetBytes(PasswordEnter\_Box.Text);

MD5 md5 = new MD5CryptoServiceProvider();

byte[] hashenc = md5.ComputeHash(hash);

string SecurePass = "";

foreach (var b in hashenc)

{

SecurePass += b.ToString("x2");

}

//MessageBox.Show(SecurePass);

connection = new SqlConnection(conStr);

try

{

connection.Open();

cmd = new SqlCommand(String.Format("SELECT \* FROM Users WHERE Email='{0}'", LoginEnter\_Box.Text), connection);

ExecuteReader = cmd.ExecuteReader();

while (ExecuteReader.Read())

{

if (ExecuteReader["Active"].ToString() != "False")

{

if (ExecuteReader["Password"].ToString() == SecurePass)

{

if (ExecuteReader["RoleID"].ToString() == "1")

{

AdminPanel AdminForm = new AdminPanel(Convert.ToInt32(ExecuteReader["ID"]));

AdminForm.Show();

this.Hide();

}

else

{

UserPanel UserForm = new UserPanel(Convert.ToInt32(ExecuteReader["ID"]), ExecuteReader["FirstName"].ToString());

UserForm.Show();

this.Hide();

}

}

else {

EnterCounter++;

MessageBox.Show("Incorrect login or password");

if(EnterCounter == 3)

{

Login\_Btn.Enabled = false;

StartValue = 10;

Tm.Start();

}

}

}

else MessageBox.Show("Account was deactivated");

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

finally

{

connection.Close();

}

}

else MessageBox.Show("Not all fields are filled");

}

void Tm\_Tick(object sender, EventArgs e)

{

if (StartValue != 0)

{

Login\_Btn.Text = (StartValue - (StartValue - (StartValue % (60 \* 60))) / (60 \* 60) \* 60 \* 60 - (StartValue - StartValue % 60) / 60 - (StartValue - (StartValue % (60 \* 60))) / (60 \* 60) \* 60 \* 60).ToString();

StartValue--;

}

else

{

(sender as Timer).Stop();

(sender as Timer).Dispose();

Login\_Btn.Text = "Login";

Login\_Btn.Enabled = true;

EnterCounter = 0;

}

}

private void ExitLogin\_Btn\_Click(object sender, EventArgs e)

{

Application.Exit();

}

}

}

Листинг файла QuestionsSummary.cs

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Avia

{

public partial class QuestionsSummary : Form

{

public String conStr = "Data Source=ANDREYNOTE\\FOREMDB;Initial Catalog=session;Persist Security Info=True;User ID=sa;Password=12345";

private SqlConnection connection;

private SqlCommand cmd;

private SqlDataReader ExecuteReader;

private int MaleCounter = 0;

private int FemaleCounter = 0;

private int AgeACounter = 0;

private int AgeBCounter = 0;

private int AgeCCounter = 0;

private int AgeDCounter = 0;

private int EconomyCounter = 0;

private int BusinessCounter = 0;

private int FirstCounter = 0;

private int AirACounter = 0;

private int AirBCounter = 0;

private int AirCCounter = 0;

private int AirDCounter = 0;

private int AirECounter = 0;

private DateTime MinInterval;

private DateTime MaxInterval;

private int Counter = 0;

public QuestionsSummary()

{

InitializeComponent();

UpdateSummary();

}

private void importResultsToolStripMenuItem\_Click(object sender, EventArgs e)

{

QuestionsImport QuestionsImportForm = new QuestionsImport();

QuestionsImportForm.Show();

}

public void UpdateSummary()

{

connection = new SqlConnection(conStr);

try

{

connection.Open();

cmd = new SqlCommand("SELECT \* FROM Questions", connection);

ExecuteReader = cmd.ExecuteReader();

while (ExecuteReader.Read())

{

if (Counter == 0) MinInterval = Convert.ToDateTime(ExecuteReader["Date"]);

else if (Convert.ToDateTime(ExecuteReader["Date"]) < MinInterval) MinInterval = Convert.ToDateTime(ExecuteReader["Date"]);

if (Convert.ToDateTime(ExecuteReader["Date"]) > MaxInterval) MaxInterval = Convert.ToDateTime(ExecuteReader["Date"]);

Counter++;

if (ExecuteReader["Gender"].ToString() == "M") MaleCounter++;

else FemaleCounter++;

switch (ExecuteReader["CabinType"].ToString())

{

case "Economy":

EconomyCounter++;

break;

case "Business":

BusinessCounter++;

break;

case "First":

FirstCounter++;

break;

}

switch (ExecuteReader["Arrival"].ToString())

{

case "AUH":

AirACounter++;

break;

case "BAH":

AirBCounter++;

break;

case "DOH":

AirCCounter++;

break;

case "RUH":

AirDCounter++;

break;

case "CAI":

AirECounter++;

break;

}

if (Convert.ToInt32(ExecuteReader["Age"]) >= 18 && Convert.ToInt32(ExecuteReader["Age"]) <= 24) AgeACounter++;

else if (Convert.ToInt32(ExecuteReader["Age"]) >= 25 && Convert.ToInt32(ExecuteReader["Age"]) <= 39) AgeBCounter++;

else if (Convert.ToInt32(ExecuteReader["Age"]) >= 40 && Convert.ToInt32(ExecuteReader["Age"]) <= 59) AgeCCounter++;

else AgeDCounter++;

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

finally

{

connection.Close();

}

MaleCounter\_Label.Text = MaleCounter.ToString();

FemaleCounter\_Label.Text = FemaleCounter.ToString();

AgeACounter\_Label.Text = AgeACounter.ToString();

AgeBCounter\_Label.Text = AgeBCounter.ToString();

AgeCCounter\_Label.Text = AgeCCounter.ToString();

AgeDCounter\_Label.Text = AgeDCounter.ToString();

EconomyCounter\_Label.Text = EconomyCounter.ToString();

BusinessCounter\_Label.Text = BusinessCounter.ToString();

FirstCounter\_Label.Text = FirstCounter.ToString();

AirACounter\_Label.Text = AirACounter.ToString();

AirBCounter\_Label.Text = AirBCounter.ToString();

AirCCounter\_Label.Text = AirCCounter.ToString();

AirDCounter\_Label.Text = AirDCounter.ToString();

AirECounter\_Label.Text = AirECounter.ToString();

QuestionsSummarySize\_Label.Text = "Sample size: " + Counter.ToString();

QuestionsSummaryInterval\_Label.Text = "Fieldwork: " + MinInterval.ToString("MMMM-yyyy") + " - " + MaxInterval.ToString("MMMM-yyyy");

}

private void viewSummaryToolStripMenuItem\_Click(object sender, EventArgs e)

{

QuestionsDetails QuestionsDetailsForm = new QuestionsDetails();

QuestionsDetailsForm.Show();

}

}

}

Листинг файла QuestionsImport.cs

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.IO;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Avia

{

public partial class QuestionsImport : Form

{

public String conStr = "Data Source=ANDREYNOTE\\FOREMDB;Initial Catalog=session;Persist Security Info=True;User ID=sa;Password=12345";

private SqlConnection connection;

private SqlCommand cmd;

private bool FirstIterate = false;

private string[] Questions;

public QuestionsImport()

{

InitializeComponent();

}

private void ImportQuestionsOpen\_Btn\_Click(object sender, EventArgs e)

{

var dialog = new OpenFileDialog();

dialog.Filter = "schedule Files|\*.csv";

dialog.Title = "Select schedule files";

if (dialog.ShowDialog() == DialogResult.OK)

{

if (dialog.FileName != null)

{

ImportQuestionsPath\_Box.Text = dialog.FileName;

Questions = File.ReadAllLines(dialog.FileName);

}

}

}

private void ImportQuestionsImport\_Btn\_Click(object sender, EventArgs e)

{

if(Questions.Length > 0)

{

connection = new SqlConnection(conStr);

try

{

connection.Open();

foreach (string Row in Questions)

{

if (FirstIterate)

{

string[] ParsedRow = Row.Split(',');

cmd = new SqlCommand(String.Format("INSERT INTO Questions (Date, Departure, Arrival, Age, Gender, CabinType, Q1, Q2, Q3, Q4) VALUES ('{0}', '{1}', '{2}', '{3}', '{4}', '{5}', '{6}', '{7}', '{8}', '{9}')", ImportQuestionsDate\_Picker.Value.ToString(), ParsedRow[0], ParsedRow[1], ParsedRow[2], ParsedRow[3], ParsedRow[4], ParsedRow[5], ParsedRow[6], ParsedRow[7], ParsedRow[8]), connection);

cmd.ExecuteNonQuery();

}

FirstIterate = true;

}

MessageBox.Show("Success!");

(Application.OpenForms["QuestionsSummary"] as QuestionsSummary).UpdateSummary();

this.Close();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

finally

{

connection.Close();

}

}

else MessageBox.Show("Read file error!");

}

}

}

Листинг файла ScheduleEdit.cs

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Avia

{

public partial class ScheduleEdit : Form

{

public String conStr = "Data Source=ANDREYNOTE\\FOREMDB;Initial Catalog=session;Persist Security Info=True;User ID=sa;Password=12345";

private SqlConnection connection;

private SqlCommand cmd;

int UserID = 0;

public ScheduleEdit(string id, string from, string to, string aircraft, string date, string time, string price)

{

InitializeComponent();

UserID = Convert.ToInt32(id);

UpdateScheduleFrom\_Label.Text = "From: " + from;

UpdateScheduleTo\_Label.Text = "To: " + to;

UpdateScheduleAircraft\_Label.Text = "Aircraft: " + aircraft;

UpdateScheduleDate\_Picker.Value = Convert.ToDateTime(date);

UpdateScheduleTime\_Picker.Value = Convert.ToDateTime(time);

UpdateSchedulePrice\_Box.Text = price;

}

private void UpdateScheduleCancel\_Btn\_Click(object sender, EventArgs e)

{

this.Close();

}

private void UpdateScheduleApply\_Btn\_Click(object sender, EventArgs e)

{

if (UpdateSchedulePrice\_Box.Text.Length > 0)

{

connection = new SqlConnection(conStr);

try

{

connection.Open();

cmd = new SqlCommand(String.Format("UPDATE Schedules SET Date='{0}', Time='{1}', EconomyPrice='{2}' WHERE ID={3}", UpdateScheduleDate\_Picker.Value.ToShortDateString(), UpdateScheduleTime\_Picker.Value.ToShortTimeString(), UpdateSchedulePrice\_Box.Text, UserID), connection);

cmd.ExecuteNonQuery();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

finally

{

connection.Close();

}

(Application.OpenForms["SchedulePanel"] as SchedulePanel).UpdateDataGridSchedules();

this.Close();

}

else MessageBox.Show("Not all fields are filled");

}

}

}

Листинг файла SchedulePanel.cs

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Avia

{

public partial class SchedulePanel : Form

{

public String conStr = "Data Source=ANDREYNOTE\\FOREMDB;Initial Catalog=session;Persist Security Info=True;User ID=sa;Password=12345";

private SqlConnection connection;

private SqlCommand cmd;

private DataSet DS;

private SqlDataAdapter da;

private SqlDataReader ExecuteReader;

private bool DateFlightChanged = false;

public SchedulePanel()

{

InitializeComponent();

UpdateDataGridSchedules();

}

public void UpdateDataGridSchedules()

{

connection = new SqlConnection(conStr);

try

{

connection.Open();

DS = new DataSet();

SqlDataAdapter DAShedule = new SqlDataAdapter("SELECT Schedules.Date AS 'Date', Schedules.Time AS 'Time', (SELECT IATACode FROM Airports WHERE ID = Routes.DepartureAirportID) AS 'From', (SELECT IATACode FROM Airports WHERE ID = Routes.ArrivalAirportID) AS 'To', Schedules.FlightNumber AS 'Flight number', Aircrafts.Name AS 'Aircraft', CAST(Schedules.EconomyPrice AS int) AS 'Economy price', FLOOR(Schedules.EconomyPrice+(Schedules.EconomyPrice\*0.35)) AS 'Business price', FLOOR(Schedules.EconomyPrice+(Schedules.EconomyPrice\*0.30)) AS 'First class price', Schedules.Confirmed, Schedules.ID FROM Schedules INNER JOIN Aircrafts ON Schedules.AircraftID = Aircrafts.ID INNER JOIN Routes ON Schedules.RouteID = Routes.ID", connection);

DAShedule.Fill(DS);

ScheduleGrid\_View.DataSource = DS.Tables[0];

foreach (DataGridViewRow row in ScheduleGrid\_View.Rows)

{

if (row.Cells[9].Value.ToString() == "False") row.DefaultCellStyle.BackColor = Color.Red;

else row.DefaultCellStyle.BackColor = Color.White;

}

DataSet DSFrom = new DataSet();

SqlDataAdapter DAFrom = new SqlDataAdapter("SELECT IATACode FROM Airports", connection);

DAFrom.Fill(DSFrom);

DSFrom.Tables[0].Rows.Add("");

DataSet DSTo = new DataSet();

SqlDataAdapter DATo = new SqlDataAdapter("SELECT IATACode FROM Airports", connection);

DATo.Fill(DSTo);

DSTo.Tables[0].Rows.Add("");

ScheduleFrom\_ComboBox.DataSource = DSFrom.Tables[0];

ScheduleFrom\_ComboBox.DisplayMember = "IATACode";

ScheduleFrom\_ComboBox.ValueMember = "IATACode";

ScheduleTo\_ComboBox.DataSource = DSTo.Tables[0];

ScheduleTo\_ComboBox.DisplayMember = "IATACode";

ScheduleTo\_ComboBox.ValueMember = "IATACode";

ScheduleFrom\_ComboBox.SelectedIndex = ScheduleFrom\_ComboBox.Items.Count - 1;

ScheduleTo\_ComboBox.SelectedIndex = ScheduleTo\_ComboBox.Items.Count - 1;

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

finally

{

connection.Close();

}

}

private void ScheduleSortApply\_Btn\_Click(object sender, EventArgs e)

{

int FilterCounter = 0;

if (ScheduleFrom\_ComboBox.SelectedValue.ToString().Length > 0) (ScheduleGrid\_View.DataSource as DataTable).DefaultView.RowFilter = string.Format("From = '{0}'", ScheduleFrom\_ComboBox.SelectedValue.ToString());

else FilterCounter++;

if (ScheduleTo\_ComboBox.SelectedValue.ToString().Length > 0) (ScheduleGrid\_View.DataSource as DataTable).DefaultView.RowFilter = string.Format("To = '{0}'", ScheduleTo\_ComboBox.SelectedValue.ToString());

else FilterCounter++;

if (ScheduleFlightNumber\_Box.Text.Length > 0) (ScheduleGrid\_View.DataSource as DataTable).DefaultView.RowFilter = string.Format("`Flight number` = '{0}'", ScheduleFlightNumber\_Box.Text);

else FilterCounter++;

if (DateFlightChanged) (ScheduleGrid\_View.DataSource as DataTable).DefaultView.RowFilter = string.Format("Date = '{0}'", ScheduleOutbound\_Picker.Value.ToShortDateString());

else FilterCounter++;

if (FilterCounter == 4 && !DateFlightChanged) (ScheduleGrid\_View.DataSource as DataTable).DefaultView.RowFilter = null;

switch (ScheduleSortBy\_ComboBox.SelectedIndex) {

case 0:

ScheduleGrid\_View.Sort(ScheduleGrid\_View.Columns["Date"], ListSortDirection.Descending);

break;

case 1:

ScheduleGrid\_View.Sort(ScheduleGrid\_View.Columns["Date"], ListSortDirection.Ascending);

break;

case 2:

ScheduleGrid\_View.Sort(ScheduleGrid\_View.Columns["Economy price"], ListSortDirection.Ascending);

break;

case 3:

ScheduleGrid\_View.Sort(ScheduleGrid\_View.Columns["Confirmed"], ListSortDirection.Ascending);

break;

}

DateFlightChanged = false;

}

private void ScheduleOutbound\_Picker\_ValueChanged(object sender, EventArgs e)

{

DateFlightChanged = true;

}

private void ScheduleCancelFlight\_Btn\_Click(object sender, EventArgs e)

{

string TempFlightActive = "";

if (ScheduleGrid\_View.SelectedRows[0].Cells[9].Value.ToString() == "True") TempFlightActive = "False";

else TempFlightActive = "True";

connection = new SqlConnection(conStr);

try

{

connection.Open();

cmd = new SqlCommand(String.Format("UPDATE Schedules SET Confirmed='{0}' WHERE ID='{1}'", TempFlightActive, ScheduleGrid\_View.SelectedRows[0].Cells[10].Value), connection);

cmd.ExecuteNonQuery();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

finally

{

connection.Close();

}

UpdateDataGridSchedules();

}

private void ScheduleEditFlight\_Btn\_Click(object sender, EventArgs e)

{

ScheduleEdit ScheduleEditForm = new ScheduleEdit(ScheduleGrid\_View.CurrentRow.Cells[10].Value.ToString(), ScheduleGrid\_View.CurrentRow.Cells[2].Value.ToString(), ScheduleGrid\_View.CurrentRow.Cells[3].Value.ToString(), ScheduleGrid\_View.CurrentRow.Cells[5].Value.ToString(), ScheduleGrid\_View.CurrentRow.Cells[0].Value.ToString(), ScheduleGrid\_View.CurrentRow.Cells[1].Value.ToString(), ScheduleGrid\_View.CurrentRow.Cells[6].Value.ToString());

ScheduleEditForm.Show();

}

private void ScheduleImport\_Btn\_Click(object sender, EventArgs e)

{

ImportSchedule ImportScheduleForm = new ImportSchedule();

ImportScheduleForm.Show();

}

}

}

Листинг файла SearchFlight.cs

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Avia

{

public partial class SearchFlight : Form

{

public String conStr = "Data Source=ANDREYNOTE\\FOREMDB;Initial Catalog=session;Persist Security Info=True;User ID=sa;Password=12345";

private SqlConnection connection;

private SqlCommand cmd;

private SqlDataAdapter da;

private SqlDataReader ExecuteReader;

private bool Returning = false;

private int UserID = 0;

public SearchFlight(int UID)

{

InitializeComponent();

UserID = UID;

connection = new SqlConnection(conStr);

try

{

connection.Open();

DataSet DSFrom = new DataSet();

SqlDataAdapter DAFrom = new SqlDataAdapter("SELECT IATACode FROM Airports", connection);

DAFrom.Fill(DSFrom);

DSFrom.Tables[0].Rows.Add("");

DataSet DSTo = new DataSet();

SqlDataAdapter DATo = new SqlDataAdapter("SELECT IATACode FROM Airports", connection);

DATo.Fill(DSTo);

DSTo.Tables[0].Rows.Add("");

DataSet DSCabin = new DataSet();

SqlDataAdapter DACabin = new SqlDataAdapter("SELECT \* FROM CabinTypes", connection);

DACabin.Fill(DSCabin);

SearchParametersFrom\_Combobox.DataSource = DSFrom.Tables[0];

SearchParametersFrom\_Combobox.DisplayMember = "IATACode";

SearchParametersFrom\_Combobox.ValueMember = "IATACode";

SearchParametersTo\_Combobox.DataSource = DSTo.Tables[0];

SearchParametersTo\_Combobox.DisplayMember = "IATACode";

SearchParametersTo\_Combobox.ValueMember = "IATACode";

SearchParametersCabin\_Combobox.DataSource = DSCabin.Tables[0];

SearchParametersCabin\_Combobox.DisplayMember = "Name";

SearchParametersCabin\_Combobox.ValueMember = "ID";

SearchParametersFrom\_Combobox.SelectedIndex = SearchParametersFrom\_Combobox.Items.Count - 1;

SearchParametersTo\_Combobox.SelectedIndex = SearchParametersTo\_Combobox.Items.Count - 1;

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

finally

{

connection.Close();

}

}

private void SearchCancel\_Btn\_Click(object sender, EventArgs e)

{

this.Close();

}

private void SearchParametersApply\_Btn\_Click(object sender, EventArgs e)

{

Returning = false;

int CountCabinPrice = 0;

string LowDateOutbound = "";

string HighDateOutbound = "";

string LowDateReturn = "";

string HighDateReturn = "";

switch (SearchParametersCabin\_Combobox.SelectedIndex)

{

case 0:

CountCabinPrice = 0;

break;

case 1:

CountCabinPrice = 35;

break;

case 2:

CountCabinPrice = 30;

break;

}

if(SearchOutboundThree\_Check.Checked)

{

LowDateOutbound = SearchParametersOutbound\_Picker.Value.AddDays(-3).ToShortDateString();

HighDateOutbound = SearchParametersOutbound\_Picker.Value.AddDays(3).ToShortDateString();

}

else

{

LowDateOutbound = SearchParametersOutbound\_Picker.Value.ToShortDateString();

HighDateOutbound = SearchParametersOutbound\_Picker.Value.ToShortDateString();

}

if (SearchReturnThree\_Check.Checked)

{

LowDateReturn = SearchParametersReturn\_Picker.Value.AddDays(-3).ToShortDateString();

HighDateReturn = SearchParametersReturn\_Picker.Value.AddDays(3).ToShortDateString();

}

else

{

LowDateReturn = SearchParametersReturn\_Picker.Value.ToShortDateString();

HighDateReturn = SearchParametersReturn\_Picker.Value.ToShortDateString();

}

connection = new SqlConnection(conStr);

try

{

connection.Open();

DataSet DSSearchFrom = new DataSet();

SqlDataAdapter DASearchFrom = new SqlDataAdapter(string.Format("SELECT '{0}' AS 'From', '{1}' AS 'To', Schedules.Date AS 'Date', Schedules.Time AS 'Time', Schedules.FlightNumber AS 'Flight number', FLOOR(Schedules.EconomyPrice+(Schedules.EconomyPrice/100\*{2})) AS 'Cabin price', '0' AS 'Number of stops', Schedules.ID FROM Schedules WHERE Date >= '{3}' AND Date <= '{4}' AND RouteID = (SELECT ID FROM Routes WHERE DepartureAirportID = (SELECT ID FROM Airports WHERE IATACode = '{0}') AND ArrivalAirportID = (SELECT ID FROM Airports WHERE IATACode = '{1}'))", SearchParametersFrom\_Combobox.SelectedValue, SearchParametersTo\_Combobox.SelectedValue, CountCabinPrice, LowDateOutbound, HighDateOutbound), connection);

DASearchFrom.Fill(DSSearchFrom);

SearchOutboundGrid\_View.DataSource = DSSearchFrom.Tables[0];

if (SearchParametersReturn\_Radio.Checked)

{

Returning = true;

DataSet DSSearchReturn = new DataSet();

SqlDataAdapter DASearchReturn = new SqlDataAdapter(string.Format("SELECT '{0}' AS 'From', '{1}' AS 'To', Schedules.Date AS 'Date', Schedules.Time AS 'Time', Schedules.FlightNumber AS 'Flight number', FLOOR(Schedules.EconomyPrice+(Schedules.EconomyPrice/100\*{2})) AS 'Cabin price', '0' AS 'Number of stops', Schedules.ID FROM Schedules WHERE Date >= '{3}' AND Date <= '{4}' AND RouteID = (SELECT ID FROM Routes WHERE DepartureAirportID = (SELECT ID FROM Airports WHERE IATACode = '{0}') AND ArrivalAirportID = (SELECT ID FROM Airports WHERE IATACode = '{1}'))", SearchParametersTo\_Combobox.SelectedValue, SearchParametersFrom\_Combobox.SelectedValue, CountCabinPrice, LowDateReturn, HighDateReturn), connection);

DASearchReturn.Fill(DSSearchReturn);

SearchReturnGrid\_View.DataSource = DSSearchReturn.Tables[0];

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

finally

{

connection.Close();

}

}

private void SearchBook\_Btn\_Click(object sender, EventArgs e)

{

string ReturnDate = "";

string ReturnNumber = "";

int ReturnCost = 0;

string ReturnSchID = "";

if (SearchReturnGrid\_View.CurrentRow != null)

{

ReturnDate = SearchReturnGrid\_View.CurrentRow.Cells[2].Value.ToString().Remove(SearchReturnGrid\_View.CurrentRow.Cells[2].Value.ToString().Length-8);

ReturnNumber = SearchReturnGrid\_View.CurrentRow.Cells[4].Value.ToString();

ReturnCost = Convert.ToInt32(SearchReturnGrid\_View.CurrentRow.Cells[5].Value);

ReturnSchID = SearchReturnGrid\_View.CurrentRow.Cells[7].Value.ToString();

}

if (SearchPassengers\_Box.Text.Length > 0 && SearchOutboundGrid\_View.CurrentRow.Cells[0].Value.ToString().Length > 0)

{

BookingFlight BookingFlightForm = new BookingFlight(SearchParametersFrom\_Combobox.SelectedValue.ToString(), SearchParametersTo\_Combobox.SelectedValue.ToString(), Convert.ToInt32(SearchParametersCabin\_Combobox.SelectedValue), SearchOutboundGrid\_View.CurrentRow.Cells[2].Value.ToString().Remove(SearchOutboundGrid\_View.CurrentRow.Cells[2].Value.ToString().Length-8), ReturnDate, SearchOutboundGrid\_View.CurrentRow.Cells[4].Value.ToString(), ReturnNumber, Returning, Convert.ToInt32(SearchOutboundGrid\_View.CurrentRow.Cells[5].Value), ReturnCost, SearchOutboundGrid\_View.CurrentRow.Cells[7].Value.ToString(), ReturnSchID, UserID);

BookingFlightForm.Show();

}

else MessageBox.Show("Enter number of passengers or select flight");

}

}

}

Листинг файла Summary.cs

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Avia

{

public partial class Summary : Form

{

public String conStr = "Data Source=ANDREYNOTE\\FOREMDB;Initial Catalog=session;Persist Security Info=True;User ID=sa;Password=12345";

private SqlConnection connection;

private SqlCommand cmd;

private DataSet DS;

private SqlDataAdapter da;

private SqlDataReader ExecuteReader;

public Summary()

{

InitializeComponent();

int ConfirmedFlightCounter = 0;

int CancelledFlightCounter = 0;

int[] AverageFlightCounter = new int[20000];

DateTime LowDate;

DateTime HighDate;

connection = new SqlConnection(conStr);

try

{

connection.Open();

cmd = new SqlCommand(String.Format("Select Schedules.Confirmed AS 'ConfirmedFlight', FlightTime, Date From Tickets INNER JOIN Schedules ON Tickets.ScheduleID = Schedules.ID INNER JOIN Routes ON Schedules.RouteID = Routes.ID WHERE Date <= GETDATE() AND Date > dateadd(day, -30, getdate())"), connection);

ExecuteReader = cmd.ExecuteReader();

int CounterArray = 0;

while (ExecuteReader.Read())

{

if (ExecuteReader["ConfirmedFlight"].ToString() == "True") ConfirmedFlightCounter++;

else CancelledFlightCounter++;

AverageFlightCounter[CounterArray] = Convert.ToInt32(ExecuteReader["FlightTime"]);

CounterArray++;

}

ExecuteReader.Close();

int FlightTimeResult = 0;

for (int i = 0; i < CounterArray; i++)

{

FlightTimeResult += AverageFlightCounter[i];

}

FlightTimeResult = FlightTimeResult / CounterArray;

SummaryFlightsCancelled\_Label.Text = "Number confirmed: " + ConfirmedFlightCounter.ToString();

SummaryFlightsCancelled\_Label.Text = "Number cancelled: " + CancelledFlightCounter.ToString();

SummaryFlightsTime\_Label.Text = "Average flight time: " + FlightTimeResult.ToString() + "mins";

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

finally

{

connection.Close();

}

}

private void SummaryExit\_Btn\_Click(object sender, EventArgs e)

{

this.Close();

}

}

}

Результат работы программы:

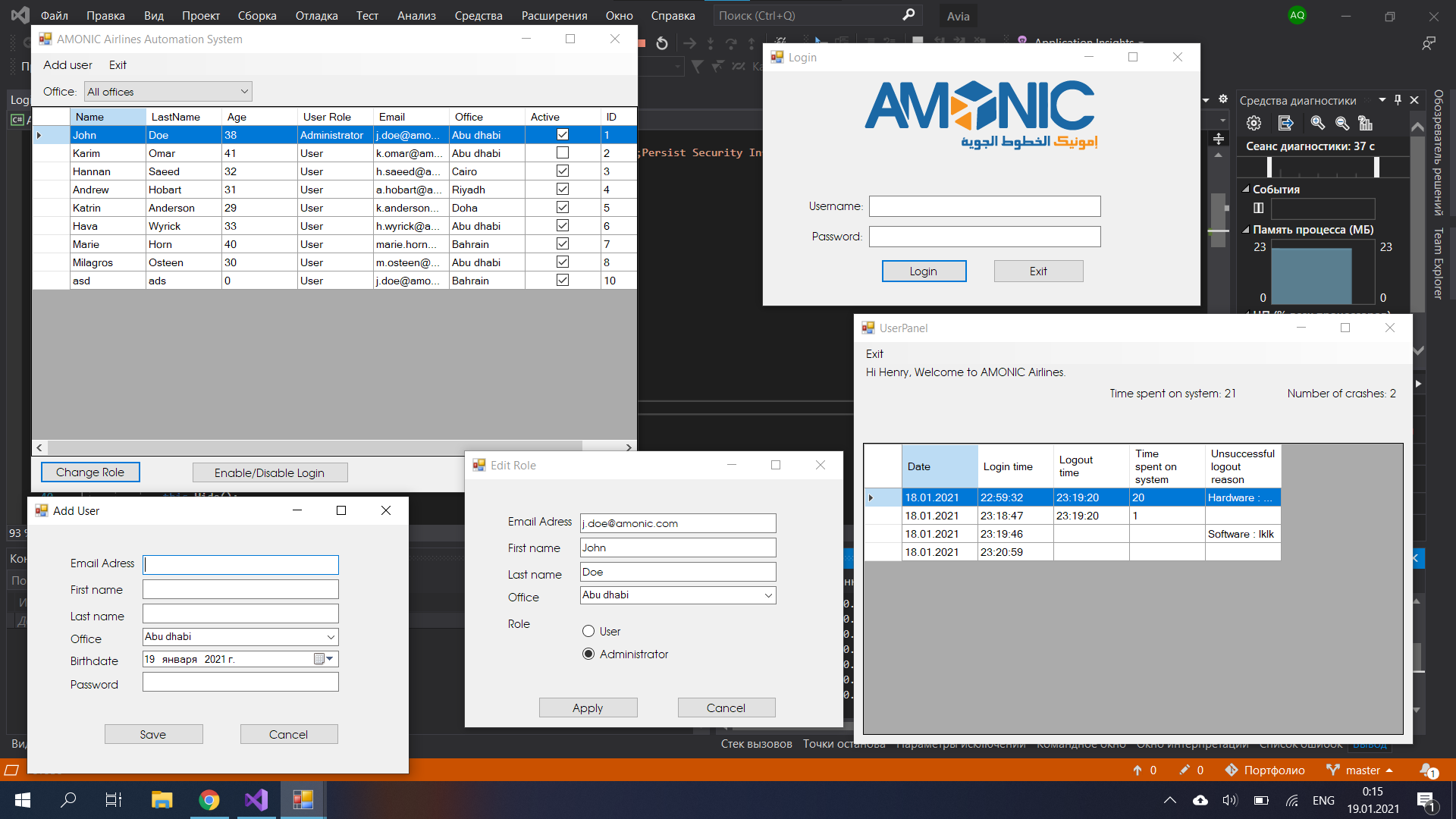


Рисунок – Сессия 1

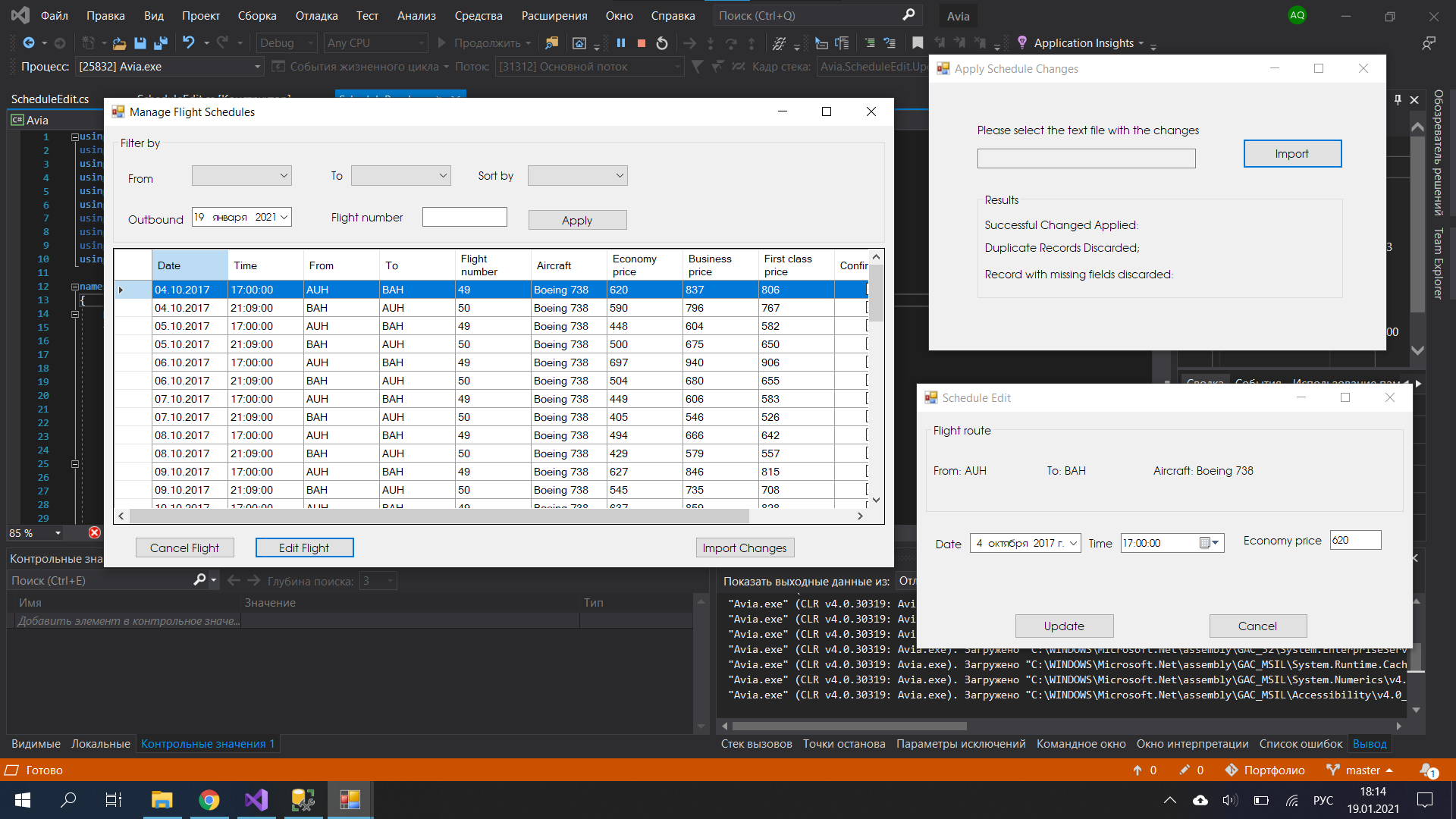


Рисунок - Сессия 2

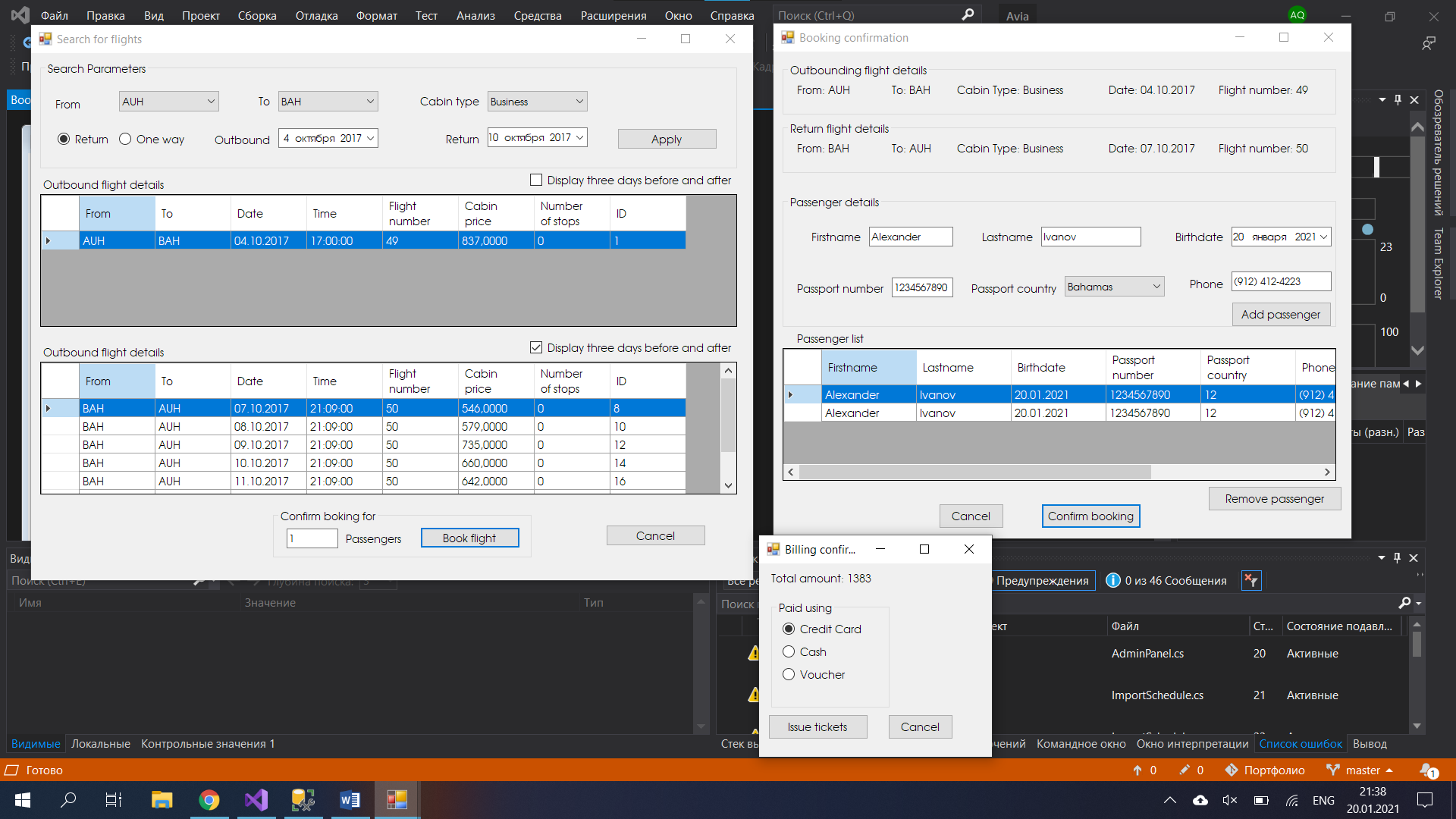


Рисунок – Сессия 3

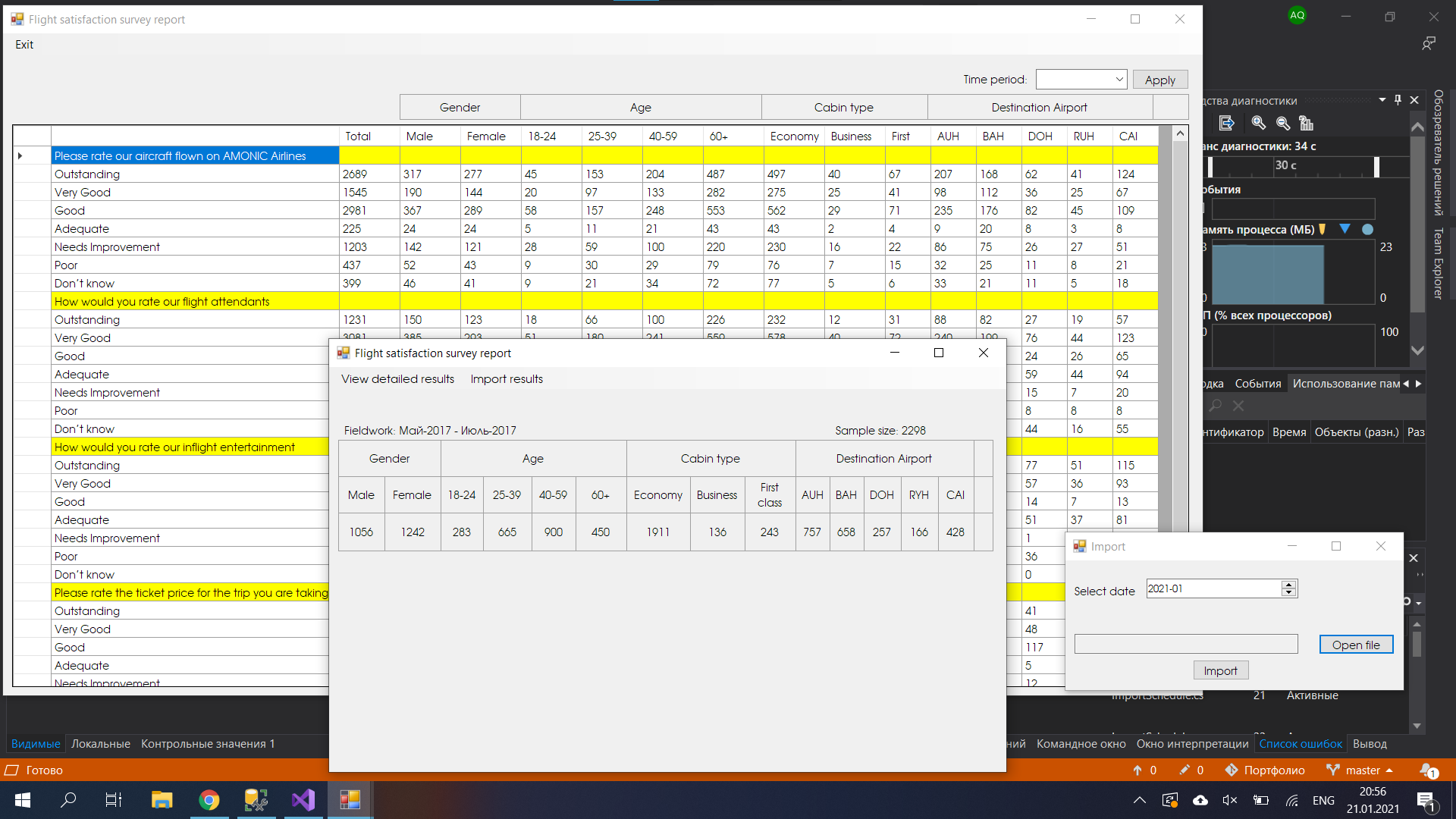


Рисунок – Сессия 4

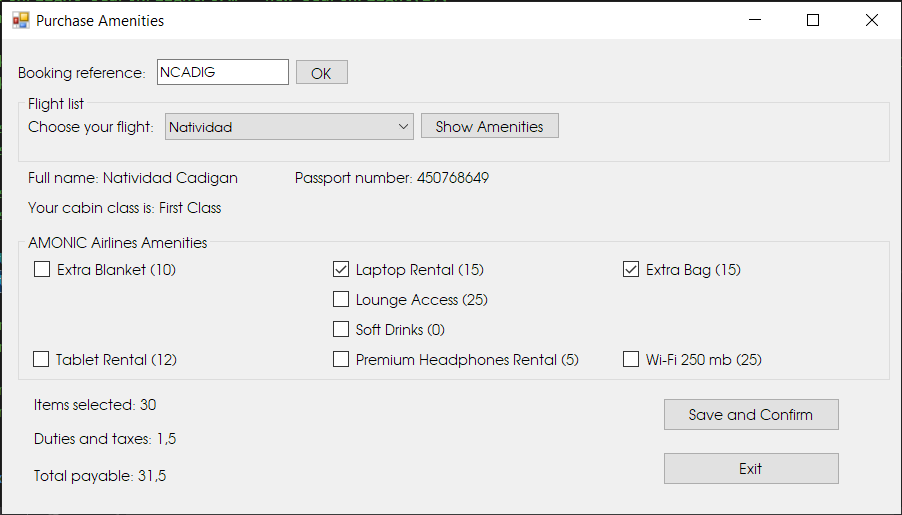


Рисунок – Сессия 5