Автоматизированная система ВУЗа «Учет кадрового обеспечения ВУЗа и составления расписания».

Автоматизированная система ВУЗа должна решать следующие проблемы:

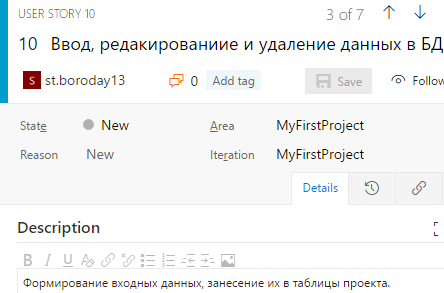
* Составление расписания;
* Учет кадров;
* Учет объектов ВУЗа.

Для разработки системы была использована технология командной разработки.

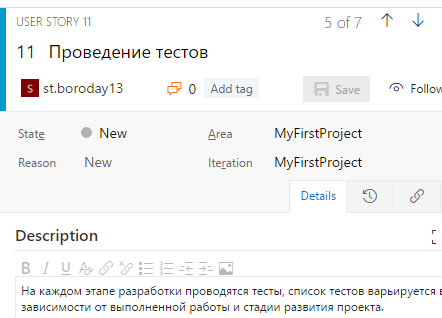
Среда разработки Microsoft Visual Studio.

User Story и User Tasks

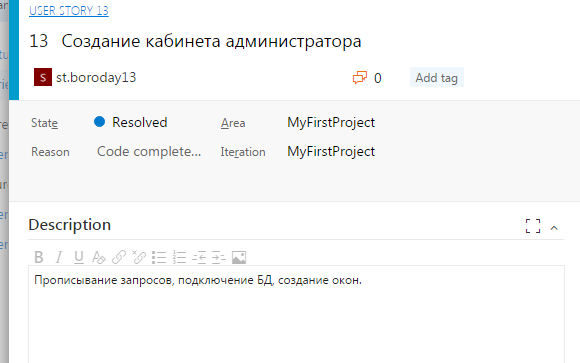
Работа с бд, запросы на ввод, удаление и изменение данных



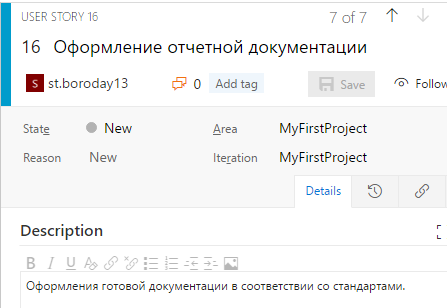
Проведение тестов. Создание unit-тестов приложения.



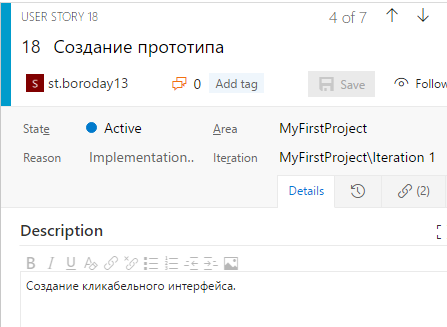
Создание серверной части. Настройка БД. Установка неообходимых библиотек.



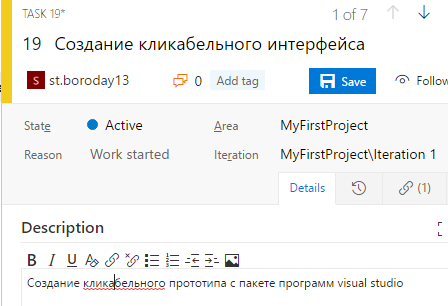
Формирование отчетной документации приложения.



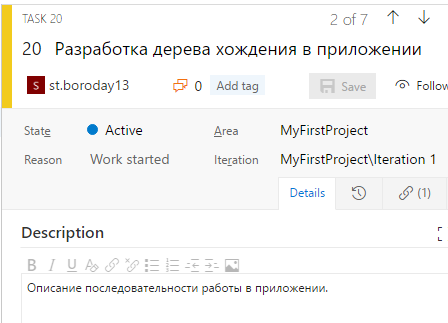
Создание прототипа интерфейса



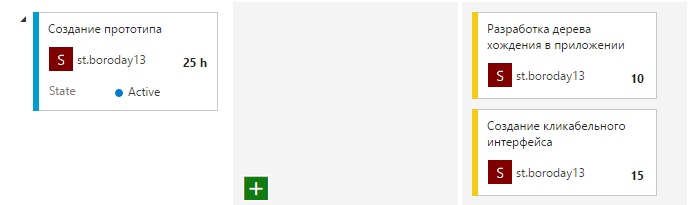
Создание кликабельного интерфейса приложения



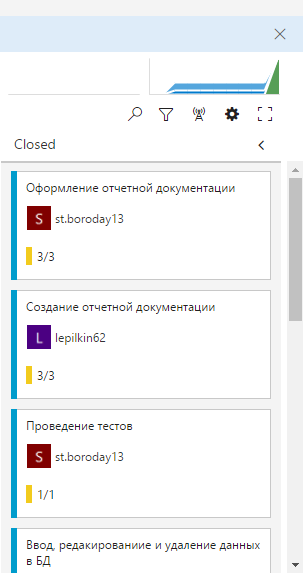
Разработка дерева хождения в приложении



Примеры заданий и распределение трудозатрат для User Story Создание прототипа.



Законченный проект



Описание базы данных приложения.

БД приложения разработана в приложении Microsoft Office Access

Бд состоит из 4 таблиц:

Таблица User

Содержит информацию о пользователях системы

Код – индентификатор

Email – почта сотрудника

Isadmin – атрибут принадлежности к классу администратор

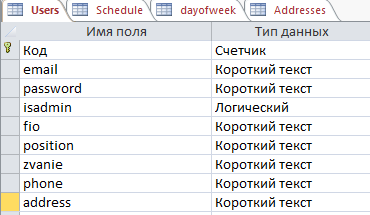
FIO – фамилия, имя, отчетсво

Position – должность

Zvanie – научное звание

Phone – номер телефона

Address – адрес места жительства



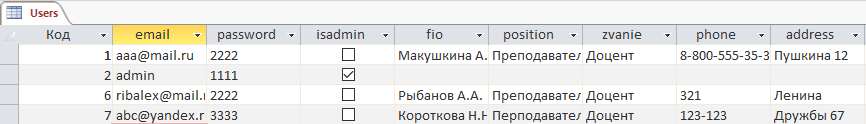


Таблица Shedular

Содержит информацию о раписании, связана с таблицей Users при помощи атрибута UserID

Код –индентификатор

UserID – поле для связи с таблицей Users

Day – день недели

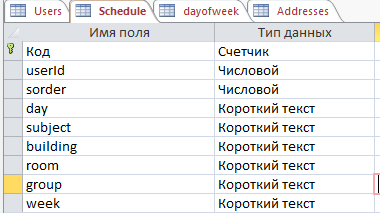
Subject – название предмета

Building – строение корпуса

Room – номер аудитории

Group – группа, у которой будут занятия

Week – неделя (Чередуются неделя А и неделя Б)



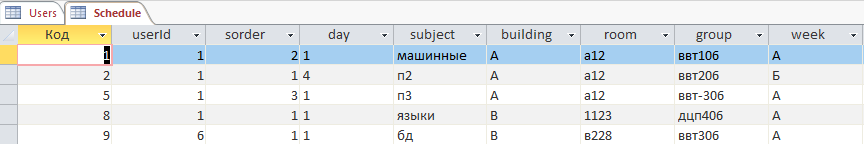
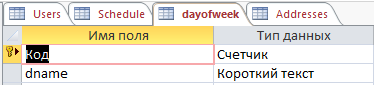


Таблица dayofweek

Создана для связи с таблицей Schedule при помощи атрибута Код

Код – идентификатор дня недели

Dname – название дня недели



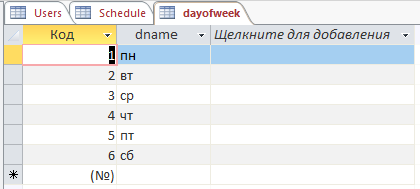


Таблица Addresses

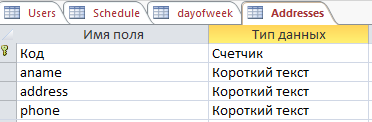
Содержит в себе описание необходимых объектов ВУЗа (корпуса, строения, подразделения)

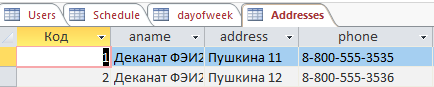
Код – идентификатор объекта

Aname – название объекта

Address – адрес объекта

Phone – телефон объекта





Модульное тестирование приложения

using System;

using Microsoft.VisualStudio.TestTools.UnitTesting;

namespace AdminTeacherTest

{

[TestClass]

public class UnitTest1

{

[TestMethod]

public void getUserScheduleTest()

{

AdminTeacher.DBHelper h = new AdminTeacher.DBHelper();

//assert

Assert.IsNotNull(h.getWorkers());

}

[TestMethod]

public void addAddressTest()

{

AdminTeacher.DBHelper h = new AdminTeacher.DBHelper();

string[] s = new string[3];

s[0] = "a";

s[1] = "b";

s[2] = "c";

//assert

Assert.IsTrue(h.addAddress(s));

}

[TestMethod]

public void deleteAddressTest()

{

AdminTeacher.DBHelper h = new AdminTeacher.DBHelper();

//assert

Assert.IsTrue(h.deleteAddress("a"));

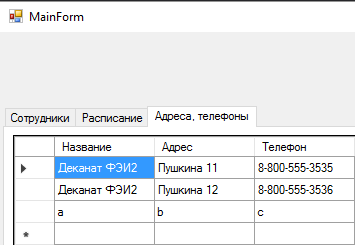
}

}

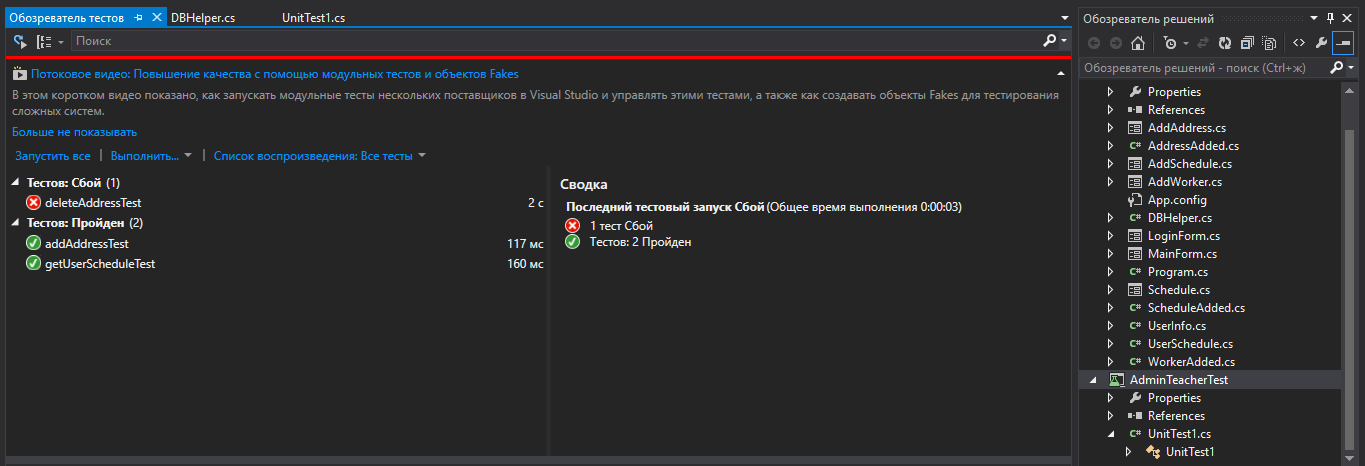
}

Тест 1 выполнится успешно

Тест 2 выполнится успешно, результаты занесутся в таблицу Адреса и телефоны



Тест 3 не выполнится, несоответсвующее поле



Прототип интерфейса

Окно авторизации



Меню приложения для сотрудников



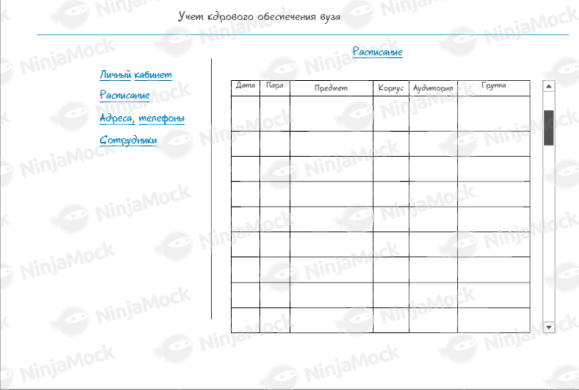
Личный кабинет сотрудников



Адреса и телефоны ВУЗа



Расписание для авторизированного преподавателя



Информация о сотрудниках ВУЗа



Авторизация от имени администратора и панель управления администратора



Листинг программы

AddAddress

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 AdminTeacher

{

public partial class AddAddress : Form

{

private string[] data;

private bool isAdd = true;

private AddressAdded adAdded;

public AddAddress(string[] data, AddressAdded i)

{

InitializeComponent();

adAdded = i;

if (data != null)

{

this.data = data;

isAdd = false;

button1.Text = "Изменить";

textBox1.Text = data[0];

textBox2.Text = data[1];

textBox3.Text = data[2];

}

}

public AddAddress(AddressAdded i)

{

InitializeComponent();

adAdded = i;

data = new string[3];

}

private void button2\_Click(object sender, EventArgs e)

{

this.Close();

}

private void button1\_Click(object sender, EventArgs e)

{

data[0] = textBox1.Text;

data[1] = textBox2.Text;

data[2] = textBox3.Text;

if (isAdd)

{

addRecord();

}

else

{

editRecord();

}

adAdded.addressAdded();

this.Close();

}

private void addRecord()

{

Program.getDBHelper().addAddress(data);

}

private void editRecord()

{

Program.getDBHelper().editAddress(data);

}

}

}

AddShedular

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 AdminTeacher

{

public partial class AddSchedule : Form

{

private UserSchedule us;

private bool isAdd = false;

private ScheduleAdded sa;

public AddSchedule(UserSchedule s, ScheduleAdded sa)

{

InitializeComponent();

us = s;

this.sa = sa;

isAdd = false;

comboBox1.Items.Add("ПН");

comboBox1.Items.Add("ВТ");

comboBox1.Items.Add("СР");

comboBox1.Items.Add("ЧТ");

comboBox1.Items.Add("ПТ");

comboBox1.Items.Add("СБ");

comboBox2.Items.Add("А");

comboBox2.Items.Add("Б");

textBox1.Text = us.order;

textBox2.Text = us.subject;

textBox3.Text = us.building;

textBox4.Text = us.room;

textBox5.Text = us.group;

comboBox1.SelectedIndex = Int32.Parse(us.day) - 1;

comboBox2.SelectedIndex = getWeekNumber(us.week);

button1.Text = "Изменить";

}

public AddSchedule(string userId, ScheduleAdded sa)

{

InitializeComponent();

isAdd = true;

this.sa = sa;

us = new UserSchedule();

us.userId = userId;

comboBox1.Items.Add("ПН");

comboBox1.Items.Add("ВТ");

comboBox1.Items.Add("СР");

comboBox1.Items.Add("ЧТ");

comboBox1.Items.Add("ПТ");

comboBox1.Items.Add("СБ");

comboBox2.Items.Add("А");

comboBox2.Items.Add("Б");

button1.Text = "Добавить";

}

private void button2\_Click(object sender, EventArgs e)

{

this.Close();

}

private int getWeekNumber(string w)

{

if (w.Equals("А"))

{

return 0;

}

else

{

return 1;

}

}

private void button1\_Click(object sender, EventArgs e)

{

us.order = textBox1.Text;

us.subject = textBox2.Text;

us.building = textBox3.Text;

us.room = textBox4.Text;

us.group = textBox5.Text;

us.day = (comboBox1.SelectedIndex + 1) + "";

us.week = comboBox2.SelectedIndex == 0 ? "А" : "Б";

if (isAdd)

{

addSchedule();

}

else

{

updateSchedule();

}

sa.scheduleAdded();

this.Close();

}

private void updateSchedule()

{

Program.getDBHelper().updateSchedule(us);

}

private void addSchedule()

{

Program.getDBHelper().addSchedule(us);

}

}

}

AddWorker

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 AdminTeacher

{

public partial class AddWorker : Form

{

private string[] data;

private bool isAdd = true;

private WorkerAdded wAdded;

public AddWorker(string[] data, WorkerAdded i)

{

InitializeComponent();

wAdded = i;

if (data != null)

{

this.data = data;

isAdd = false;

textBox1.Text = data[3];

textBox2.Text = data[0];

textBox3.Text = data[1];

textBox4.Text = data[2];

textBox5.Text = data[4];

textBox6.Text = data[5];

textBox7.Text = data[7];

button1.Text = "Изменить";

}

}

public AddWorker(WorkerAdded i)

{

InitializeComponent();

wAdded = i;

data = new string[7];

}

private void button2\_Click(object sender, EventArgs e)

{

this.Close();

}

private void button1\_Click(object sender, EventArgs e)

{

data[3] = textBox1.Text;

data[0] = textBox2.Text;

data[1] = textBox3.Text;

data[2] = textBox4.Text;

data[4] = textBox5.Text;

data[5] = textBox6.Text;

if (isAdd)

{

data[6] = textBox7.Text;

addUser();

}

else

{

data[7] = textBox7.Text;

editUser();

}

wAdded.workerAdded();

this.Close();

}

private void addUser()

{

Program.getDBHelper().addWorker(data);

}

private void editUser()

{

Program.getDBHelper().editWorker(data);

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Data.OleDb;

using System.Data.SqlClient;

using System.Data;

namespace AdminTeacher

{

public class DBHelper

{

private OleDbCommand cmd = new OleDbCommand();

private OleDbConnection cn = new OleDbConnection();

private string dbPath = @"C:\Users\Иван\YandexDisk\Инста\8 семестр\8 семестр. Технол. команд. разр. прогр. сист\Контрольная\AdminTeacher\MyDB2.accdb";

public DBHelper()

{

cn.ConnectionString = @"Provider=Microsoft.ACE.OLEDB.12.0;Data Source=" + dbPath;

cmd.Connection = cn;

}

public List<String[]> getWorkers()

{

List<String[]> result = new List<String[]>();

string[] cels;

string q = "select \* from Users where isadmin=False";

OleDbCommand cm = new OleDbCommand(q, cn);

cn.Open();

OleDbDataReader reader = cm.ExecuteReader();

while (reader.Read())

{

cels = new string[8];

cels[0] = reader[1].ToString();

cels[1] = reader[4].ToString();

cels[2] = reader[5].ToString();

cels[3] = reader[6].ToString();

cels[4] = reader[7].ToString();

cels[5] = reader[8].ToString();

cels[6] = reader[2].ToString();

cels[7] = reader[0].ToString();

result.Add(cels);

}

reader.Close();

cn.Close();

return result;

}

public bool addAddress(string[] data)

{

string q = "insert into [Addresses] ([aname], [address], [phone])"

+ " values ('" + data[0]+ "', '" + data[1] + "', '" + data[2] + "')";

return executeCommand(q);

}

public bool deleteAddress(string id)

{

string q = "delete from Addresses where Код=" + id + "";

return executeCommand(q);

}

public bool deleteWorker(string id)

{

string q = "delete from Users where Код=" + id + "";

return executeCommand(q);

}

public bool deleteSchedule(string id)

{

string q = "delete from Schedule where Код=" + id + "";

return executeCommand(q);

}

public bool editAddress(string[] data)

{

string q = "update [Addresses] set [aname]='" + data[0] + "', [address]='" +

data[1] + "', [phone]='" + data[2] + "' where Код=" + data[3];

return executeCommand(q);

}

public bool editWorker(string[] data)

{

string q = "update [Users] set [email]='" + data[3] + "', [fio]='" +

data[0] + "', [position]='" + data[1] + "', [zvanie]='" + data[2] + "', "

+ "[phone]='" + data[4] + "', " + "[address]='," + data[5]

+ "', [password]='" + data[7] + "' where Код=" + data[6];

return executeCommand(q);

}

public bool updateSchedule(UserSchedule us)

{

string q = "update [Schedule] set [sorder]=" + us.order + ", [mday]=" + us.day +

", [subject]='" + us.subject + "', [building]='" + us.building +

"', [room]='" + us.room + "', [mgroup]='" + us.group + "', [week]=" + getWeek(us.week) +

" where Код=" + us.id;

return executeCommand(q);

}

public bool addWorker(string[] data)

{

string q = "insert into [Users] ([email], [password], [isadmin], [fio], [position], [zvanie], [phone], [address])"

+ " values ('" + data[3] + "', '" + data[6] + "', 0, '" + data[0] + "', '" + data[1] +

"', '" + data[2] + "', '" + data[4] + "', '" + data[5]

+"')";

return executeCommand(q);

}

private string getWeek(string week)

{

return week.Equals("А") ? "1" : "2";

}

public bool addSchedule(UserSchedule us)

{

string q = "insert into [Schedule] ([userId], [sorder], [mday], [subject], [building], [room], [mgroup], [week]) values " +

"(" + us.userId + ", " + us.order + ", " + us.day + ", '" + us.subject + "', '" + us.building + "', '" + us.room + "', '" +

us.group + "', " + getWeek(us.week) + ")";

return executeCommand(q);

}

public bool checkRoom(UserSchedule us)

{

string q = "select \* from Schedule where mday=" + us.day + " and room='" + us.room + "' and week=" + getWeek(us.week) +

" and building='" + us.building + "' and sorder=" + us.order;

OleDbCommand cm = new OleDbCommand(q, cn);

bool result = true;

cn.Open();

OleDbDataReader reader = cm.ExecuteReader();

while (reader.Read())

{

if (reader[0] == null)

{

result = true;

}

else

{

result = false;

}

}

reader.Close();

cn.Close();

return result;

}

public List<string> getAllGroups()

{

List<string> result = new List<string>();

string q = "SELECT DISTINCT mgroup from Schedule";

OleDbCommand cm = new OleDbCommand(q, cn);

cn.Open();

OleDbDataReader reader = cm.ExecuteReader();

while (reader.Read())

{

result.Add(reader[0].ToString());

}

reader.Close();

cn.Close();

return result;

}

public List<UserSchedule> getGroupsSchedule(string group)

{

List<UserSchedule> result = new List<UserSchedule>();

string q = "SELECT Schedule.userId, Schedule.sorder, Schedule.mday, Schedule.subject, Schedule.building, Schedule.room, Schedule.week, Users.fio " +

"FROM [Schedule], [Users] WHERE (([Schedule].[mgroup]='" +group + "') AND ([Schedule].[userId]=[Users].[Код])) order by week, mday, sorder";

OleDbCommand cm = new OleDbCommand(q, cn);

cn.Open();

OleDbDataReader reader = cm.ExecuteReader();

while (reader.Read())

{

UserSchedule u = new UserSchedule();

u.order = reader[1].ToString();

u.day = reader[2].ToString();

u.subject = reader[3].ToString();

u.building = reader[4].ToString();

u.room = reader[5].ToString();

u.week = reader[6].ToString();

u.userId = reader[7].ToString();

result.Add(u);

}

reader.Close();

cn.Close();

return result;

}

public List<UserSchedule> getUserSchedule(String id)

{

List<UserSchedule> result = new List<UserSchedule>();

string q = "select \* from Schedule where userid=" + id + " order by week, mday, sorder";

OleDbCommand cm = new OleDbCommand(q, cn);

cn.Open();

OleDbDataReader reader = cm.ExecuteReader();

while (reader.Read())

{

UserSchedule u = new UserSchedule(

reader[0].ToString(), reader[1].ToString(),

reader[2].ToString(), reader[3].ToString(),

reader[4].ToString(), reader[5].ToString(),

reader[6].ToString(), reader[7].ToString(),

reader[8].ToString());

result.Add(u);

}

reader.Close();

cn.Close();

return result;

}

private bool executeCommand(string command)

{

try

{

cn.Open();

cmd.CommandText = command;

cmd.ExecuteNonQuery();

cn.Close();

}

catch (Exception er)

{

cn.Close();

return false;

}

return true;

}

public List<String[]> getAddresses()

{

List<String[]> result = new List<String[]>();

string[] cels;

string q = "select \* from Addresses";

OleDbCommand cm = new OleDbCommand(q, cn);

cn.Open();

OleDbDataReader reader = cm.ExecuteReader();

while (reader.Read())

{

cels = new string[4];

cels[0] = reader[1].ToString();

cels[1] = reader[2].ToString();

cels[2] = reader[3].ToString();

cels[3] = reader[0].ToString();

result.Add(cels);

}

reader.Close();

cn.Close();

return result;

}

public bool loginUser(string email, string pass)

{

string[] cels = {"", "", "", "", "", "", "", "", ""};

string q = "select \* from Users where email='" + email + "' and password='" + pass + "'";

OleDbCommand cm = new OleDbCommand(q, cn);

cn.Open();

OleDbDataReader reader = cm.ExecuteReader();

while (reader.Read())

{

cels[0] = reader[1].ToString();

cels[1] = reader[2].ToString();

cels[2] = reader[3].ToString();

cels[3] = reader[4].ToString();

cels[4] = reader[5].ToString();

cels[5] = reader[6].ToString();

cels[6] = reader[7].ToString();

cels[7] = reader[8].ToString();

cels[8] = reader[0].ToString();

}

reader.Close();

cn.Close();

if (cels[2].Length > 0)

{

Program.getUserInfo().isAdmin = Boolean.Parse(cels[2]);

if (!Program.getUserInfo().isAdmin)

{

Program.getUserInfo().email = cels[0];

Program.getUserInfo().fio = cels[3];

Program.getUserInfo().position = cels[4];

Program.getUserInfo().zvanie = cels[5];

Program.getUserInfo().phone = cels[6];

Program.getUserInfo().address = cels[7];

Program.getUserInfo().id = cels[8];

}

return true;

}

return false;

}

}

}

LoginForm

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 AdminTeacher

{

public partial class LoginForm : Form

{

public LoginForm()

{

InitializeComponent();

}

private void button1\_Click(object sender, EventArgs e)

{

if (textBox1.Text.Length > 0 && textBox2.Text.Length > 0)

{

if (Program.getDBHelper().loginUser(textBox1.Text, textBox2.Text))

{

this.Hide();

var form2 = new MainForm();

form2.Closed += (s, args) => this.Close();

form2.Show();

}

else

{

MessageBox.Show(this, "Пользователь не найден.");

}

}

else

{

MessageBox.Show(this, "Заполните все поля");

}

}

}

}

MainForm

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 AdminTeacher

{

public partial class MainForm : Form, AddressAdded, WorkerAdded

{

public MainForm()

{

InitializeComponent();

if (Program.getUserInfo().isAdmin)

{

initAdmin();

}

else

{

initTeacher();

}

}

private void initAdmin()

{

label1.Text = "Здравствуйте, Администратор.";

tabControl1.Visible = false;

tabControl2.Visible = true;

refreshAddress();

refreshUsers(dataGridView3);

showUsersForSchedule();

}

private void showUsersForSchedule()

{

refreshUsers(dataGridView5);

}

private void initTeacher()

{

label1.Text = "Здравствуйте, " + Program.getUserInfo().fio;

tabControl1.Visible = true;

tabControl2.Visible = false;

tabControl1.SelectedIndex = 0;

UserInfo ui = Program.getUserInfo();

label14.Text = ui.fio;

label13.Text = ui.position;

label12.Text = ui.zvanie;

label11.Text = ui.email;

label10.Text = ui.phone;

label9.Text = ui.address;

List<String[]> addresses = Program.getDBHelper().getAddresses();

if (addresses.Count > 0)

{

fillAddressTable(addresses, dataGridView1);

}

List<String[]> workers = Program.getDBHelper().getWorkers();

if (workers.Count > 0)

{

fillWorkersTable(workers, dataGridView2);

}

initSchedule(ui.id);

}

private void initSchedule(string id)

{

List<UserSchedule> sch = Program.getDBHelper().getUserSchedule(id);

if (sch.Count > 0)

{

fillScheduleTable(sch);

}

}

private void fillScheduleTable(List<UserSchedule> list)

{

DataTable table = new DataTable("Расписание");

DataColumn c0 = new DataColumn("День");

DataColumn c1 = new DataColumn("Предмет");

DataColumn c2 = new DataColumn("№ пары");

DataColumn c3 = new DataColumn("Корпус");

DataColumn c4 = new DataColumn("Аудитория");

DataColumn c5 = new DataColumn("Группа");

DataColumn c6 = new DataColumn("Неделя");

DataColumn c7 = new DataColumn("ID");

DataColumn c8 = new DataColumn("UID");

table.Columns.Add(c0);

table.Columns.Add(c1);

table.Columns.Add(c2);

table.Columns.Add(c3);

table.Columns.Add(c4);

table.Columns.Add(c5);

table.Columns.Add(c6);

table.Columns.Add(c7);

table.Columns.Add(c8);

foreach (UserSchedule item in list)

{

DataRow row = table.NewRow();

row["День"] = getDay(item.day);

row["Предмет"] = item.subject;

row["№ пары"] = item.order;

row["Корпус"] = item.building;

row["Аудитория"] = item.room;

row["Группа"] = item.group;

row["Аудитория"] = item.room;

row["Неделя"] = item.week;

row["ID"] = item.id;

row["UID"] = item.userId;

table.Rows.Add(row);

}

dataGridView6.DataSource = table;

dataGridView6.Columns[dataGridView6.Columns.Count - 1].Visible = false;

dataGridView6.Columns[dataGridView6.Columns.Count - 2].Visible = false;

}

private string getDay(string d)

{

switch (d)

{

case "1":

return "пн";

case "2":

return "вт";

case "3":

return "ср";

case "4":

return "чт";

case "5":

return "пт";

case "6":

return "сб";

default:

return "";

}

}

private void fillWorkersTable(List<String[]> list, DataGridView grid)

{

DataTable table = new DataTable("Сотрудники");

DataColumn c0 = new DataColumn("ФИО");

DataColumn c1 = new DataColumn("Должность");

DataColumn c2 = new DataColumn("Научное звание");

DataColumn c3 = new DataColumn("Почта");

DataColumn c4 = new DataColumn("Телефон");

DataColumn c5 = new DataColumn("Адрес");

DataColumn c6 = new DataColumn("ID");

DataColumn c7 = new DataColumn("PAS");

table.Columns.Add(c0);

table.Columns.Add(c1);

table.Columns.Add(c2);

table.Columns.Add(c3);

table.Columns.Add(c4);

table.Columns.Add(c5);

table.Columns.Add(c6);

table.Columns.Add(c7);

foreach (String[] item in list)

{

DataRow row = table.NewRow();

row["ФИО"] = item[1];

row["Должность"] = item[2];

row["Научное звание"] = item[3];

row["Почта"] = item[0];

row["Телефон"] = item[4];

row["Адрес"] = item[5];

row["PAS"] = item[6];

row["ID"] = item[7];

table.Rows.Add(row);

}

grid.DataSource = table;

grid.Columns[grid.Columns.Count - 1].Visible = false;

grid.Columns[grid.Columns.Count - 2].Visible = false;

}

private void fillAddressTable(List<String[]> list, DataGridView grid)

{

DataTable table = new DataTable("Адреса");

DataColumn c0 = new DataColumn("Название");

DataColumn c1 = new DataColumn("Адрес");

DataColumn c2 = new DataColumn("Телефон");

DataColumn c3 = new DataColumn("ID");

table.Columns.Add(c0);

table.Columns.Add(c1);

table.Columns.Add(c2);

table.Columns.Add(c3);

foreach (String[] item in list)

{

DataRow row = table.NewRow();

row["Название"] = item[0];

row["Адрес"] = item[1];

row["Телефон"] = item[2];

row["ID"] = item[3];

table.Rows.Add(row);

}

grid.DataSource = table;

grid.Columns[grid.Columns.Count - 1].Visible = false;

}

private void label2\_Click(object sender, EventArgs e)

{

Program.logout();

this.Hide();

var form2 = new LoginForm();

form2.Closed += (s, args) => this.Close();

form2.Show();

}

private string[] getCells(DataGridView dataGrid)

{

int index = dataGrid.CurrentRow.Index;

string[] result = new string[dataGrid.ColumnCount];

for (int i = 0; i < result.Length; i++)

{

if (dataGrid.Rows[index].Cells[i].Value != null)

result[i] = dataGrid.Rows[index].Cells[i].Value.ToString();

}

return result;

}

private void button6\_Click(object sender, EventArgs e)

{

DataGridView d = tabControl2.SelectedIndex == 0 ? dataGridView3 : dataGridView4;

AddAddress f = new AddAddress(getCells(d), this);

f.Show();

}

private void button1\_Click(object sender, EventArgs e)

{

DataGridView d = tabControl2.SelectedIndex == 0 ? dataGridView3 : dataGridView4;

AddWorker f = new AddWorker(getCells(d), this);

f.Show();

}

private void button2\_Click(object sender, EventArgs e)

{

DataGridView d = tabControl2.SelectedIndex == 0 ? dataGridView3 : dataGridView4;

string[] data = getCells(d);

AddWorker f = new AddWorker(this);

f.Show();

}

private void button5\_Click(object sender, EventArgs e)

{

DataGridView d = tabControl2.SelectedIndex == 0 ? dataGridView3 : dataGridView4;

string[] data = getCells(d);

AddAddress f = new AddAddress(this);

f.Show();

}

private void button3\_Click(object sender, EventArgs e)

{

DataGridView d = tabControl2.SelectedIndex == 0 ? dataGridView3 : dataGridView4;

string[] data = getCells(d);

Program.getDBHelper().deleteWorker(data[data.Length - 2]);

refreshUsers(dataGridView3);

}

private void button4\_Click(object sender, EventArgs e)

{

DataGridView d = tabControl2.SelectedIndex == 0 ? dataGridView3 : dataGridView4;

string[] data = getCells(d);

Program.getDBHelper().deleteAddress(data[data.Length - 1]);

refreshAddress();

}

private void refreshAddress()

{

List<String[]> addresses = Program.getDBHelper().getAddresses();

if (addresses.Count > 0)

{

fillAddressTable(addresses, dataGridView4);

}

}

private void refreshUsers(DataGridView v)

{

List<String[]> workers = Program.getDBHelper().getWorkers();

if (workers.Count > 0)

{

fillWorkersTable(workers, v);

}

}

void AddressAdded.addressAdded()

{

refreshAddress();

}

void WorkerAdded.workerAdded()

{

refreshUsers(dataGridView3);

}

private void button7\_Click(object sender, EventArgs e)

{

string[] data = getCells(dataGridView5);

Schedule f = new Schedule(data[data.Length - 2]);

f.Show();

}

}

}

Program

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace AdminTeacher

{

static class Program

{

private static UserInfo ui;

private static DBHelper dbHelper;

/// <summary>

/// The main entry point for the application.

/// </summary>

[STAThread]

static void Main()

{

ui = new UserInfo();

dbHelper = new DBHelper();

Application.EnableVisualStyles();

Application.SetCompatibleTextRenderingDefault(false);

Application.Run(new LoginForm());

}

public static UserInfo getUserInfo()

{

if (ui == null) {

ui = new UserInfo();

}

return ui;

}

public static void logout()

{

ui = null;

}

public static DBHelper getDBHelper()

{

if (dbHelper == null)

{

dbHelper = new DBHelper();

}

return dbHelper;

}

}

}

Schedular

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 AdminTeacher

{

public partial class Schedule : Form, ScheduleAdded

{

private string userId;

public Schedule(string id)

{

InitializeComponent();

userId = id;

initSchedule(userId);

}

private void initSchedule(string id)

{

List<UserSchedule> sch = Program.getDBHelper().getUserSchedule(id);

if (sch.Count > 0)

{

fillTable(sch);

}

}

private void fillTable(List<UserSchedule> list)

{

DataTable table = new DataTable("Расписание");

DataColumn c0 = new DataColumn("День");

DataColumn c1 = new DataColumn("Предмет");

DataColumn c2 = new DataColumn("№ пары");

DataColumn c3 = new DataColumn("Корпус");

DataColumn c4 = new DataColumn("Аудитория");

DataColumn c5 = new DataColumn("Группа");

DataColumn c6 = new DataColumn("Неделя");

DataColumn c7 = new DataColumn("ID");

DataColumn c8 = new DataColumn("UID");

table.Columns.Add(c0);

table.Columns.Add(c1);

table.Columns.Add(c2);

table.Columns.Add(c3);

table.Columns.Add(c4);

table.Columns.Add(c5);

table.Columns.Add(c6);

table.Columns.Add(c7);

table.Columns.Add(c8);

foreach (UserSchedule item in list)

{

DataRow row = table.NewRow();

row["День"] = getDay(item.day);

row["Предмет"] = item.subject;

row["№ пары"] = item.order;

row["Корпус"] = item.building;

row["Аудитория"] = item.room;

row["Группа"] = item.group;

row["Аудитория"] = item.room;

row["Неделя"] = item.week;

row["ID"] = item.id;

row["UID"] = item.userId;

table.Rows.Add(row);

}

dataGridView1.DataSource = table;

dataGridView1.Columns[dataGridView1.Columns.Count - 1].Visible = false;

dataGridView1.Columns[dataGridView1.Columns.Count - 2].Visible = false;

}

private string getDay(string d)

{

switch (d)

{

case "1":

return "пн";

case "2":

return "вт";

case "3":

return "ср";

case "4":

return "чт";

case "5":

return "пт";

case "6":

return "сб";

default:

return "";

}

}

private string getDayNumber(string d)

{

switch (d.ToLower())

{

case "пн":

return "1";

case "вт":

return "2";

case "ср":

return "3";

case "чт":

return "4";

case "пт":

return "5";

case "сб":

return "6";

default:

return "";

}

}

private void button3\_Click(object sender, EventArgs e)

{

string[] data = getCells(dataGridView1);

Program.getDBHelper().deleteSchedule(data[data.Length - 1]);

initSchedule(userId);

}

private string[] getCells(DataGridView dataGrid)

{

int index = dataGrid.CurrentRow.Index;

string[] result = new string[dataGrid.ColumnCount];

for (int i = 0; i < result.Length; i++)

{

if (dataGrid.Rows[index].Cells[i].Value != null)

result[i] = dataGrid.Rows[index].Cells[i].Value.ToString();

}

return result;

}

private void button2\_Click(object sender, EventArgs e)

{

string[] data = getCells(dataGridView1);

data[0] = getDayNumber(data[0]);

AddSchedule f = new AddSchedule(new UserSchedule(data), this);

f.Show();

}

private void button1\_Click(object sender, EventArgs e)

{

string[] data;

AddSchedule f;

try

{

data = getCells(dataGridView1);

f = new AddSchedule(data[data.Length - 2], this);

f.Show();

}

catch (Exception ex) {

f = new AddSchedule(userId, this);

f.Show();

}

}

void ScheduleAdded.scheduleAdded()

{

initSchedule(userId);

}

}

}