Содержание

[Введение 5](#_Toc41267963)

[1 Анализ предметной области 6](#_Toc41267964)

[2 Анализ методов решения 7](#_Toc41267965)

[3 Информационная модель системы 8](#_Toc41267966)

[3.1 Концептуальная модель 8](#_Toc41267967)

[3.1 Логическая модель 8](#_Toc41267968)

[3.2 Физическая модель 10](#_Toc41267969)

[4 Разработка программно-информационных компонентов системы 11](#_Toc41267970)

[4.1 Разработка пользовательского интерфейса 11](#_Toc41267971)

[4.2 Описание программных модулей 12](#_Toc41267972)

[Заключение 24](#_Toc41267973)

[Список используемых источников 25](#_Toc41267974)

[Приложение А – Антиплагиат 26](#_Toc41267975)

Введение

В наше время, которое называют веком информационных технологий, особое значение во всех сферах человеческой деятельности играет информация. Сейчас все большее внимание уделяется информированности о том или ином объекте, будь это крупная фирма или человек. Без полной информации практически невозможно решать серьезные задачи, добиваться реализации своей цели.

Отдел кадров – это место, куда стекается информация о рабочих предприятия. В отделе кадров содержится информация о людях, работающих на данном предприятии, которая составляет их персональные данные. Функциональная роль отдела кадров на предприятии очень велика по той причине, что именно в отделе кадров мы находим интересующую нас информацию о рабочих и по этой же причине мы должны обеспечить ее конфиденциальность т.е. ограничить доступ к ней посторонних лиц, а также контролировать доступ к ней.

1. Анализ предметной области

Основными предпосылками для исследования предметной области «Отдел кадров», является потребность в автоматизации работы инспектор отдела кадров – создание базы данных кадрового состава предприятия (организация добавления, модификации информации о работниках предприятия, должностные инструкции и назначения его должности и отдела для дальнейшей работы. Целью является решение вышеописанной проблемы и создание такого программного обеспечения, которое удовлетворит все потребности по передачи, хранению данных, тем самым достичь снижения финансовых и временных затрат.

Анализируя предметную область «Отдел кадров» можно установить, что инспектор отдела кадров вводит данные о сотруднике.

Субъектами данной автоматизированной системы являются:

1. Инспектор отдела кадров
2. Администратор
3. Сотрудники (инженер, оператор, бухгалтер)

Объектами данной автоматизированной системы являются:

1. Сотрудник
2. Должность
3. Отдел
4. Таблица логинов паролей

2 Анализ методов решения

Курсовая работа выполнена в интегрированной среде программирования MS Visual Studio 2017. Microsoft Visual Studio — это набор инструментов для создания программного обеспечения: от планирования до разработки пользовательского интерфейса, написания кода, тестирования, отладки, анализа качества кода и производительности, развертывания в средах клиентов и сбора данных телеметрии по использованию.

Эти инструменты предназначены для максимально эффективной совместной работы; все они доступны в интегрированной среде разработки (IDE) Visual Studio. Visual Studio можно использовать для создания различных типов приложений, от простых приложений до больших и сложных систем, обслуживающих предприятия и центры обработки данных.

* Приложения и игры, которые выполняются не только на платформе
* Windows, но и на Android и ios;
* Веб - сайты и веб - службы на основе ASP.NET, jquery, angularjs и других популярных платформ;
* Приложения для самых разных платформ и устройств, включая, но не ограничиваясь: Office, Sharepoint, Hololens, Kinect и "Интернета вещей";

По умолчанию Visual Studio обеспечивает поддержку C#, C и C++, JavaScript, F# и Visual Basic. Visual Studio хорошо работает и интегрируется со сторонними приложениями, например, Unity и Apache Cordova, с помощью расширений Набор средств Visual Studio для Unity и инструментов Visual Studio для Apache Cordova соответственно.

Также можно самостоятельно расширить Visual Studio, создав собственные инструменты для выполнения специализированных задач. Стоит отметить, что в данной работе используется Windows Form, доступный только в этой IDE.

1. Информационная модель системы

3.1 Концептуальная модель

Концептуальная модель – это модель, представленная множеством понятий и связей между ними, определяющих смысловую структуру рассматриваемой предметной области или её конкретного объекта.

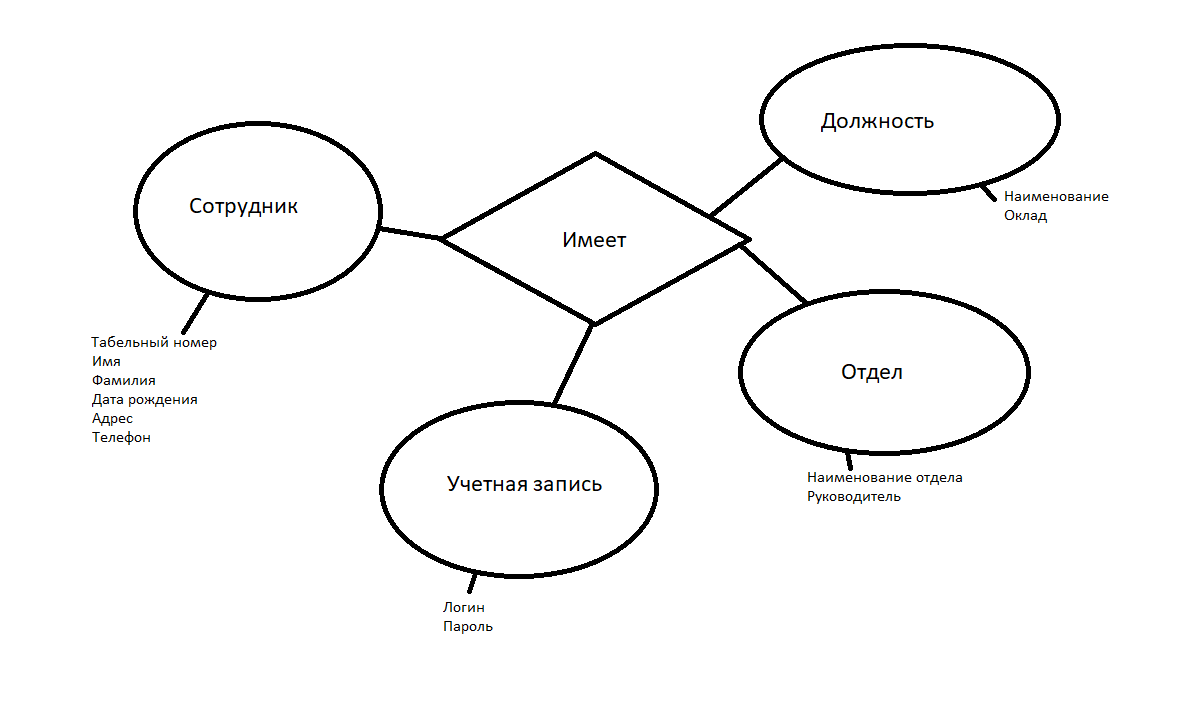


Рисунок 1 – Концептуальная модель

* 1. Логическая модель

Логическое проектирование — создание схемы базы данных на основе конкретной модели данных, например, реляционной модели данных. Для реляционной модели данных логическая модель — набор схем отношений, обычно с указанием первичных ключей, а также «связей» между отношениями, представляющих собой внешние ключи.

Логическая модель «Отдел Кадров» представлена на рисунке 2

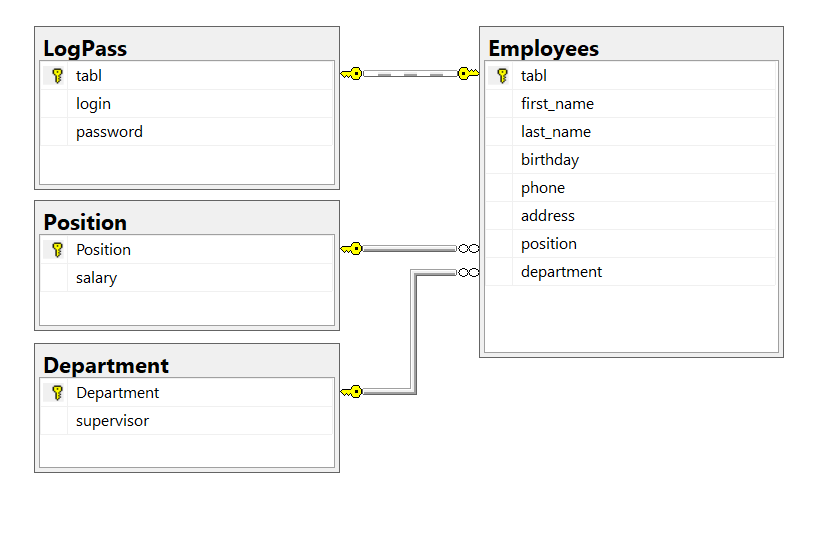


Рисунок 2 – Логическая модель

* 1. Физическая модель

Физическая модель представляет собой совокупность БД на материальном носителе.

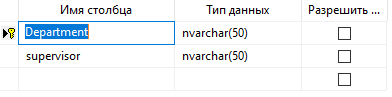


Рисунок 3 – Таблица отдела

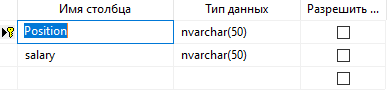


Рисунок 4 – Таблица должности

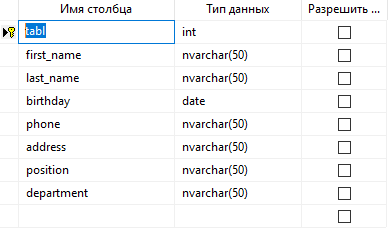


Рисунок 5 – Таблица сотрудников

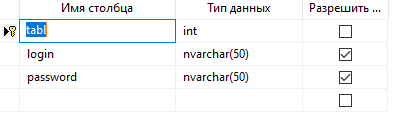


Рисунок 10 – Таблица учетной записи

1. Разработка программно-информационных компонентов системы
   1. Разработка пользовательского интерфейса

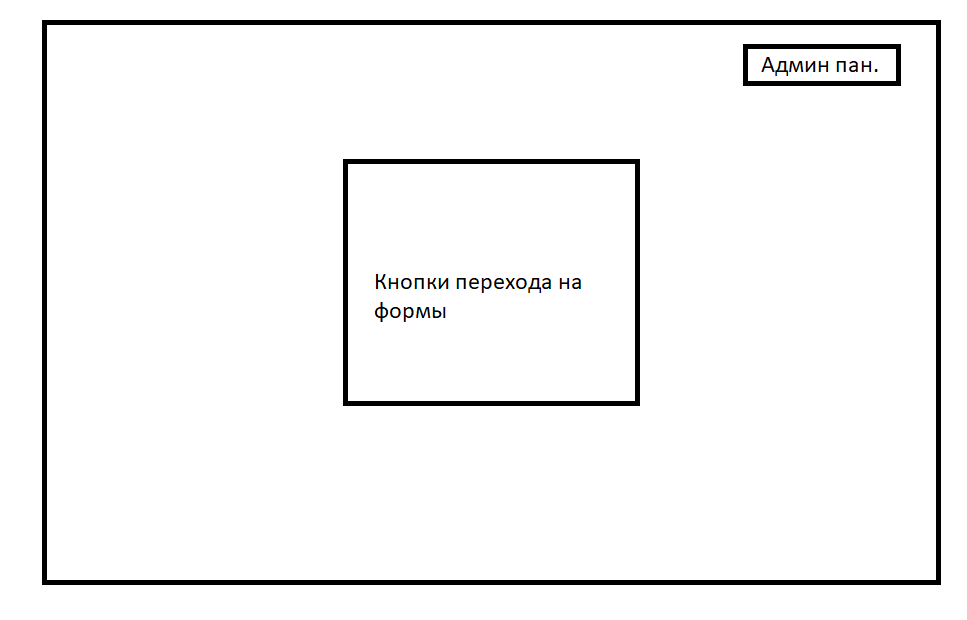


Рисунок 12 – Макет формы «Меню»

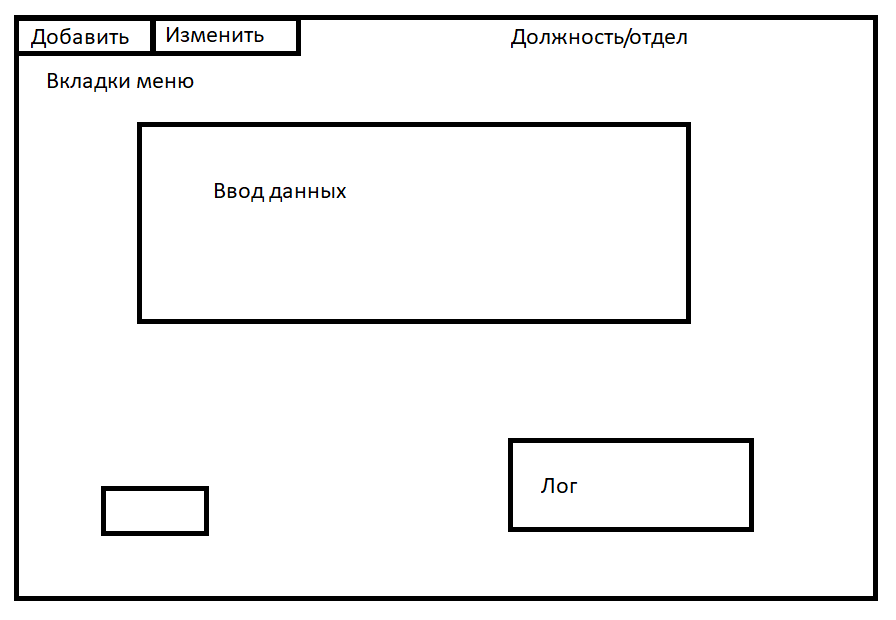


Рисунок 13 – Макет формы «Должность» и «Отдел»

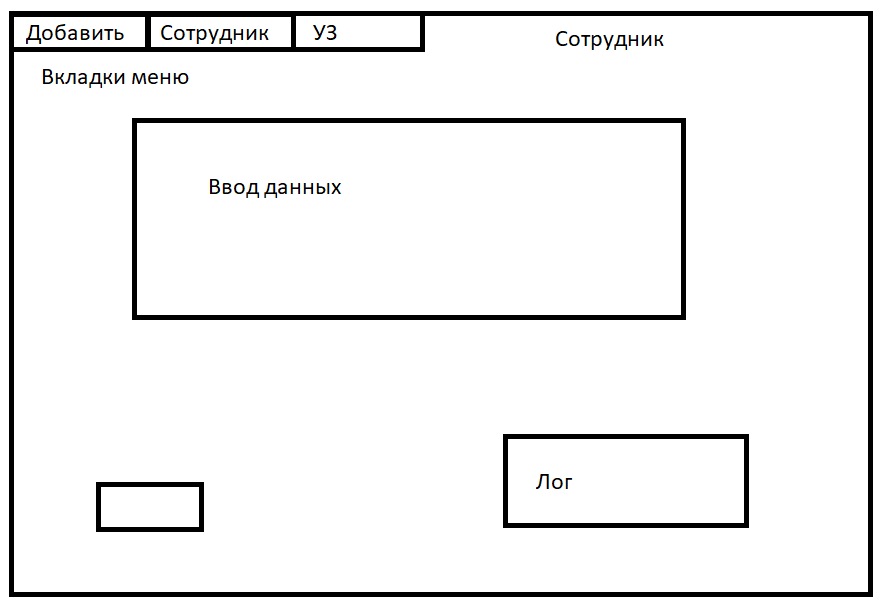


Рисунок 13 – Макет формы «Должность» и «Отдел»

* 1. Описание программных модулей

Данная программная среда позволяет реализовать объектные модули объектно-ориентированного программирования, и это дает возможность реализовать множество различных программных модулей, которые были использовано в данном курсовом проекте.

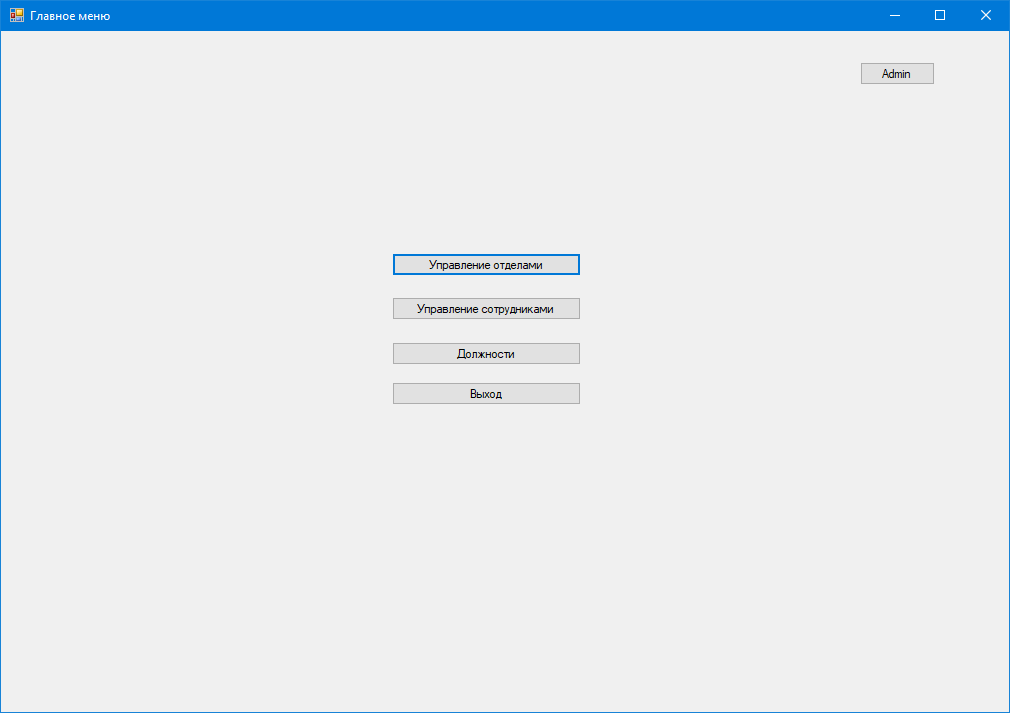


Рисунок 14 – Начальная форма «Menu»

Код формы:

using System;

using System.Windows.Forms;

namespace dbtest

{

public partial class Main : Form

{

public Main()

{

InitializeComponent();

}

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

{

Main.ActiveForm.Hide();

Department Department = new Department();

Department.ShowDialog();

Close();

}

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

{

Main.ActiveForm.Hide();

Employees Employees = new Employees();

Employees.ShowDialog();

Close();

}

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

{

Main.ActiveForm.Hide();

Positions Positions = new Positions();

Positions.ShowDialog();

Close();

}

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

{

Application.Exit();

}

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

{

Main.ActiveForm.Hide();

Admin Admin = new Admin();

Admin.ShowDialog();

Close();

}

}

}

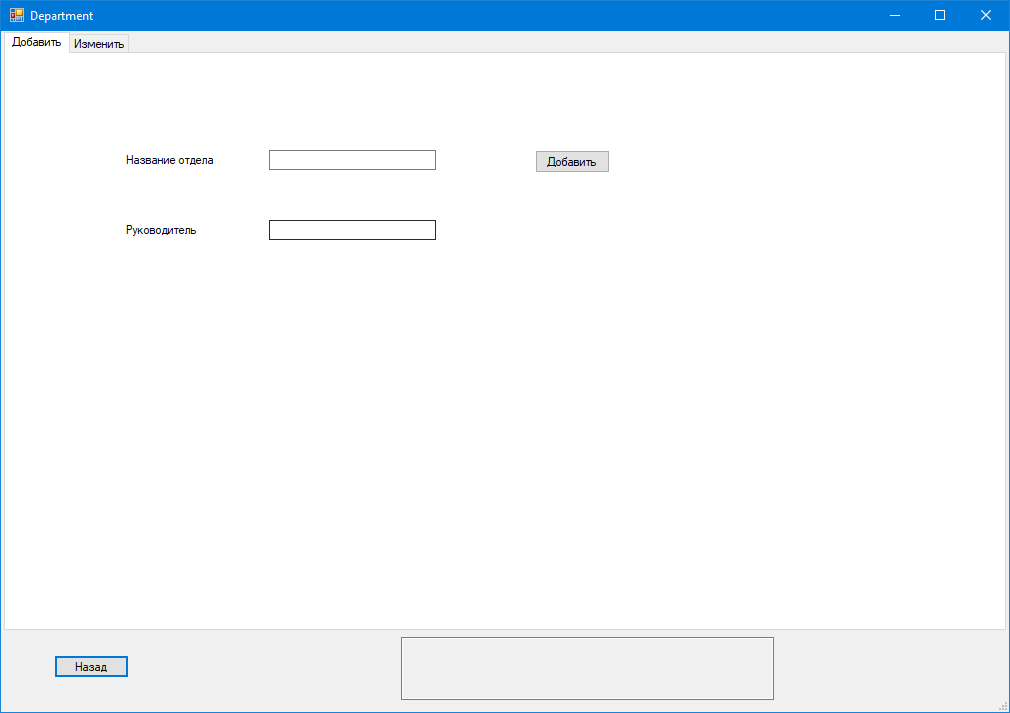


Рисунок 15 – Форма добавления отдела

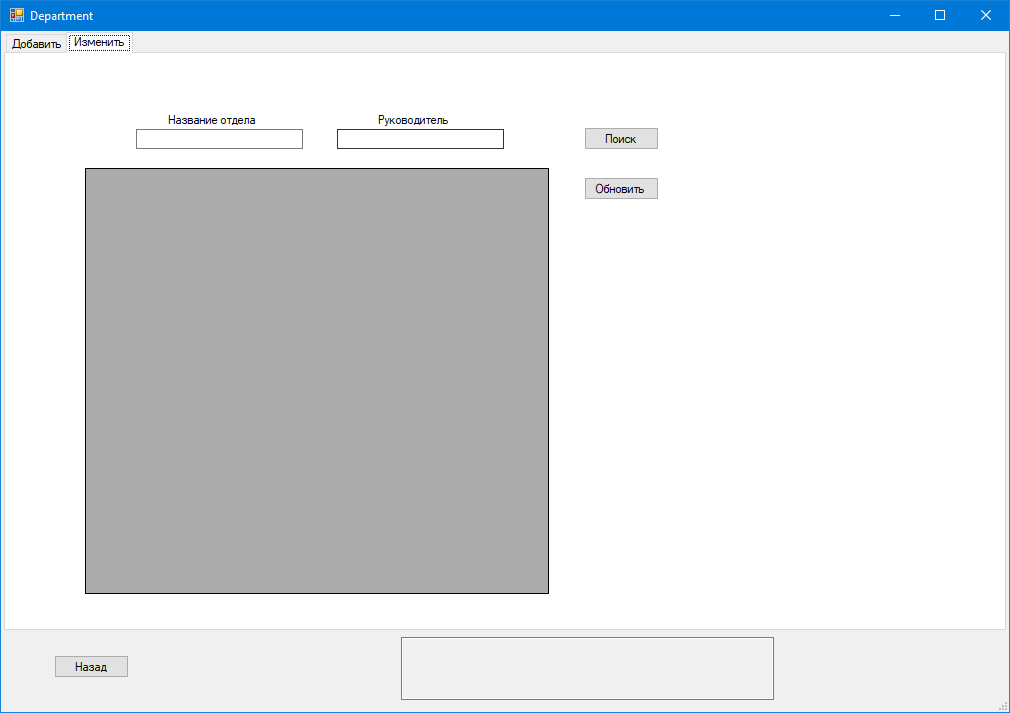


Рисунок 16 – Форма редактирования/поиска отдела

Код формы:

using System;

using System.Windows.Forms;

namespace dbtest

{

public partial class Positions : Form

{

public Positions()

{

InitializeComponent();

}

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

{

Positions.ActiveForm.Hide();

Main Main = new Main();

Main.ShowDialog();

Close();

}

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

{

string sql = String.Format("SELECT \* FROM Position WHERE position LIKE '{0}' AND salary LIKE '{1}'", textBoxPosition.Text + "%", textBoxSalary.Text + "%");

dataGridView1.DataSource = db.GetTable(sql);

textBoxLog.Text = db.Log;

}

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

{

string[] arr = { textBoxPos.Text, textBoxSal.Text };

if (db.valid(arr)) {

string sql = String.Format("INSERT INTO Position Values('{0}', '{1}')", textBoxPos.Text, textBoxSal.Text);

db.InsertGetTable(sql);

textBoxLog.Text = db.Log;

}

else textBoxLog.Text = db.Log;

}

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

{

db.UpdateGetTable((DataTable)dataGridView1.DataSource);

}

}

}

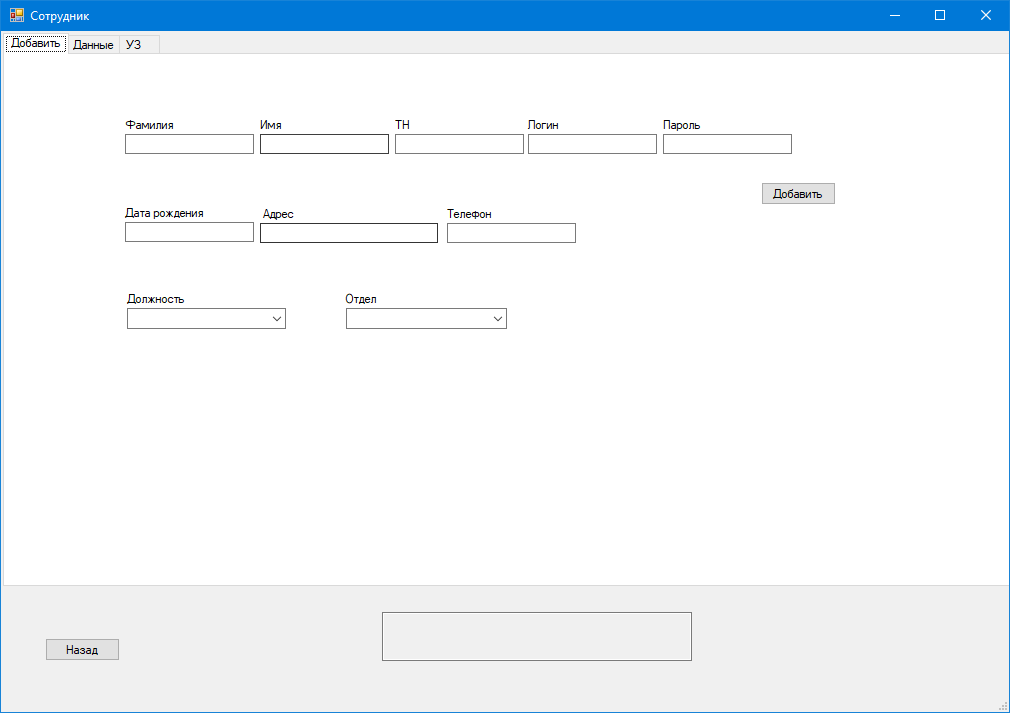


Рисунок 17 – Форма добавления сотрудника

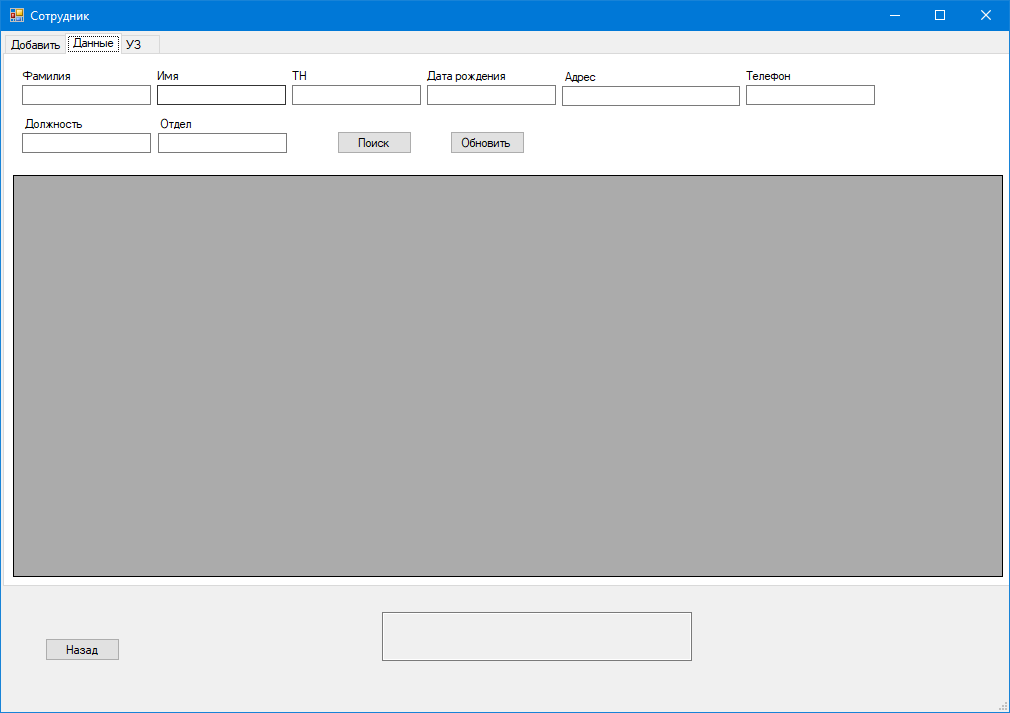


Рисунок 18 – Форма редактирования/поиска сотрудника

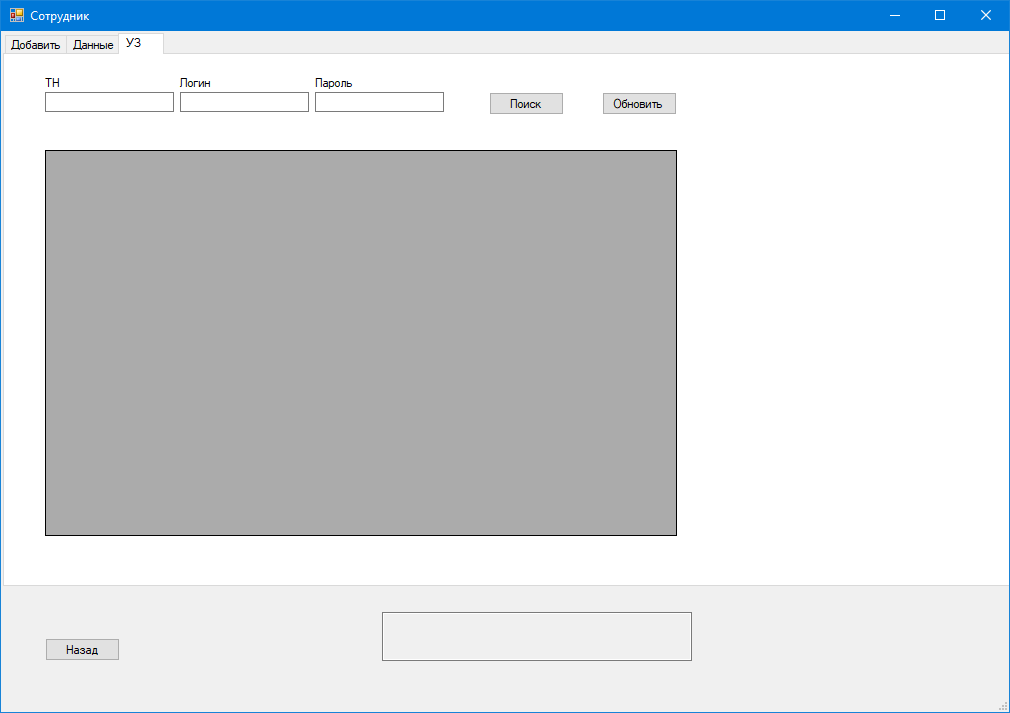


Рисунок 19 – Форма редактирования/поиска УЗ сотрудника

Код формы:

using System;

using System.Windows.Forms;

namespace dbtest

{

public partial class Employees : Form

{

public Employees()

{

InitializeComponent();

DataTable departments = db.GetTable("Select department From Department");

DataTable positions = db.GetTable("Select position From Position");

foreach (DataRow dr in departments.Rows) {

comboBoxDepartment.Items.Add(dr["department"].ToString());

}

foreach (DataRow dr in positions.Rows)

{

comboBoxPosition.Items.Add(dr["position"].ToString());

}

}

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

{

Employees.ActiveForm.Hide();

Main Main = new Main();

Main.ShowDialog();

Close();

}

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

{

string[] arr = {textBoxLastName.Text, textBoxFirstName.Text, textBoxTablNumber.Text, textBoxLogin.Text, textBoxPassword.Text, textBoxBirthday.Text, textBoxAddress.Text, textBoxPhone.Text, comboBoxPosition.Text, comboBoxDepartment.Text

};

if (db.valid(arr))

{

string str;

string sql = String.Format("INSERT INTO Employees Values('{0}', '{1}', '{2}', '{3}', '{4}', '{5}', '{6}', '{7}')", textBoxTablNumber.Text, textBoxFirstName.Text, textBoxLastName.Text, textBoxBirthday.Text, textBoxPhone.Text, textBoxAddress.Text, comboBoxPosition.Text, comboBoxDepartment.Text);

db.InsertGetTable(sql);

str = db.Log;

if (str == "Данные изменены") {

sql = String.Format("INSERT INTO LogPass Values('{0}', '{1}', '{2}')", textBoxTablNumber.Text, textBoxLogin.Text, textBoxPassword.Text);

db.InsertGetTable(sql);

str += Environment.NewLine + db.Log;

}

textBoxLog.Text = str;

}

else textBoxLog.Text = db.Log;

}

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

{

string sql = String.Format("SELECT \* FROM Employees WHERE tabl LIKE '{0}' AND first\_name LIKE '{1}' AND last\_name LIKE '{2}' AND birthday LIKE '{3}' AND phone LIKE '{4}' AND address LIKE '{5}' AND position LIKE '{6}' AND department LIKE '{7}'", textBoxTN.Text + "%", textBoxFN.Text + "%", textBoxLN.Text + "%", textBoxB.Text + "%", textBoxPh.Text + "%", textBoxA.Text + "%", textBoxPos.Text + "%", textBoxD.Text + "%");

dataGridView1.DataSource = db.GetTable(sql);

textBoxLog.Text = db.Log;

}

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

{

db.UpdateGetTable((DataTable)dataGridView1.DataSource);

textBoxLog.Text = db.Log;

}

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

{

Employees.ActiveForm.Hide();

Main Main = new Main();

Main.ShowDialog();

Close();

}

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

{

string sql = String.Format("SELECT \* FROM LogPass WHERE tabl LIKE '{0}' AND login LIKE '{1}' AND password LIKE '{2}'", textBoxTN.Text + "%", textBoxLogin.Text + "%", textBoxPassword.Text + "%");

dataGridView2.DataSource = db.GetTable(sql);

textBoxLog.Text = db.Log;

}

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

{

db.UpdateGetTable((DataTable)dataGridView2.DataSource);

textBoxLog.Text = db.Log;

}

}

}

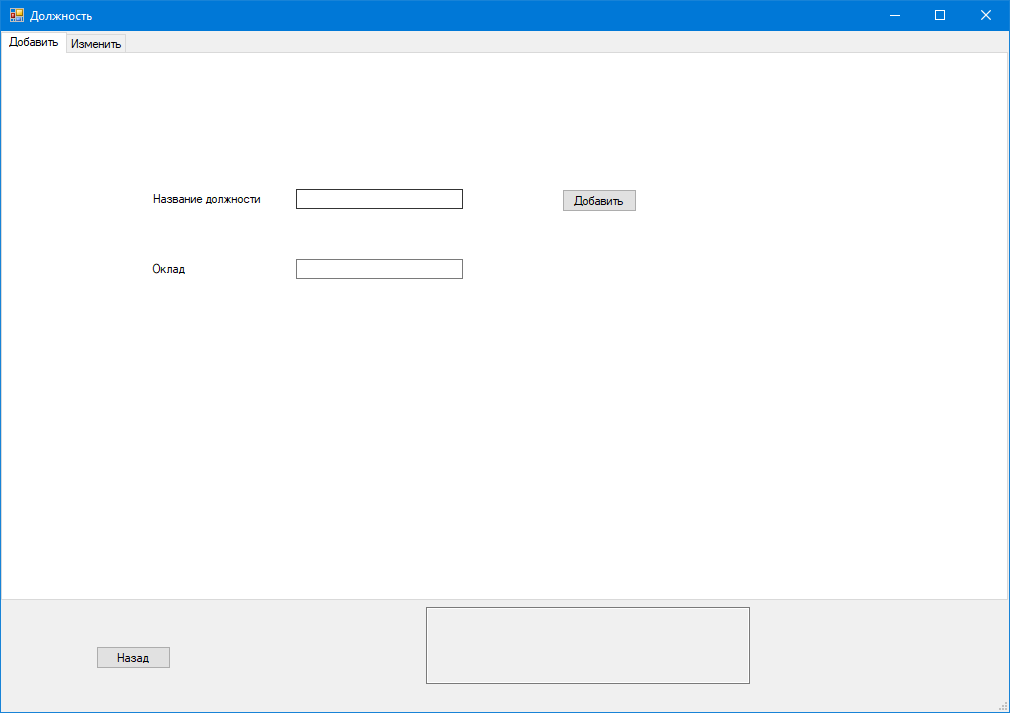


Рисунок 20 – Форма добавления должности

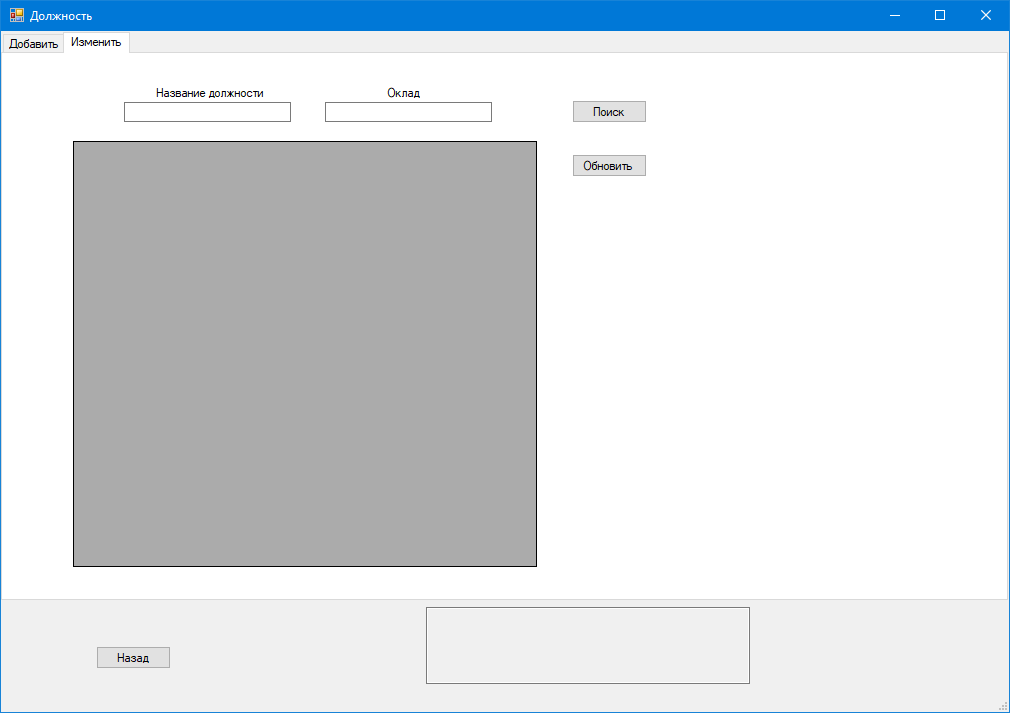


Рисунок 21 – Форма редактирования/поиска должности

Код формы:

using System;

using System.Windows.Forms;

namespace dbtest

{

public partial class Positions : Form

{

public Positions()

{

InitializeComponent();

}

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

{

Positions.ActiveForm.Hide();

Main Main = new Main();

Main.ShowDialog();

Close();

}

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

{

string sql = String.Format("SELECT \* FROM Position WHERE position LIKE '{0}' AND salary LIKE '{1}'", textBoxPosition.Text + "%", textBoxSalary.Text + "%");

dataGridView1.DataSource = db.GetTable(sql);

textBoxLog.Text = db.Log;

}

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

{

string[] arr = { textBoxPos.Text, textBoxSal.Text };

if (db.valid(arr)) {

string sql = String.Format("INSERT INTO Position Values('{0}', '{1}')", textBoxPos.Text, textBoxSal.Text);

db.InsertGetTable(sql);

textBoxLog.Text = db.Log;

}

else textBoxLog.Text = db.Log;

}

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

{

db.UpdateGetTable((DataTable)dataGridView1.DataSource);

}

}

}

Логика работы с базы данных:

using System;

using System.Data;

namespace dbtest

{

public static class db

{

public static string Log = "";

static string connectionString = @"Data Source=LAPTOP-CVVJT58B;Initial Catalog=HRD;Integrated Security=True";

public static SqlDataAdapter dataAdapter;

static SqlConnection connection = new SqlConnection(connectionString);

public static DataTable GetTable(string sql)

{

try

{

connection.Close();

DataTable dataTable = new DataTable();

connection.Open();

SqlCommand sqlCommand = new SqlCommand(sql, connection);

dataAdapter = new SqlDataAdapter(sqlCommand);

SqlCommandBuilder commandBuilder = new SqlCommandBuilder(dataAdapter);

dataAdapter.Fill(dataTable);

connection.Close();

Log = "Запрос выполнен\n";

return dataTable;

}

catch (SqlException ex)

{

Log = "Error: " + ex.Message + "\n";

return new DataTable();

}

}

public static void UpdateGetTable(DataTable DataSource)

{

try // Пытаемся записать изменения в SQL Server

{ // открываем соединение

connection.Close();

connection.Open();

dataAdapter.Update(DataSource);

connection.Close();

Log = "Данные изменены\n";

}

catch (Exception ex) // при ошибке выводим сообщение

{

Log = "Update Error" + ex.Message + "\n";

}

}

public static void InsertGetTable(string sql)

{

try // пытаемся записать новую строку

{

connection.Close();

connection.Open();

SqlCommand sqlCommand = new SqlCommand(sql,connection);

sqlCommand.ExecuteNonQuery();

connection.Close();

Log = "Данные изменены\n";

}

catch (Exception ex) // при ошибке записи новой строки выводим сообщение

{

Log = "Insert Error: " + ex.Message + " \n";

}

}

public static bool valid(string[] arr) {

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

if (arr[i] == "") {

Log = "Заполните все поля";

return false;

}

}

return true;

}

}

}

# Заключение

В процессе реализации курсового проекта была достигнута цель – разработки, приложения «Отдел кадров» для автоматизации учета сотрудников предприятия.

Данная система отвечает поставленным целям: простой пользовательский интерфейс, отчет документации, функции добавления, редактирования и поиска необходимых данных.

Процесс создания системы производится согласно разделам пояснительной записки: были проанализированы и изучены поставленные цели: проанализирована предметная область, выполнена постановка задачи и концептуальное проектирование, разработана технологическая цепочка обработки информации, были обоснованы выбранные методы решения создания приложения «Отдел кадров», построены логическая и физическая модели данных, создание программно-информационного компонента.

Список используемых источников

1. Голицына О.Л., Максимов H.B., Попов И.И. Базы данных: Учебное пособие. – М.: ФОРУМ: ИНФРА-М, 2005. – 420 с.
2. Кренке Д. Теория и практика построения баз данных. - 9 - е изд. – СПб.: Питер, 2005. - 858 с.
3. Диго С.М. Базы данных – М.: Финансы и статистика, 2005. – 456 с.
4. Марков А.С., Лисовский К.Ю. Базы данных. Введение в теорию и методологию: учебник. – М.: Финансы и статистика, 2004. – 536 с.
5. Когаловский М.Р. Энциклопедия технологий баз данных. – М.: Фи­нансы и статистика, 2002. – 520 с.
6. Базы данных. Учебник для высших заведений /Под ред. проф. А.Д. Хомоненко. – СПб.: Корона принт, 2000. – 416 с.
7. Хансен Г., Хансен Д. Базы данных: разработка и управление. – М.: Бином, 1999. – 704 с.
8. Дэйт К. Введение в системы баз данных. – М.: Наука, 1998. – 630 с.
9. Грабер М. Введение в SQL. – М.: ЛОРИ, 1996. – 360 с.

# Приложение А – Антиплагиат