

Министерство науки и высшего образования Российской Федерации
ФГБОУ ВО «Кубанский государственный технологический университет»
(ФГБОУ ВО «КубГТУ»)

Многоотраслевой институт подготовки и переподготовки специалистов
(МИППС)

Факультет: по работе со студентами ускоренного обучения по
индивидуальным учебным планам
Кафедра: информационных систем и программирования
Направление подготовки: 09.03.04 Программная инженерия
Профиль: беспрофильный

КУРСОВОЙ ПРОЕКТ

по дисциплине: Базы данных

на тему: Автоматизированная информационная система «Отдел кадров»

Выполнил студент 3 курса группы 17-ЗКБс-ПР1 Бабич М.М.

Допущен к защите _____

Руководители работы: _____ зав.кафедры. М.В. Янаева

Нормоконтролер _____ зав.кафедры. М.В. Янаева

Защищен _____ Оценка _____

Члены комиссии: _____ ст. преп А.Г. Мурлин

_____ ст. преп А.Г. Волик

Краснодар
2020 г.

Министерство науки и высшего образования Российской Федерации
ФГБОУ ВО «Кубанский государственный технологический университет»
(ФГБОУ ВО «КубГТУ»)

Многоотраслевой институт подготовки и переподготовки специалистов
(МИППС)

Факультет: по работе со студентами ускоренного обучения по
индивидуальным учебным планам
Кафедра: информационных систем и программирования
Направление подготовки: 09.03.04 Программная инженерия
Профиль: беспрофильный

УТВЕРЖДАЮ
Заведующая кафедрой ИСП
канд. техн. наук, доц. М.В. Янаева

«_____» _____ 20__ г.

ЗАДАНИЕ

на курсовой проект

Студенту Бабич М.М. 3 курса группы 17-ЗКБс-ПР1
Тема проекта: Автоматизированная информационная система «Отдел кадров»
(утверждена указанием директора института № 10 от 11.02.2020г.)
План проекта: 1. Описание предметной области
2. Концептуальная модель базы данных
3. Физическая модель базы данных
4. Реализация приложения

Объем проекта:

- а) пояснительная записка 26 с.
б) иллюстрированная часть 13 листов

Рекомендуемая литература: Голицына О.Л., Максимов Н.В., Попов И.И. Базы данных: Учебное пособие. – М.: ФОРУМ: ИНФРА-М, 2015. – 420 с.

Срок выполнения работы: с «11» 02 2020 г. по «25» 05 2020 г.
Срок защиты: «10» 06 2020 г.
Дата выдачи задания: «11» 02 2020 г.
Дата сдачи работы на кафедру: «25» 05 2020 г.

Руководители работы: _____ зав. кафедры М.В. Янаева

Задание принял студент «11» 02 2020 г.  Бабич М.М.

Реферат

Пояснительная записка курсового проекта 26 с., 21 рис., 5 источников.

БАЗЫ ДАННЫХ, ОТДЕЛ КАДРОВ, СИСТЕМЫ УПРАВЛЕНИЯ
БАЗАМИ ДАННЫХ, ОБЪЕКТНО-РЕЛЯЦИОННАЯ СИСТЕМА
УПРАВЛЕНИЯ БАЗАМИ ДАННЫХ, MYSQL.

Цель работы – создать информационную систему, упрощающую ввод, хранение, изменение, используемых для учета сотрудников для отдела кадров.

Задачей курсового проекта является разработка информационной систем обладающей простым и удобным интерфейсом, при помощи, которой возможно просмотреть информацию необходимую для учета отдела кадров.

В результате работы была разработана база данных для приложения «Отдел кадров». Это позволяет упростить процесс поиска, редактирования, добавления информации сотрудников организации.

Содержание

Введение.....	5
1 Анализ предметной области	6
2 Анализ методов решения.....	7
3 Информационная модель системы	8
3.1 Концептуальная модель.....	8
3.1 Логическая модель	8
3.2 Физическая модель	10
4 Разработка программно-информационных компонентов системы	11
4.1 Разработка пользовательского интерфейса.....	11
4.2 Описание программных модулей.....	12
Заключение	24
Список используемых источников.....	25
Приложение А – Отчет о проверке на заимствования	26

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 передаче, 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

Объектами данной автоматизированной системы являются:

- 1) Сотрудник
- 2) Должность
- 3) Отдел

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.

3 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.

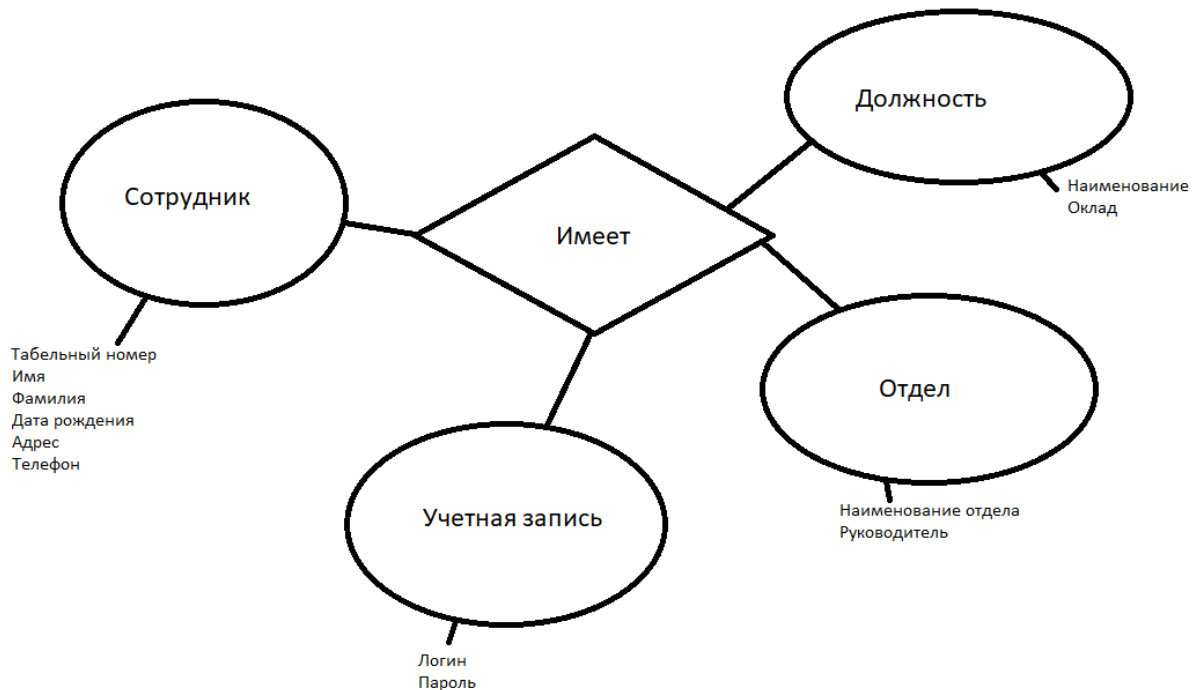


Рисунок 1 – Концептуальная модель

3.1 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

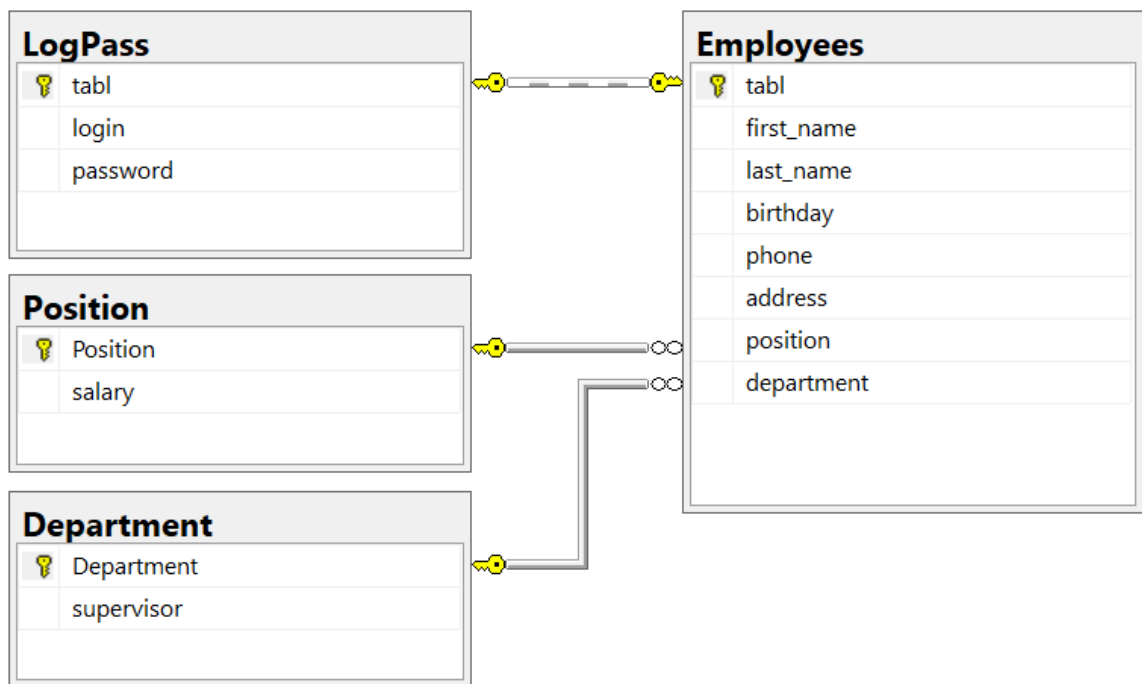


Рисунок 2 – Логическая модель

3.2 Физическая модель

Физическая модель представляет собой совокупность БД на материальном носителе.

	Имя столбца	Тип данных	Разрешить ...
►🔑	Department	nvarchar(50)	<input type="checkbox"/>
	supervisor	nvarchar(50)	<input type="checkbox"/>
			<input type="checkbox"/>

Рисунок 3 – Таблица отдела

	Имя столбца	Тип данных	Разрешить ...
►🔑	Position	nvarchar(50)	<input type="checkbox"/>
	salary	nvarchar(50)	<input type="checkbox"/>
			<input type="checkbox"/>

Рисунок 4 – Таблица должности

	Имя столбца	Тип данных	Разрешить ...
►🔑	tabl	int	<input type="checkbox"/>
	first_name	nvarchar(50)	<input type="checkbox"/>
	last_name	nvarchar(50)	<input type="checkbox"/>
	birthday	date	<input type="checkbox"/>
	phone	nvarchar(50)	<input type="checkbox"/>
	address	nvarchar(50)	<input type="checkbox"/>
	position	nvarchar(50)	<input type="checkbox"/>
	department	nvarchar(50)	<input type="checkbox"/>
			<input type="checkbox"/>

Рисунок 5 – Таблица сотрудников

	Имя столбца	Тип данных	Разрешить ...
►🔑	tabl	int	<input type="checkbox"/>
	login	nvarchar(50)	<input checked="" type="checkbox"/>
	password	nvarchar(50)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Рисунок 10 – Таблица учетной записи

4 Разработка программно-информационных компонентов системы

4.1 Разработка пользовательского интерфейса



Рисунок 12 – Макет формы «Меню»

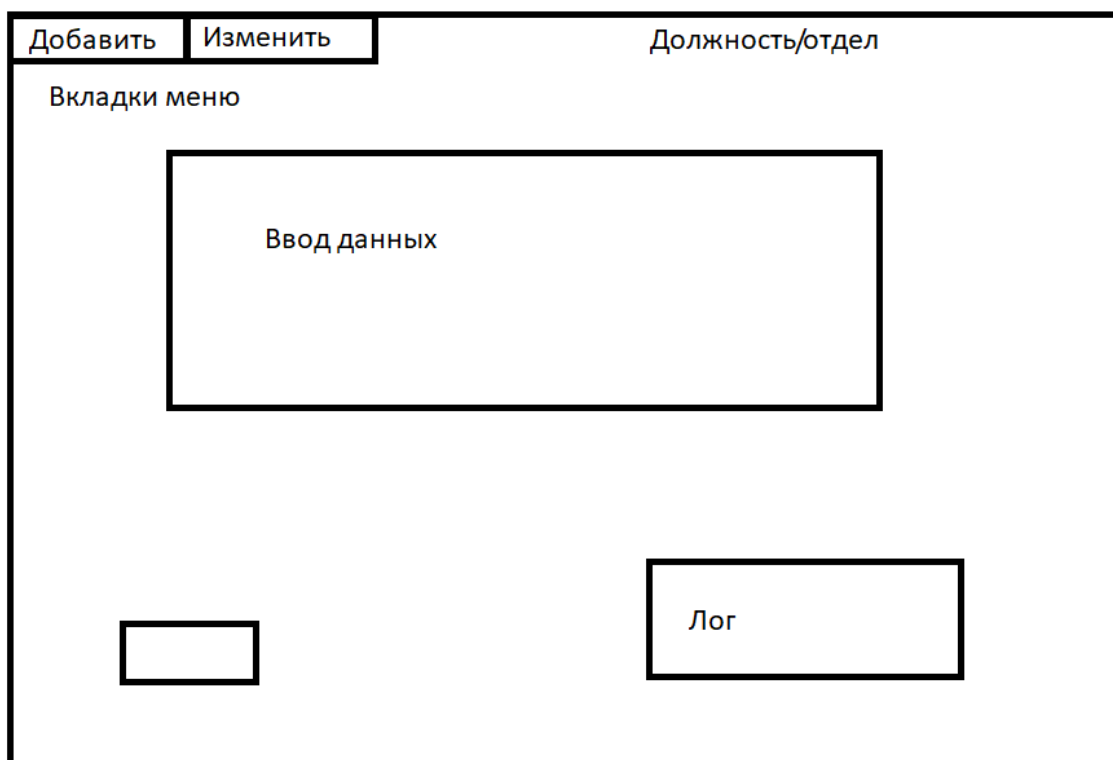


Рисунок 13 – Макет формы «Должность» и «Отдел»

The mockup shows a web form interface. At the top, there are three tabs: 'Добавить' (Add), 'Сотрудник' (Employee), and 'УЗ' (Department). The 'Сотрудник' tab is currently selected, and the text 'Сотрудник' is displayed to its right. Below the tabs, the text 'Вкладки меню' (Menu tabs) is visible. The main content area contains a large rectangular box labeled 'Ввод данных' (Data input). In the bottom left corner, there is a small, empty rectangular box. In the bottom right corner, there is a rectangular box labeled 'Лог' (Log).

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

4.2 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.

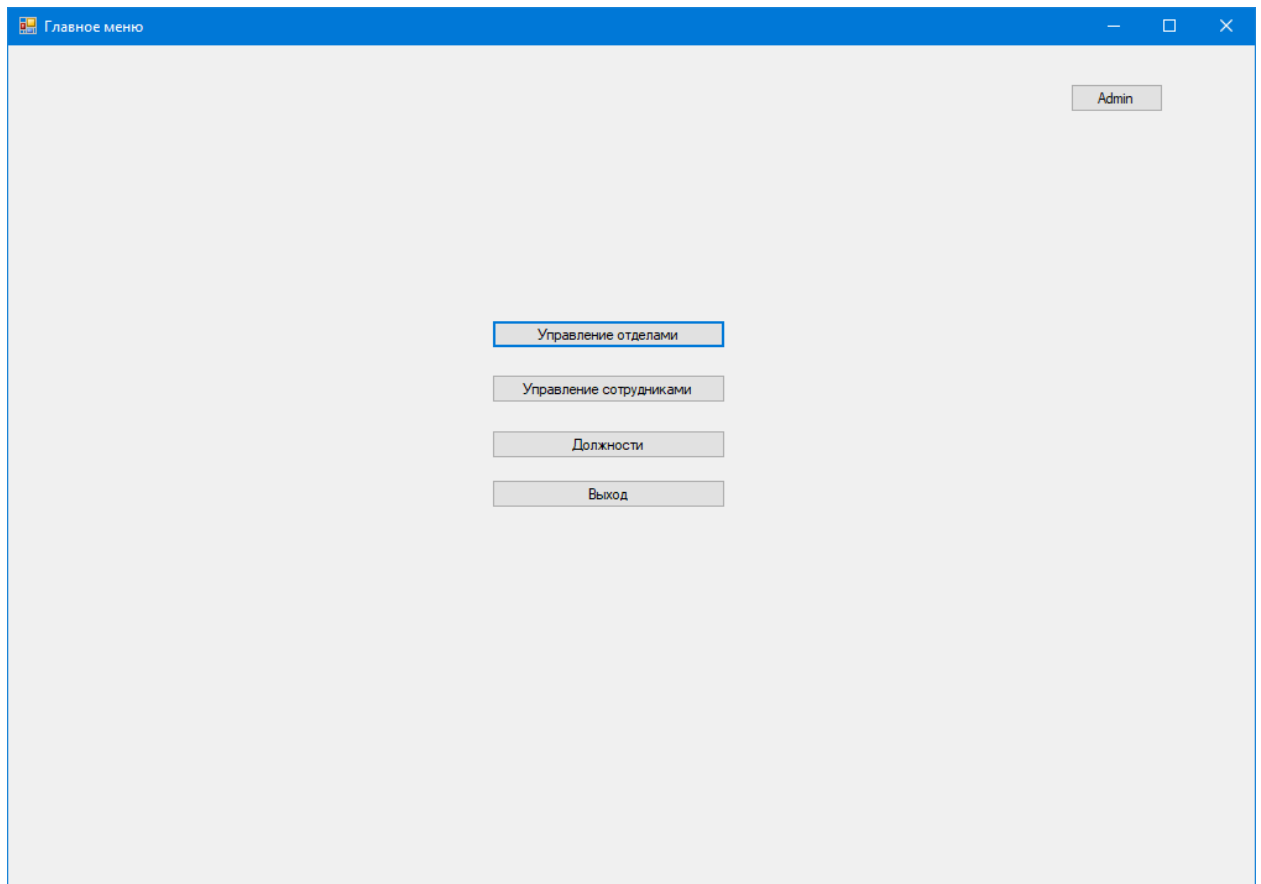


Рисунок 14 – Начальная форма «Menu»

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();
}
}
}

```

The screenshot shows a Windows application window titled "Department". It has a standard Windows interface with a title bar and window controls. Below the title bar, there are two tabs: "Добавить" (Add) and "Изменить" (Edit). The "Добавить" tab is currently selected. The main area of the window contains two text input fields. The first is labeled "Название отдела" (Department Name) and has a "Добавить" (Add) button to its right. The second is labeled "Руководитель" (Manager) and is empty. At the bottom of the window, there is a "Назад" (Back) button on the left and a large, empty text input field on the right.

Рисунок 15 – Форма добавления отдела

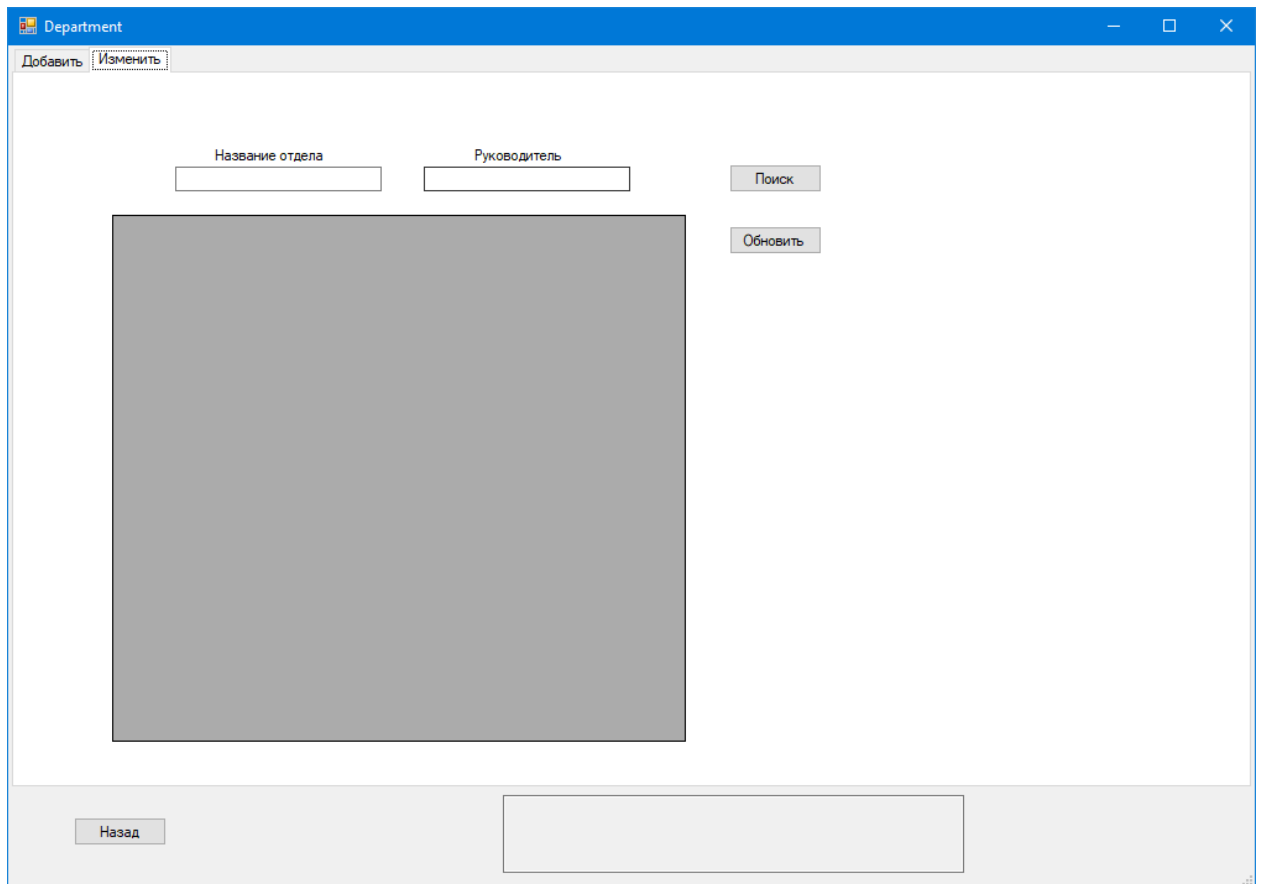


Рисунок 16 – Форма редактирования/поиска отдела

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);
}
}
}

```

Сотрудник

Добавить Данные УЗ

Фамилия Имя ТН Логин Пароль

Дата рождения Адрес Телефон

Должность Отдел

Добавить

Назад

Рисунок 17 – Форма добавления сотрудника

Сотрудник

Добавить Данные УЗ

Фамилия Имя ТН Дата рождения Адрес Телефон

Должность Отдел Поиск Обновить

Назад

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;
    }
}

```

The screenshot shows a Windows application window titled "Должность" (Position). The window has a blue title bar with standard Windows controls (minimize, maximize, close). Below the title bar is a menu bar with two items: "Добавить" (Add) and "Изменить" (Edit). The main content area is white and contains two input fields. The first field is labeled "Название должности" (Position Name) and has a "Добавить" (Add) button to its right. The second field is labeled "Оклад" (Salary). At the bottom of the window, there is a gray bar containing a "Назад" (Back) button on the left and a large, empty rectangular area on the right.

Рисунок 20 – Форма добавления должности

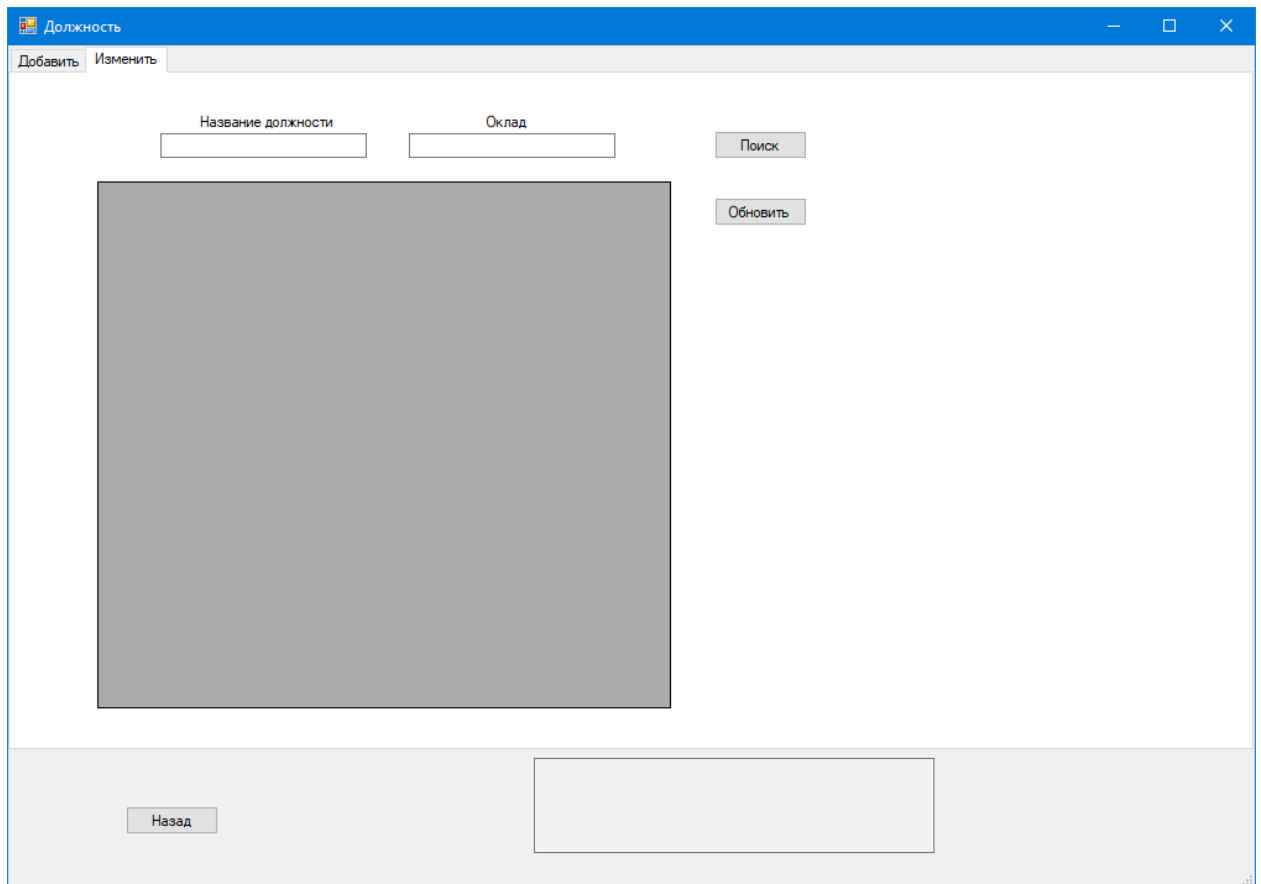


Рисунок 21 – Форма редактирования/поиска должности

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-CVWJT58B;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, автоматизированной информационной системы «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 автоматизированной информационной системы «Отдел кадров», построены логическая и физическая модели данных, создание программно-информационного компонента.

Список используемых источников

1 Янаева М. В., Мурлин А. Г., Мурлина В. А., «Запросы SQL в примерах: учеб. Пособие» / Кубан. Гос. Технол. ун-т. – Краснодар: Изд. ФГБОУ ВПО «КубГТУ», 2013.- 114 с.

2 Янаева М. В., Мурлин А. Г., Мурлина В. А., «Базы данных: краткие конспекты лекций по изучению дисциплины для студентов всех форм обучения и МИППС направления 090303.62 Прикладная информатика»/ Сост. Кубан. гос. технол. ун-т. Кафедра информационных систем и программирования. – Краснодар: Изд. КубГТУ, 2015. – 31 с.

3 Джеймс Р. Грофф, Пол Н. Вайнберг, Эндрю Дж. Оппель, «SQL. Полное руководство», Вильямс, 2015. - 960 с.

4 Бейли Линн, «Изучаем SQL», Питер, 2015 - 592с.

5 Эрик Редмонд, Джим Р. Уилсон, «Семь баз данных за семь недель. Введение в современные базы данных и идеологию NoSQL», Питер, 2015 - 358с.

Приложение А – Отчет о проверке на заимствования



Отчет о проверке на заимствования №1



Автор: mihaill2771@gmail.com / ID: 6317881
Проверяющий: mihaill2771@gmail.com / ID: 6317881
Отчет предоставлен сервисом «Антиплагиат»- <http://users.antiplagiat.ru>

ИНФОРМАЦИЯ О ДОКУМЕНТЕ

№ документа: 28
Начало загрузки: 25.05.2020 03:05:56
Длительность загрузки: 00:00:01
Имя исходного файла: КП БД Бабиц М.М. 17-ЗКБс-003.pdf
Название документа: КП БД Бабиц М.М. 17-ЗКБс-003
Размер текста: 1 кБ
Символов в тексте: 15976
Слов в тексте: 1762
Число предложений: 230

ИНФОРМАЦИЯ ОБ ОТЧЕТЕ

Последний готовый отчет (ред.)
Начало проверки: 25.05.2020 03:05:57
Длительность проверки: 00:00:05
Комментарии: не указано
Модуль поиска: Модуль поиска Интернет



ЗАИМСТВОВАНИЯ
22,63%

САМОЦИТИРОВАНИЯ
0%

ЦИТИРОВАНИЯ
0%

ОРИГИНАЛЬНОСТЬ
77,37%

Заимствования — доля всех найденных текстовых пересечений, за исключением тех, которые система отнесла к цитированиям, по отношению к общему объему документа.
Самоцитирования — доля фрагментов текста проверяемого документа, совпадающий или почти совпадающий с фрагментом текста источника, автором или соавтором которого является автор проверяемого документа, по отношению к общему объему документа.
Цитирования — доля текстовых пересечений, которые не являются авторскими, но система посчитала их использование корректным, по отношению к общему объему документа. Сюда относятся оформленные по ГОСТу цитаты; общепотребительные выражения; фрагменты текста, найденные в источниках из коллекций нормативно-правовой документации.
Текстовое пересечение — фрагмент текста проверяемого документа, совпадающий или почти совпадающий с фрагментом текста источника.
Источник — документ, проиндексированный в системе и содержащийся в модуле поиска, по которому проводится проверка.
Оригинальность — доля фрагментов текста проверяемого документа, не обнаруженных ни в одном источнике, по которым шла проверка, по отношению к общему объему документа.
Заимствования, самоцитирования, цитирования и оригинальность являются отдельными показателями и в сумме дают 100% что соответствует всему тексту проверяемого документа.
Обращаем Ваше внимание, что система находит текстовые пересечения проверяемого документа с проиндексированными в системе текстовыми источниками. При этом система является вспомогательным инструментом, определение корректности и правомерности заимствований или цитирований, а также авторства текстовых фрагментов проверяемого документа остается в компетенции проверяющего.

№	Доля в отчете	Доля в тексте	Источник	Ссылка	Актуален на	Модуль поиска	Блоков в отчете	Блоков в тексте
[01]	6,25%	8,81%	Среда разработки Visual Studio	http://referatya.ru	30 Мая 2018	Модуль поиска Интернет	2	4
[02]	0%	8,73%	2017_ФИСИ_09.03.01_ПОУТС_БР_Перцев_Сергей_Алекс.>	https://psuci.ru	10 Авг 2018	Модуль поиска Интернет	0	5
[03]	0%	8,73%	Разработка игрового обучающего приложения для Andr...	http://worldreferat.ru	27 Окт 2018	Модуль поиска Интернет	0	5

Еще источников: 17
Еще заимствований: 16,38%