**ФЕДЕРАЛЬНОЕ АГЕНТСТВО СВЯЗИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТЕЛЕКОММУНИКАЦИЙ ИМ. ПРОФ. М.А. БОНЧ-БРУЕВИЧА»**

**(СПбГУТ)**

Факультет «Институт магистратуры»

Кафедра «Автоматизации предприятий связи»

|  |  |
| --- | --- |
| Направление подготовки: | 27.04.04 - Управление в технических системах |
| Направленность (профиль): | Цифровые системы управления |

**КУРСОВОЙ ПРОЕКТ**

по дисциплине:

|  |
| --- |
| **Технологии радиочастотной идентификации в автоматизированных системах мониторинга** |

на тему:

|  |
| --- |
| **Создание системы контроля и управления доступом** |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | |  | Выполнил студент группы | | ИСТ-051м | |
|  | |  | Гугнин А.И. | | | |
|  | |  | *Подпись, Фамилия И. О.* | | | |
|  |  | Проверил | к.т.н., доцент | |
| *отметка о зачете* |  |  | *уч. степень, уч. звание* | |
|  |  | Белоус К.В. | | |
| *дата, подпись* |  | *Фамилия И. О.* | | |

**ОГЛАВЛЕНИЕ**

Введение 3

Часть 1 Задание на проект4

* 1. Постановка задачи4
  2. Анализ требований к создаваемому приложению 5

Часть 2 Практическая реализация6

* 1. Исходный код сущностей (классов предметной области) 6
  2. Структура базы данных MSSQL.11

Часть 3 Практические результаты13

* 1. Демонстрация работы приложения13

Заключение17

# Введение

# СКУД – это система контроля и управления доступом. Данные системы являются одним из элементов комплексных решений для обеспечения высокого уровня безопасности. Принцип работы СКУД и алгоритмы позволяют строго контролировать любые перемещения в зоне действия. Сегодня эти устройства контроля можно встретить практически везде – в офисных зданиях, на объектах промышленности, и везде, где необходимо тщательно следить за поддержанием безопасности.

# СКУД позволяет снизить нагрузку на сотрудников службы охраны предприятия или офиса и значительно увеличивает эффективность работы этого подразделения. Охраннику больше не придется заниматься рутиной по идентификации – с этим оборудованием сотрудник службы безопасности сможет заниматься своими прямыми обязанностями.

# Любые системы для контроля и управления доступом позволяют:

# Ограничить проход рабочих или сотрудников, а также гостей в зоны с ограниченным доступом;

# Временно контролировать любые движения территории объекта;

# Осуществлять контроль за сотрудниками безопасности, если они находятся на дежурстве;

# Учитывать количество рабочего времени для каждого из работающих в офисе или на предприятии;

# Фиксировать время посещения и ухода посетителей;

# Осуществлять временные, а также персональный контроль всех внутренних входов и выходов – специалист службы охраны точно знает, кто и когда открыл ту или иную дверь;

# Регистрировать и выдавать отчеты о всех попытках доступа посторонних лиц на охраняемые объекты.

Часть 1 Задание на проект

1.1. Постановка задачи

Целью выполнения курсовой работы является разработка системы контроля доступа для предприятия, приобретение студентом навыков создания программных приложений. В ходе выполнения курсовой работы студент создает действующее приложение на языке C# с использование технологии Win Forms и сервера базы данных MS SQL Server.

Технология ASP.NET входит в состав платформы .NET и является одной из наиболее распространенных технологий создания распределенных приложений.

Приобретенные в процессе выполнения курсовой работы навыки по требуются при изучении последующих дисциплин: «Объектно- ориентированное программирование в АСУТП и АСУП», «Проектирование ERP систем», «Интегрированные системы проектирования и управления», а также при написании выпускной квалификационной работы. Кроме того, так как современная автоматизация технологических процессов и производств немыслима без информатизации, навыки создания программных приложений будут востребованы в будущей профессиональной деятельности студента.

1.2. Анализ требований к создаваемому приложению

Курсовая работа выполняется студентом индивидуально. Задания согласовываются с преподавателем и утверждаются.

Приложение, выполненное в рамках курсового проектирования, должно:

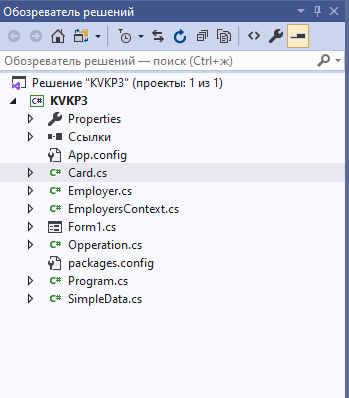
* разработать базу данных, позволяющую хранить информацию о сотрудниках и привязанных к ним картам. Необходимо учесть, что карта может сломаться/потеряться и др., т.е. необходимо предусмотреть механизм контроля «активности» карт и «архивации» неактивных карт;
* создать таблицу аудита операций входа/выхода по критерию «успех»/ «отказ»;
* реализовать функцию отчетов по входу/выходу сотрудника (всех или конкретного) за день, неделю, месяц или произвольный интервал. Аналогичную функцию реализовать для аудита.
* реализовать функцию отчёта по активным и утерянным картам (как по всем сотрудникам, так и по конкретной персоналии).
* Отчёты должны генерироваться в формате pdf. Можно воспользоваться библиотекой ITextSharp.
* Учесть, что на работу может быть принят новый сотрудник, а также то, что сотрудник может быть уволен.

Для написания приложения использовать IDE Microsoft Visual Studio, тип приложения – «Windows Forms», СУБД – MS SQL Server или MySQL.

**Часть 2 Практическая реализация**

* 1. **Исходный код сущностей (классов предметной области)**

Общее содержимое:

****

Далее будет представлены файлы, которые были изменены/добавлены

**Programm.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace KVKP3

{

static class Program

{

/// <summary>

/// Главная точка входа для приложения.

/// </summary>

[STAThread]

static void Main()

{

Application.EnableVisualStyles();

Application.SetCompatibleTextRenderingDefault(false);

Application.Run(new Form1());

}

}

}

**Form1.cs**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Data.OleDb;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using iTextSharp;

using iTextSharp.text;

using iTextSharp.text.pdf;

using System.IO;

using System.Data.Entity;

namespace KVKP3

{

public partial class Form1 : Form

{

//Определяем объект DataSet

DataSet MyDataSet = new DataSet();

//Имя каталога открываемого файла БД

string catName = "";

//Непосредственное имя самого файла БД

string fileName = "";

DataTable data1 = new DataTable();

DataTable data2 = new DataTable();

public Form1()

{

InitializeComponent();

}

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

{

using (EmployersContext db = new EmployersContext())

{

data1.Reset();

data1.Columns.Add("ID");

data1.Columns.Add("Name");

data1.Columns.Add("CardID");

data1.Columns.Add("CardActivated");

foreach (Employer emp in db.Employers.Include(emp=>emp.cards))

{

foreach(Card c in emp.cards)

{

DataRow dr1 = data1.NewRow();

dr1["ID"] = emp.EmployerId;

dr1["Name"] = emp.employer\_name;

dr1["CardID"] = c.CardId;

dr1["CardActivated"] = c.card\_activated;

data1.Rows.Add(dr1);

}

}

dataGridView1.DataSource = data1;

}

}

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

{

if (data1.Columns.Count>0) MyDataSet.Tables.Add(data1);

if (data2.Columns.Count > 0) MyDataSet.Tables.Add(data2);

//Объект документа пдф

iTextSharp.text.Document doc = new iTextSharp.text.Document();

//Создаем объект записи пдф-документа в файл

PdfWriter.GetInstance(doc, new FileStream("pdfTables.pdf", FileMode.Create));

//Открываем документ

doc.Open();

//Определение шрифта необходимо для сохранения кириллического текста

//Иначе мы не увидим кириллический текст

//Если мы работаем только с англоязычными текстами, то шрифт можно не указывать

BaseFont baseFont = BaseFont.CreateFont(@"C:\Windows\Fonts\Arial.ttf", BaseFont.IDENTITY\_H, BaseFont.NOT\_EMBEDDED);

iTextSharp.text.Font font = new iTextSharp.text.Font(baseFont, iTextSharp.text.Font.DEFAULTSIZE, iTextSharp.text.Font.NORMAL);

//Обход по всем таблицам датасета (хотя в данном случае мы можем опустить

//Так как в нашей бд только одна таблица)

for (int i = 0; i < MyDataSet.Tables.Count; i++)

{

//Создаем объект таблицы и передаем в нее число столбцов таблицы из нашего датасета

PdfPTable table = new PdfPTable(MyDataSet.Tables[i].Columns.Count);

//Добавим в таблицу общий заголовок

PdfPCell cell = new PdfPCell(new Phrase("БД " + fileName + ", таблица №" + (i + 1), font));

cell.Colspan = MyDataSet.Tables[i].Columns.Count;

cell.HorizontalAlignment = 1;

//Убираем границу первой ячейки, чтобы балы как заголовок

cell.Border = 0;

table.AddCell(cell);

//Сначала добавляем заголовки таблицы

for (int j = 0; j < MyDataSet.Tables[i].Columns.Count; j++)

{

cell = new PdfPCell(new Phrase(new Phrase(MyDataSet.Tables[i].Columns[j].ColumnName, font)));

//Фоновый цвет (необязательно, просто сделаем по красивее)

cell.BackgroundColor = iTextSharp.text.BaseColor.LIGHT\_GRAY;

table.AddCell(cell);

}

//Добавляем все остальные ячейки

for (int j = 0; j < MyDataSet.Tables[i].Rows.Count; j++)

{

for (int k = 0; k < MyDataSet.Tables[i].Columns.Count; k++)

{

table.AddCell(new Phrase(MyDataSet.Tables[i].Rows[j][k].ToString(), font));

}

}

//Добавляем таблицу в документ

doc.Add(table);

}

//Закрываем документ

doc.Close();

MessageBox.Show("Pdf-документ сохранен");

}

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

{

using (EmployersContext db = new EmployersContext())

{

data2.Reset();

data2.Columns.Add("CardID");

data2.Columns.Add("OpperationID");

data2.Columns.Add("Date");

data2.Columns.Add("Succefull");

foreach (Card c in db.Cards.Include(c => c.opperations))

{

foreach (Opperation o in c.opperations)

{

DataRow dr1 = data2.NewRow();

dr1["CardID"] = c.CardId;

dr1["OpperationID"] = o.OpperationId;

dr1["Date"] = o.use\_date;

dr1["Succefull"] = o.used\_successfull;

data2.Rows.Add(dr1);

}

}

dataGridView1.DataSource = data2;

}

}

}

}

**EmployerContext.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Data.Entity;

namespace KVKP3

{

class EmployersContext : DbContext

{

public EmployersContext() :

base("DBConnection")

{ }

public DbSet<Employer> Employers { get; set; }

public DbSet<Card> Cards { get; set; }

public DbSet<Opperation> Opperations { get; set; }

}}

**Employer.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace KVKP3

{

class Employer

{

public int EmployerId { get; set; }

public string employer\_name { get; set; }

public ICollection<Card> cards { get; set; }

public Employer()

{

cards = new List<Card>();

}

}

}

**Card.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace KVKP3

{

class Card

{

public int CardId { get; set; }

public bool card\_activated { get; set; }

public int EmployerId { get; set; }

public Employer Employer { get; set; }

public ICollection<Opperation> opperations { get; set; }

public Card()

{

opperations = new List<Opperation>();

}

}

}

**Opperation.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace KVKP3

{

class Opperation

{

public int OpperationId { get; set; }

public DateTime use\_date { get; set; }

public bool used\_successfull { get; set; }

public int CardId { get; set; }

public Card Card { get; set; }

}

}

* 1. **Структура базы данных MSSQL**

Разработанная схема базы данных на сайте diagram.io:

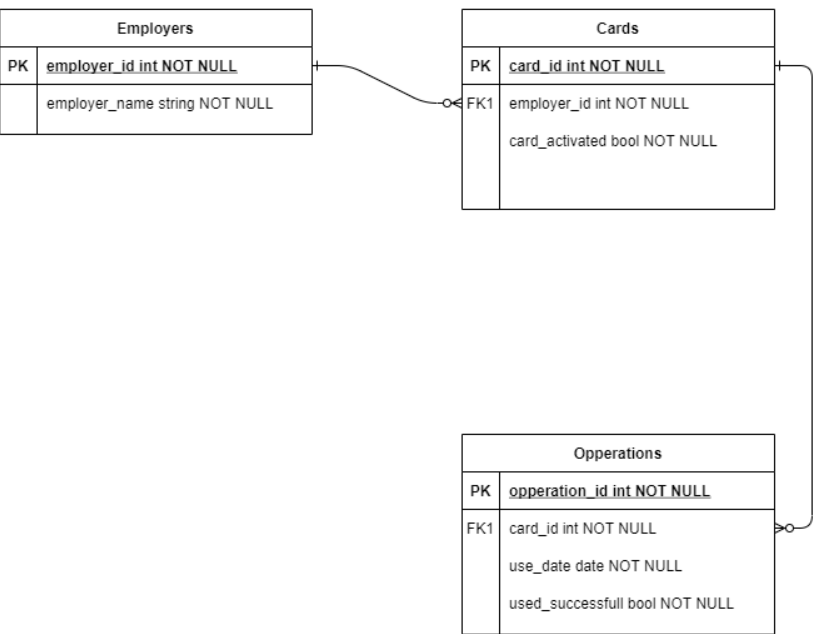
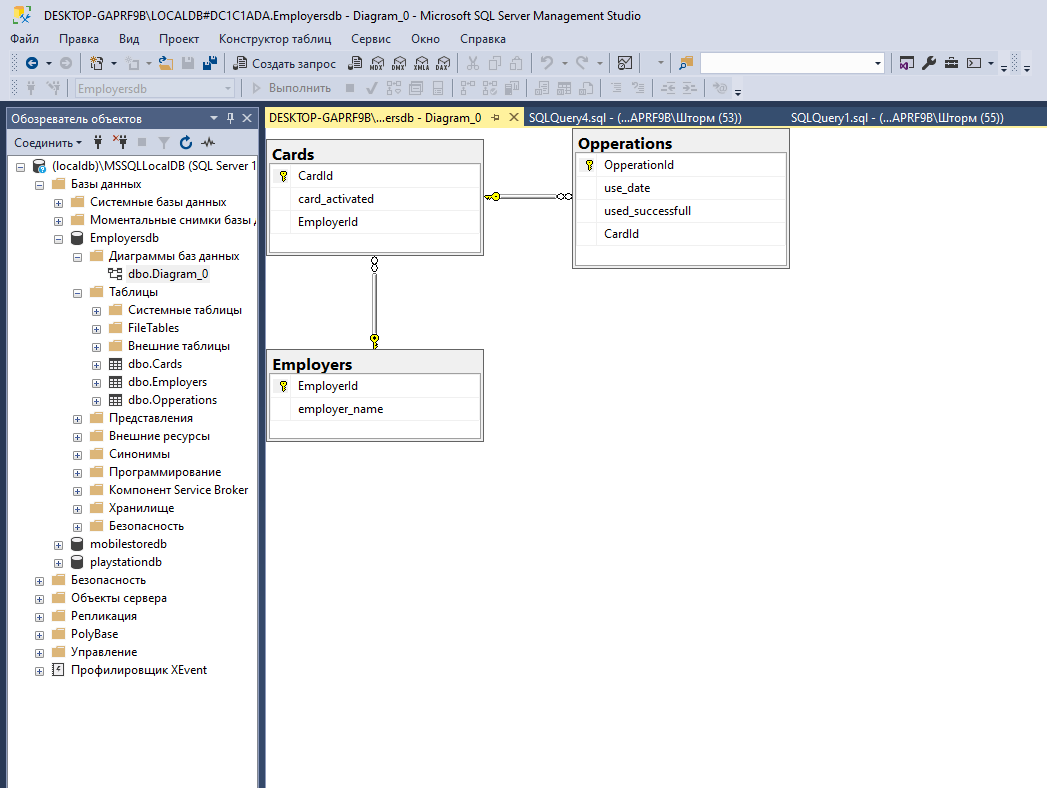
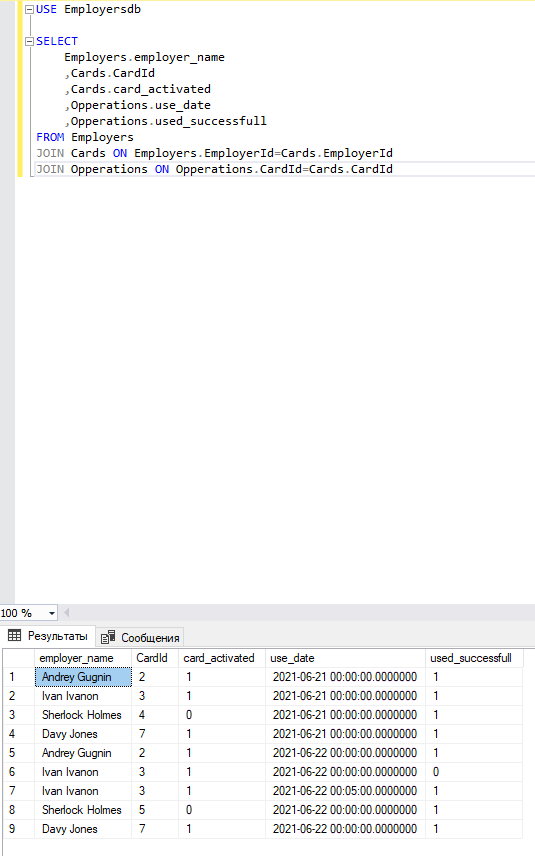


Диаграмма созданной базы данных в SSMS:



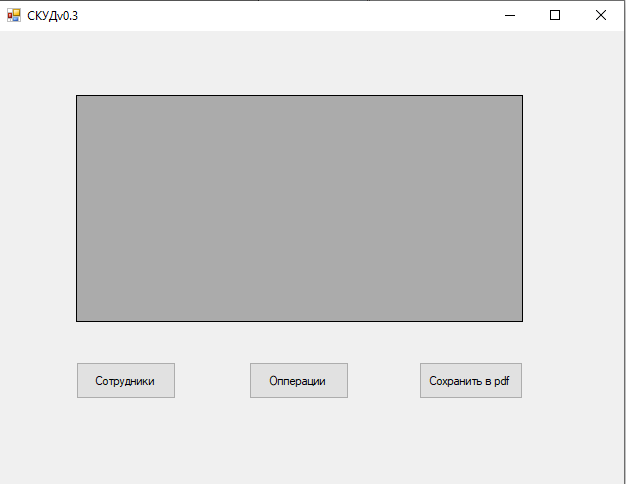
Вывод всех данных из заранее созданной базы данных запросом SELECT:



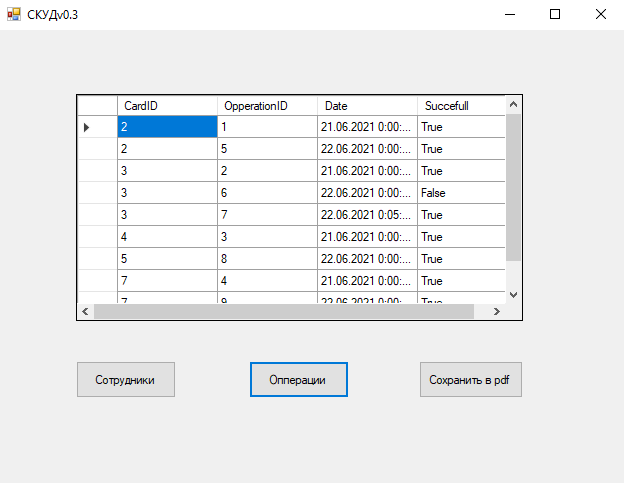
**Часть 3 Практическая реализация**

* 1. **Демонстрация работы приложения**

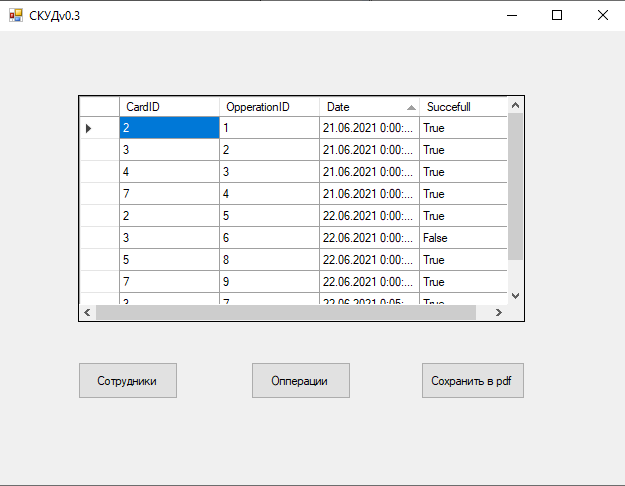
Интерфейс разработанного приложения с кнопками для вывода списка сотрудников предприятия, операций с картами в элемент dataGridView и сохранения отчета в pdf файл:



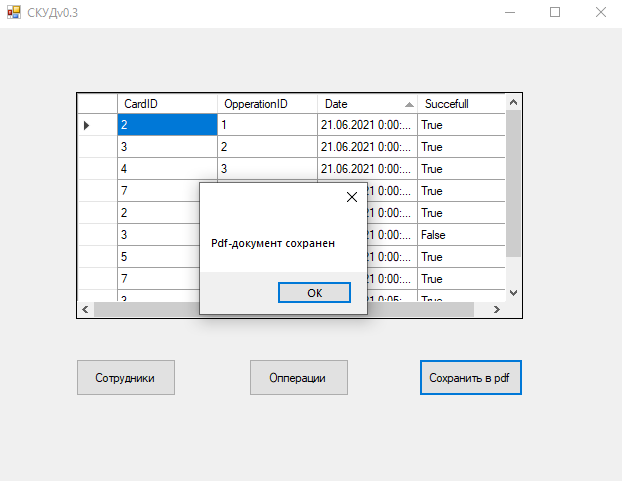
После нажатия на кнопку Операции выводится список операций, произведенных с картами из таблиц БД Operations и Cards:



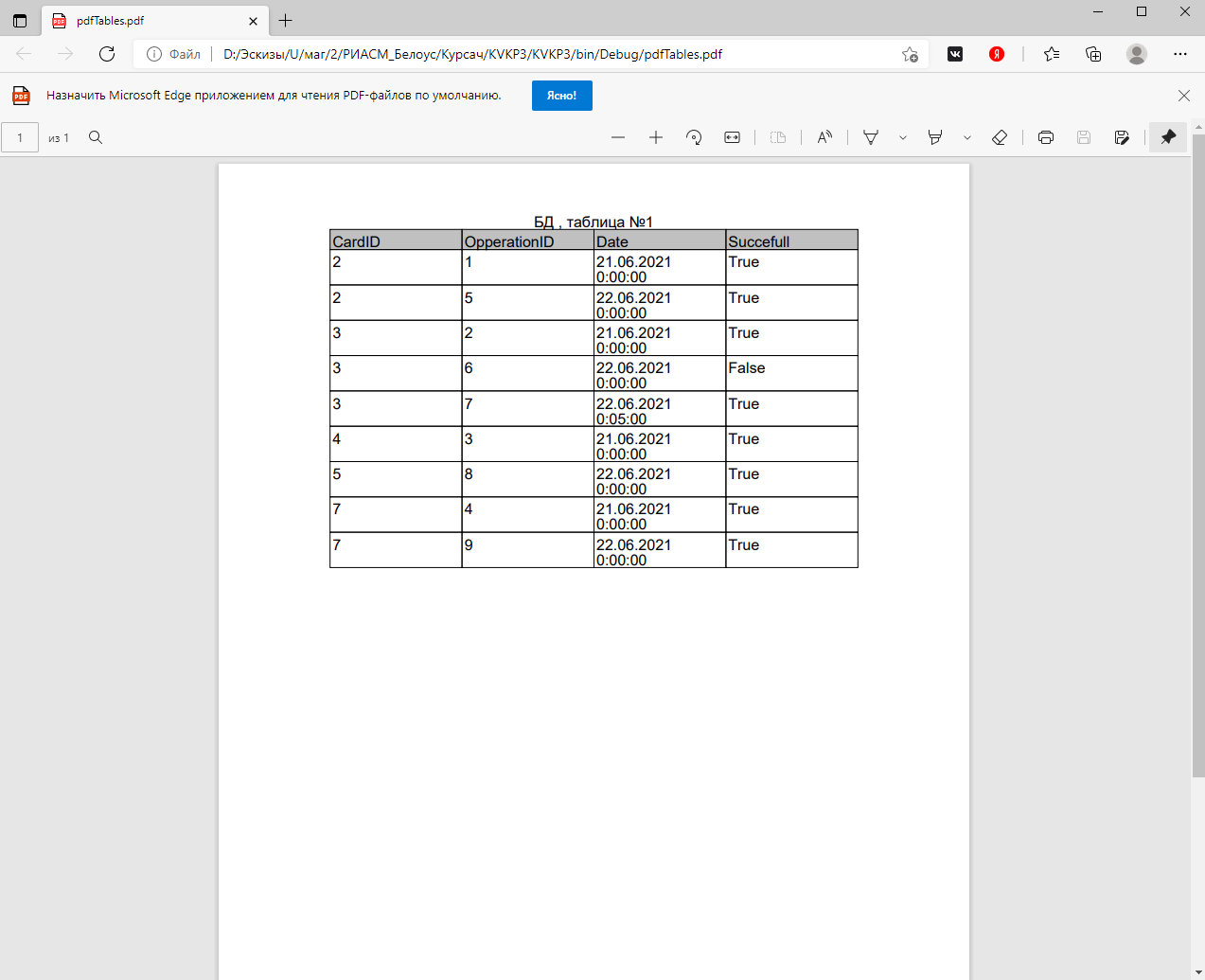
Список можно отсортировать по нужному столбцу, например, по дате:



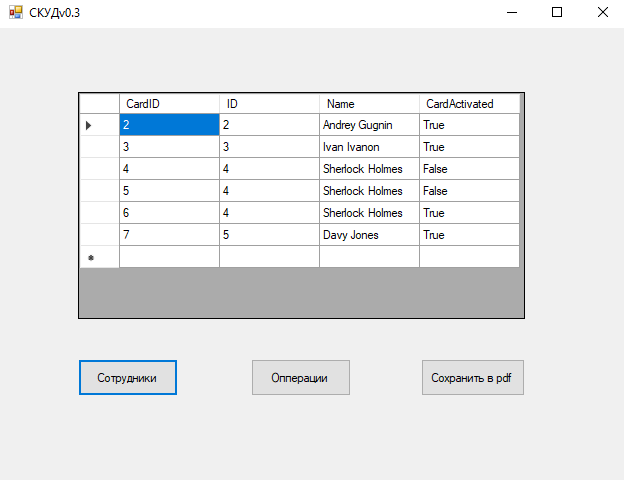
После нажатия на кнопку сохранения создается pdf файл:



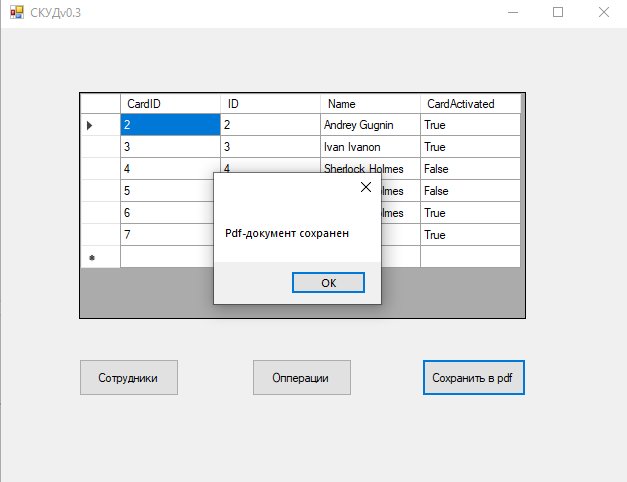
Содержимое pdf файла:



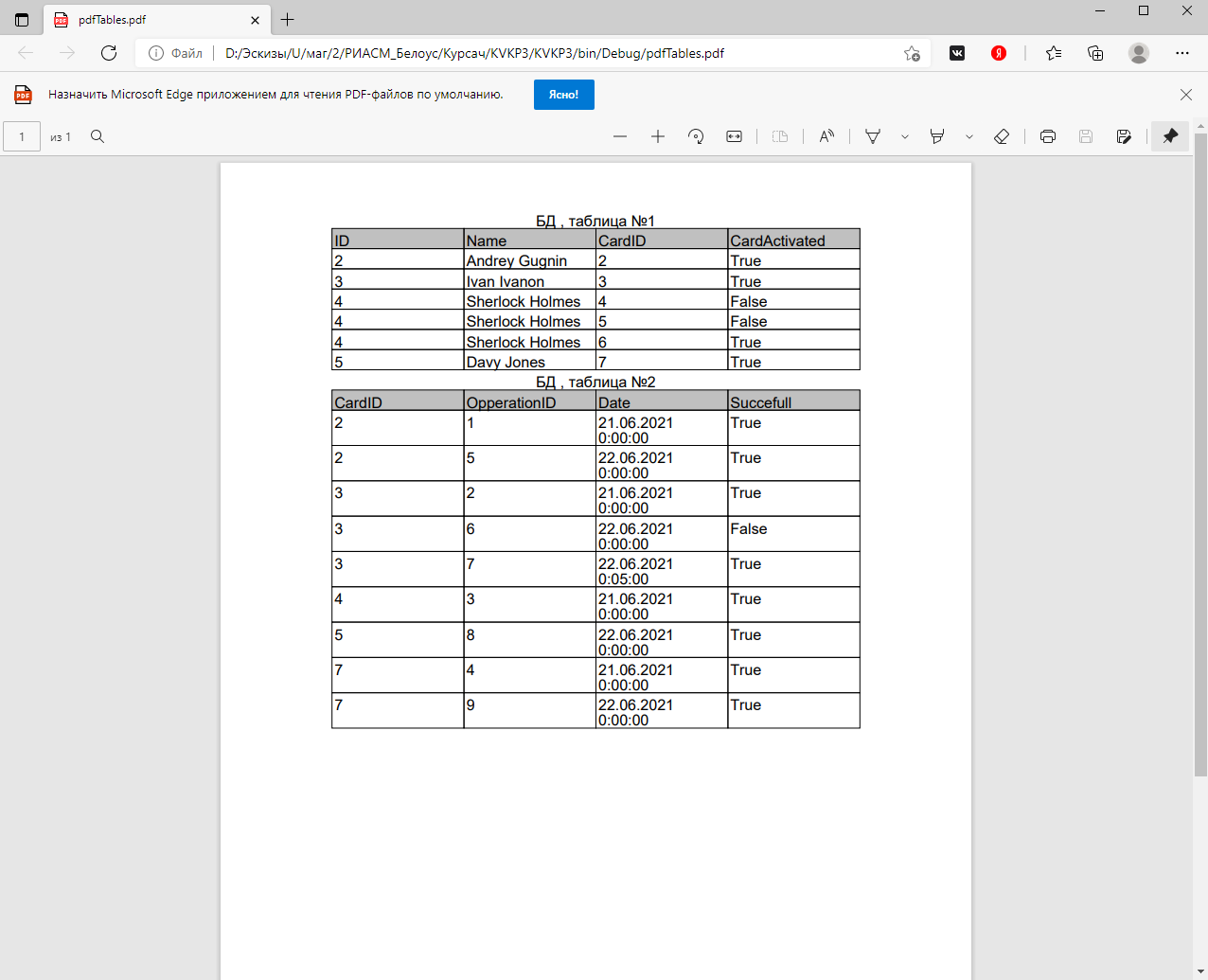
После нажатия кнопки «Сотрудники» выводиться соответствующий список:



Также реализована возможность вывода в файл нескольких таблиц:



Содержимое pdf файла:



# Заключение

В результате выполнения курсового проекта были выполнены следующие задачи:

* Разработано Приложение с помощью Windows forms.NET Framework в среде разработки Visual Studio 2019;
* Создана база данных при помощи технологии ADO.NET Entity Framework;
* Реализованы функции добавления, удаления и редактирования записей базы данных с выводом их на интерфейс приложения.

Разработанная система контроля управления доступом содержит в себе функции вывода основной информации о всех сотрудниках и операциях, сохранение отчетов в pdf файл.

