**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 2](#_Toc27470120)

[1.ТЕХНИЧЕСКОЕ ЗАДАНИЕ НА РАЗРАБОТКУ ИНФОРМАЦИОННОЙ СИСТЕМЫ 3](#_Toc27470121)

[2. ОБЩАЯ ЧАСТЬ 5](#_Toc27470122)

[2.1 Выбор и обоснование средств реализации БД 5](#_Toc27470123)

[2.2. Выбор и обоснование средств разработки программного продукта 5](#_Toc27470124)

[2.3 Анализ прототипов, существующих ИС 9](#_Toc27470125)

[3. СПЕЦИАЛЬНАЯ ЧАСТЬ 11](#_Toc27470126)

[3.1 Разработка алгоритма и проектирование структуры базы данных 11](#_Toc27470127)

[3.2 Разработка программного продукта 13](#_Toc27470128)

[3.3 Руководство пользователя 16](#_Toc27470129)

[ЗАКЛЮЧЕНИЕ 22](#_Toc27470130)

[СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ 23](#_Toc27470131)

[ПРИЛОЖЕНИЯ 24](#_Toc27470132)

# ВВЕДЕНИЕ

Основой для любой информационной системы является база данных, правильное проектирование которой позволяет избежать многих нюансов при подготовке документации к ней, что влечет за собой интенсивную положительную динамику в экономическом плане.

Базы данных представляют собой различные структуры, возможности которых безграничны. Поэтому они предоставляют информацию не только в понятной структурированной форме, но и помогает упорядочить большое число информации, что не требует много времени. Все это ставит базы данных на лидирующие мировые позиции.

Целью данного курсового проекта, является разработка Автоматизированной информационной системы для гостиничного бизнеса с использованием СУБД Microsoft SQL Server.

Актуальность темы обуславливается необходимостью создания информационной системы, которая упростит хранение и обработку данных

Объектом в данной работе является «Информационная система для гостиничного бизнеса».

Предметом является возможность оптимизации процессов, связанных с обработкой информации и её обменом с пользователями, регистрация незарегистрированных, и авторизация в системе уже зарегистрированных клиентов.

Для достижения цели необходимо выполнить следующие задачи:

* Проанализировать прототипы;
* Произвести анализ и проектирование структуры базы данных;
* Выбрать и обосновать программные средства реализации;
* Разработать техническое задание;
* Разработать информационную систему;
* Провести тестирование и отладить информационную систему;
* Разработать руководство пользователя.

# ТЕХНИЧЕСКОЕ ЗАДАНИЕ НА РАЗРАБОТКУ ИНФОРМАЦИОННОЙ СИСТЕМЫ

Разработка информационной системы «Автоматизированная информационная система для гостиничного бизнеса»

Требования к функциональным характеристикам следующие:

1. Необходимо предусмотреть два режима работы – режим администратора и режим пользователя.
2. В режиме пользователя должна быть возможность зарегистрироваться, заполнить свою анкету (ФИО, дата рождения, пол, заселён да/нет, логин, пароль). Реализовывать возможность смены пароля и редактирования данных.
3. В режиме пользователя можно забронировать свободный номер и узнать его цену.
4. При бронировании номера должна быть возможность выбора даты заселения и указания количества персон.
5. В режиме администратора должна быть реализована возможность просматривать, добавлять и редактировать данные постояльцев, а также персонала.
6. Также у администратора имеется возможность формировать отчеты: по номерам и по постояльцам.

Требования к информационному обеспечению следующие:

К приложению должна быть предоставлена документация, содержащая техническое описание продукта и руководство пользователя.

Требования к лингвистическому обеспечению следующие:

Приложение должно иметь интерфейс и документацию на русском языке.

Требования к программному обеспечению следующие:

Разработка программного продукта предусматривается под OS Windows 10.

Требования к техническому обеспечению следующие:

Для работы приложения требуется подключение к программе Microsoft SQL Server, а также использование программ Visual Studio (2017 и выше). Также, необходимо не менее 4 Гб. оперативной памяти и 100Мб свободного места на жестком диске.

# 2. ОБЩАЯ ЧАСТЬ

## 2.1 Выбор и обоснование средств реализации БД

Основным требованием работы является наличие интерфейса приложения, разрабатываемого в СУБД Microsoft SQL Server, далее MS SQL Server.

Microsoft SQL Server является одной из наиболее популярных систем управления базами данных (СУБД) в мире. Данная СУБД подходит для самых различных проектов: от небольших приложений до больших высоконагруженных проектов. Центральным аспектом в MS SQL Server, как и в любой СУБД, является база данных. База данных представляет хранилище данных, организованных определенным способом. Нередко физически база данных представляет файл на жестком диске, хотя такое соответствие необязательно. Для хранения и администрирования баз данных применяются системы управления базами данных (database management system) или СУБД (DBMS). И как раз MS SQL Server является одной из такой СУБД.

Microsoft SQL Server характеризуется такими особенностями как:

* Производительность. MS SQL Server работает очень быстро.
* Надежность и безопасность. MS SQL Server предоставляет шифрование данных.
* Простота. С данной СУБД относительно легко работать и вести администрирование.
* MS SQL Server предоставляет такие опции, как усиленная безопасность, улучшенное сжатие данных, сервисы интеграции.

## 2.2. Выбор и обоснование средств разработки программного продукта

При проектировании приложения были использованы такие программные средства как Microsoft SQL Server и Visual Studio 2017.

Основу любой Информационной системы составляет, Система управления базами данных (СУБД), поэтому среди большого количества СУБД, которые есть в Интернете были рассмотрены три основных сервера, которые имеют определённые преимущества перед другими - это Oracle Database, MySQL и MS SQL Server.

СУБД Oracle это мощный программный комплекс, позволяющий создавать приложения любой степени сложности. Ядром этого комплекса является база данных, хранящая информацию, количество которой за счет предоставляемых средств масштабирования практически безгранично. C высокой эффективностью работать с этой информацией одновременно может практически любое количество пользователей (при наличии достаточных аппаратных ресурсов), не проявляя тенденции к снижению производительности системы при резком увеличении их числа.

Система управления базами данных (СУБД) MySQL - разработка шведской компании MySQL AB. СУБД MySQL является программным обеспечением с открытым исходным кодом, распространяемым по лицензии GNU (GPL) и коммерческой лицензии для ситуаций, не подпадающих под действие лицензии GPL. MySQL поддерживает реляционную модель данных, т. е. представляет собой реляционную СУБД.

Основные достоинства СУБД MySQL:

Высокое качество - MySQL характеризуется устойчивой работой.

Наряду с Oracle, MySQL считается одной из самых быстрых СУБД в мире.

Открытый код доступен для просмотра и модернизации, что позволяет постоянно улучшать программный продукт.

СУБД MySQL, разработанная с использованием языков C/C++, протестирована на многих платформах, среди которых Windows, Linux, FreeBSD, Mac OS X, OS/2, Solaris и др.

MySQL поддерживает API (Application Programming Interface, программный интерфейс приложения) для С, C++, Eiffel, Java, Perl, PHP, Python, Ruby и Tcl. MySQL можно успешно применять как для построения Web-страниц с использованием Perl, PHP и Java, так и для работы прикладной программы, созданной с использованием Delphi, Builder C++ или платформы .NET.

СУБД MySQL предоставляет широкий выбор типов таблиц, в том числе и сторонних разработчиков, что позволяет реализовать оптимальную для решаемой задачи производительность и функциональность.

Локализация в MySQL выполнена корректно. У пользователя, как правило, не возникает проблем при обработке русского содержимого БД.

Microsoft SQL Server является мощнейшей системой управления базами данных. Помимо стандартных для СУБД функций, MS SQL Server содержит большой набор интегрированных служб по анализу данных. Доступ к данным, расположенным на SQL Server могут получить любые приложения, разработанные на .Net и Microsoft Visual Studio, а также приложения пакета Microsoft Office. MS SQL Server обеспечивает высочайшую в своём классе масштабируемость, производительность и безопасность.

Microsoft SQL Server поддерживают более 15000 поставщиков программного обеспечения, его используют более 74,7% крупнейших компаний мира. Microsoft SQL Server используют практически все ERP и CRMсистемы, как SAP, Microsoft Dynamics, 1C: Предприятие, Microsoft CRM.

После анализа СУБД, выбор был сделан в пользу Microsoft SQL Server. Он полностью отвечает всем требованиям и является достаточно мощным инструментом для работы с базами данных.

При выполнении данной работы был выбран язык программирования C# (Си Шарп) ­­­­­– Компания Microsoft уделяет значительное внимание поддержке языка разработки, а потому регулярно появляются обновления и дополнения, исправляются выявленные баги в компиляторе, расширяются библиотеки. Разработчики заинтересованы в популяризации инструмента и прилагают к этому массу усилий.

Разработчики предоставляют подробную и развернутую документацию на своих официальных ресурсах. Кроме того, ответы практически на любые вопросы, связанные с работой в C#, можно найти в сети. Популярность языка привела к появлению множества профессиональных сообществ, посвященных C#. Существует множество учебников, курсов для новичков, видео подборок и других обучающих материалов.

Перечень возможностей разработки практически не имеет ограничений благодаря широчайшему набору инструментов и средств. Конечно, все это можно реализовать при помощи других языков, но некоторых из них узкоспециализированные, в других придется использовать дополнительные инструменты сторонних разработчиков. В C# решение широкого круга задач возможна быстрее, проще и с меньшими затратами времени и ресурсов.

C# может обрабатывать исключения. С помощью этого инструмента можно легко выявлять и обрабатывать ошибки в коде. Способ является структурированным с широким набором функций. При этом важно не злоупотреблять возможностями работы с исключениями, так как при неправильном использовании появляется риск появления «багов».

В языке принята общая система работы с типами, начиная от примитивов и заканчивая сложными, в том числе, пользовательскими наборами. Применяется единый набор операций для обработки и хранения значений типизации. Также можно использовать ссылочные типы пользователя, что позволит динамически выделить память под объект или хранить упрощенную структуру в сети. Он запрещает обращение к переменным, которые не были инициализированы, что исключает возможность выполнения бесконтрольного приведения типов или выхода за пределы определенного массива данных.

Для работы приложений на C# необходимо установить и настроить платформу NET Framework. Она поставляется полностью бесплатно, применяется крайне широко, а потому проблем с пользовательскими устройствами обычно не возникает. Платформа встроена в установочный пакет Windows, при необходимости ее также можно скачать и «поставить» отдельно. Существуют версии для Линукс и MAC.

C# на протяжении долгого времени уверенно лидирует в рейтинге лучших и наиболее востребованных на рынке разработки языков. Сначала им заинтересовались только разработчики, которые пишут приложения под Windows. Но в процессе развития C# «научился» работать на Mac, Linux, iOS и Android. А после того, как код платформы открыли для всех желающих, были сняты практически все возможные ограничения в применении C#. В результате язык активно развивается, применяется все шире.

## 2.3 Анализ прототипов, существующих ИС

При реализации программного продукта, разработчик анализирует и систематизирует данные, просматривая аналогичные приложения. На данном этапе происходит весь сбор информации, выявление достоинств и недостатков, которые имеются у прототипов. После проведённой работы, он продумывает план создания приложения, куда будут включены плюсы, а также исключены ошибки прототипов.

Схема анализа прототипов включает в себя:

* дизайн приложения и общее впечатление от него;
* доступность контактной информации для пользователей;
* структура программного продукта и удобство использования;
* навигация в приложении;
* объем и актуальность представленной информации в данном программном продукте;

удобство использования панели администратора данного приложения.

При проектировании приложения «ИС для гостиничного бизнеса» прототипом стал сайт одной из крупнейшей гостиничной компании «Radisson Hotels» на который можно перейти по ссылке www.radissonhotels.com представлено на рисунке 1.

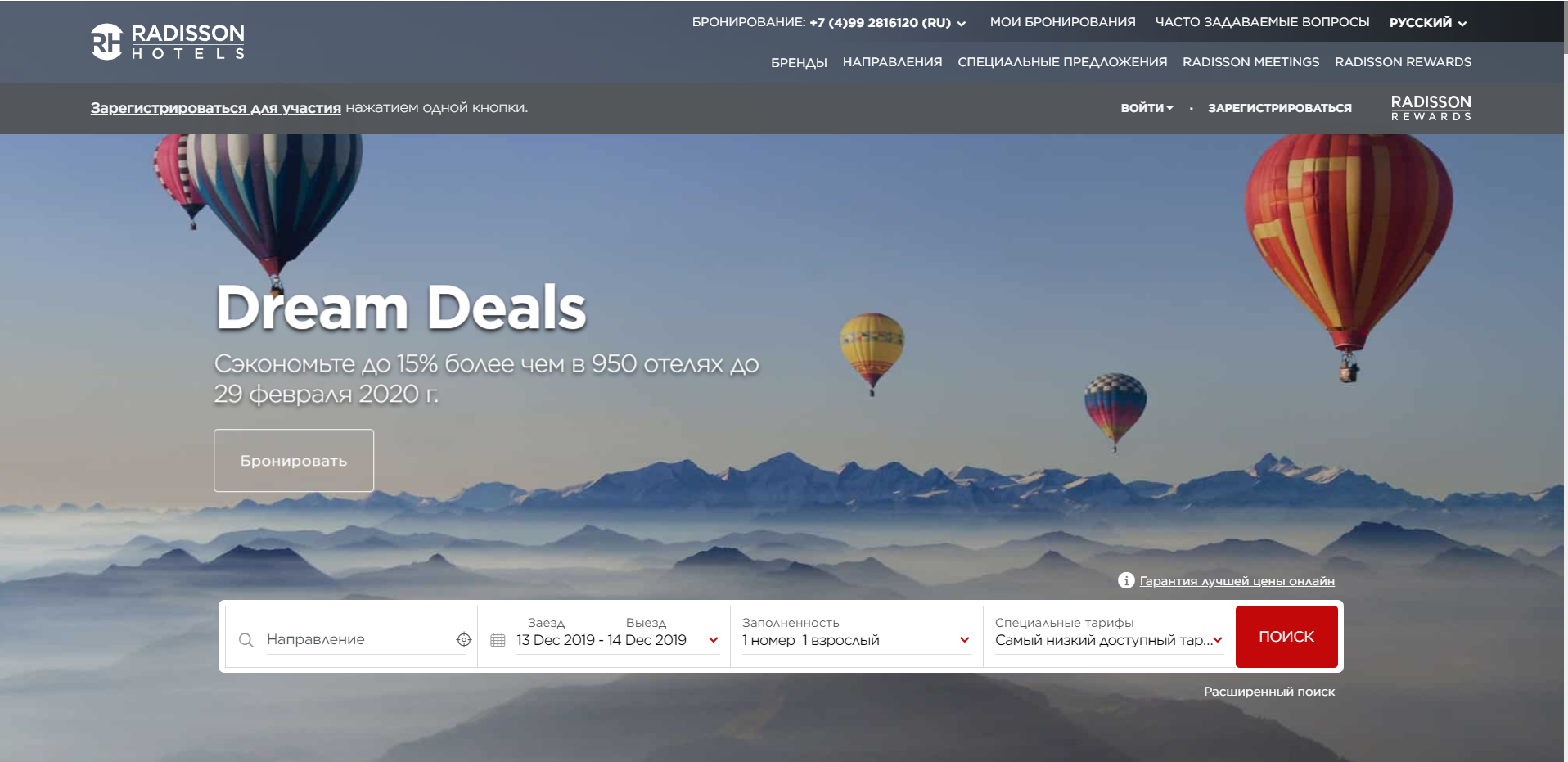


Рисунок 1 - «Сайт «Radisson Hotels»»

Простой дизайн и навигация, имеется возможность выбора даты заезда и выезда из гостиницы. Также можно зарегистрироваться в личном кабинете.

При бронировании номера есть возможность просмотра различных категорий номеров с описанием и ценой. Представлено на рисунке 2.

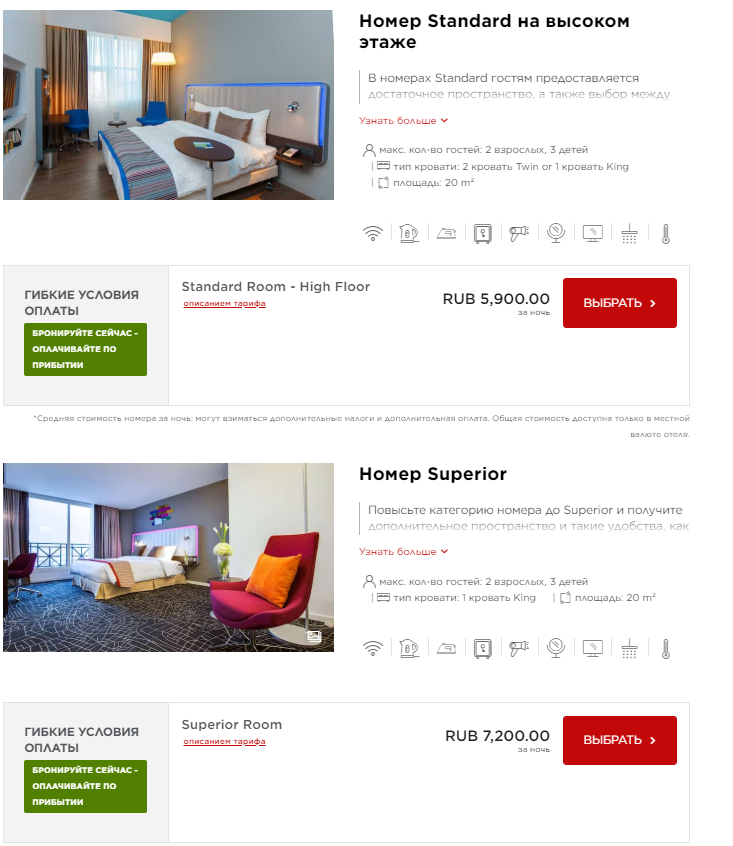


Рисунок 2 - «Категории номеров»

# 3. СПЕЦИАЛЬНАЯ ЧАСТЬ

## 3.1 Разработка алгоритма и проектирование структуры базы данных

Программный продукт «ИС для гостиничного бизнеса» был создан на основании задания, представленного в приложении А.

Модель базы данных «ИС для гостиничного бизнеса» создана в СУБД Microsoft SQL Server.

При создании таблиц базы данных использовался режим конструктора MS SQL, где были установлены свойства полей всех таблиц данной базы данных.

База данных «ИС для гостиничного бизнеса» состоит из четырёх таблиц:

Для создания таблиц использовался режим конструктора, позволяющий установить свойства полей таблицы.

Модель базы данных «ИС для гостиничного бизнеса» состоит из четырёх таблиц, каждая из которых имеет свою структуру, необходимую для установления связи между всеми таблицами:

* «bron» (Бронирование);
* «klienti» (Клиенты);
* «nomera» (Номера);
* «sotrudniki» (Сотрудники).

Таблица «bron» предназначена для хранения информации о бронировании номеров. Ключевым полем является id (Код бронирования), тип данных int. Также имеются поля fam (Фамилия), тип данных nchar (20), im (Имя) тип данных nchar (20), otch (Отчество) тип данных nchar (20), nomer (Номер) тип данных nchar (15), col\_mest (количество мест) тип данных int, zaezd (Заезд) тип данных datetime, viezd (Выезд) тип данных datetime.

Таблица «klienti» предназначена для хранения информации о клиентах. Ключевым полем является id (Код клиента), тип данных int. Также имеются поля fam (Фамилия), тип данных nchar (20), im (Имя) тип данных nchar (20), otch (Отчество) тип данных nchar (20), date\_born (Дата рождения) тип данных date, pol (пол) тип данных nchar (3), zacelen (Заселён) тип данных nchar (3), login (Логин) тип данных nchar (20), password (Пароль) тип данных nchar (20).

Таблица «nomera» предназначена для хранения информации о номерах. Ключевым полем является id (Код номера), тип данных int. Также имеются поля categoria (Категория), тип данных nchar (15), col\_mest (Количество мест) тип данных int, opisanie (Описание) тип данных nchar (200), stoimost (Стоимость) тип данных float, photo (фото) тип данных nchar (30).

Таблица «nomera» предназначена для хранения информации о номерах. Ключевым полем является id (Код номера), тип данных int. Также имеются поля categoria (Категория), тип данных nchar (15), col\_mest (Количество мест) тип данных int, opisanie (Описание) тип данных nchar (200), stoimost (Стоимость) тип данных float, photo (фото) тип данных nchar (30).

Таблица «sotrudniki» предназначена для хранения информации о сотрудниках. Ключевым полем является id (Код сотрудника), тип данных int. Также имеются поля fam (Фамилия), тип данных nchar (20), im (Имя) тип данных nchar (20), otch (Отчество) тип данных nchar (20), date\_born (Дата рождения) тип данных date, dolzhnost тип данных nchar (20), zarplata тип данных float, photo (Фото) тип данных nchar (30), type (Тип учётной записи) тип данных bit, login (Логин) тип данных nchar (20), password (Пароль) тип данных nchar (20). Схема связей таблиц представлена на рисунке 3.

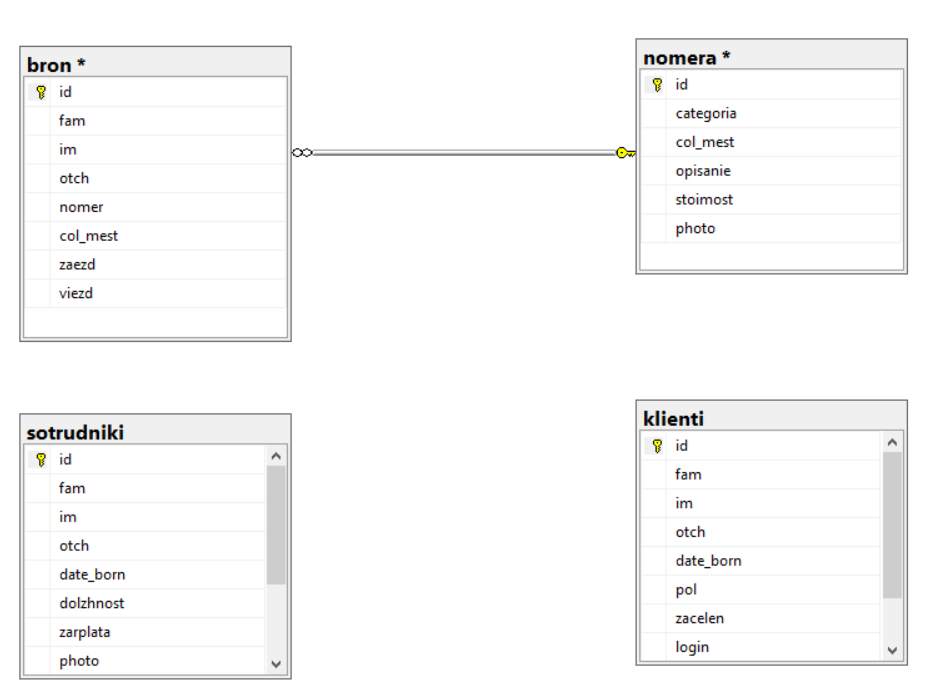


Рисунок 3 - «Схема связей таблиц»

## 3.2 Разработка программного продукта

При разработке данного программного продукта в нём было реализовано несколько подсистем. Первая из них – это система авторизации.

Алгоритм работы выглядит так, что при загрузке страницы, происходит проверка на существование такого пользователя в базе данных и сопоставляет введённый логин и пароль с тем, что имеется в базе данных.

Также имеется проверка на введение пустых данных.

Алгоритм представлен на листинге 1.

Листинг 1 код кнопки «Войти» на форме входа администратора:

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

{

string login = textBox1.Text.Trim();

string password = textBox2.Text.Trim();

if (login == "" || password == "")

{

MessageBox.Show("Вы не ввели данные для входа");

}

else

{

SqlConnection connect = new SqlConnection(Program.st\_connect);

connect.Open();

string m = "select \* from sotrudniki where login='" + login + "' and password='" + password + "'";

SqlCommand command = new SqlCommand(m, connect);

SqlDataReader reader = command.ExecuteReader();

if (reader.HasRows)

{

reader.Read();

Program.id\_user = reader.GetInt32(0);

Program.type\_user = reader.GetBoolean(8);

if (Program.type\_user == true)

{

Form6 f = new Form6();

f.Show();

this.Hide();

}

{

}

}

else

{

MessageBox.Show("Пользователь не найден");

}

connect.Close();

}

}

Вторая из подсистем – это регистрация администратора. При заполнении полей определённой длины и определёнными символами происходит добавление нового пользователя в базу данных. За это отвечает кнопка «Регистрация» на форме регистрации администратора. На форме вводятся все необходимые данные нового пользователя. Код кнопки «Регистрация» представлен на листинге 2.

Листинг 2 код кнопки «Регистрация».

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

{

if (textBox1.Text.Trim() == "" ||

textBox2.Text.Trim() == "" ||

textBox3.Text.Trim() == "" ||

textBox4.Text.Trim() == "" ||

textBox6.Text.Trim() == "" ||

textBox7.Text.Trim() == "" ||

textBox8.Text.Trim() == "" )

{

MessageBox.Show("Не все поля заполнены!!");

}

else

{

SqlConnection conn

= new SqlConnection(Program.st\_connect);

conn.Open();

string s1 = "select \* from sotrudniki where login='"

+ textBox7.Text.Trim() + "'";

SqlCommand comm1 = new SqlCommand(s1, conn);

SqlDataReader read1 = comm1.ExecuteReader();

if (read1.HasRows)

{

MessageBox.Show("Логин занят!!");

}

else

{

conn.Close();

conn.Open();

string s = "insert into sotrudniki " + "" +

"(fam, im, otch, date\_born, dolzhnost, zarplata, type, login, password) values " +

"('" + textBox1.Text.Trim() + "'," +

" '" + textBox2.Text.Trim() + "', " +

"'" + textBox3.Text.Trim() + "', " +

"'" + textBox4.Text.Trim() + "', 'Администратор' ," +

"'" + textBox6.Text.Trim() + "', 1," +

"'" + textBox7.Text.Trim() + "', " +

"'" + textBox8.Text.Trim() + "') ";

SqlCommand comm = new SqlCommand(s, conn);

comm.ExecuteScalar();

this.Close();

}

conn.Close();

}

}

}

Третья из подсистем – это регистрация гостя. При заполнении полей определённой длины и определёнными символами происходит добавление нового пользователя в базу данных. За это отвечает кнопка «Регистрация» на форме регистрации гостя. На форме вводятся все необходимые данные нового пользователя. Код кнопки «Регистрация» представлен на листинге 3

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

{

if (textBox1.Text.Trim() == "" ||

textBox2.Text.Trim() == "" ||

textBox3.Text.Trim() == "" ||

textBox4.Text.Trim() == "" ||

comboBox2.Text.Trim() == "" ||

comboBox1.Text.Trim() == "" ||

textBox7.Text.Trim() == "" ||

textBox8.Text.Trim() == "")

{

MessageBox.Show("Не все поля заполнены!!");

}

else

{

SqlConnection conn

= new SqlConnection(Program.st\_connect);

conn.Open();

string s1 = "select \* from klienti where login='"

+ textBox7.Text.Trim() + "'";

SqlCommand comm1 = new SqlCommand(s1, conn);

SqlDataReader read1 = comm1.ExecuteReader();

if (read1.HasRows)

{

MessageBox.Show("Логин занят!!");

}

else

{

conn.Close();

conn.Open();

string s = "insert into klienti " + "" +

"(fam, im, otch, date\_born, pol, zacelen, login, password) values " +

"('" + textBox1.Text.Trim() + "'," +

" '" + textBox2.Text.Trim() + "', " +

"'" + textBox3.Text.Trim() + "', " +

"'" + textBox4.Text.Trim() + "', " +

"'" + comboBox2.Text.Trim() + "', " +

"'" + comboBox1.Text.Trim() + "', " +

"'" + textBox7.Text.Trim() + "', " +

"'" + textBox8.Text.Trim() + "') ";

SqlCommand comm = new SqlCommand(s, conn);

comm.ExecuteScalar();

this.Hide();

}

conn.Close();

Form6 f = new Form6();

f.Show();

}

}

}

}

Четвёртая из подсистем – это главная форма администратора. На ней можно посмотреть список сотрудников, гостей и номеров. Также под каждым списком имеются кнопки «Добавить», «Просмотреть информацию», «Удалить», а также сформировать отчёт. Коды кнопок представлены в приложении Б.

## 3.3 Руководство пользователя

Для того, чтобы зайти в данное приложение, необходимо дважды щёлкнуть по ярлыку «Гостиница» на рабочем столе.

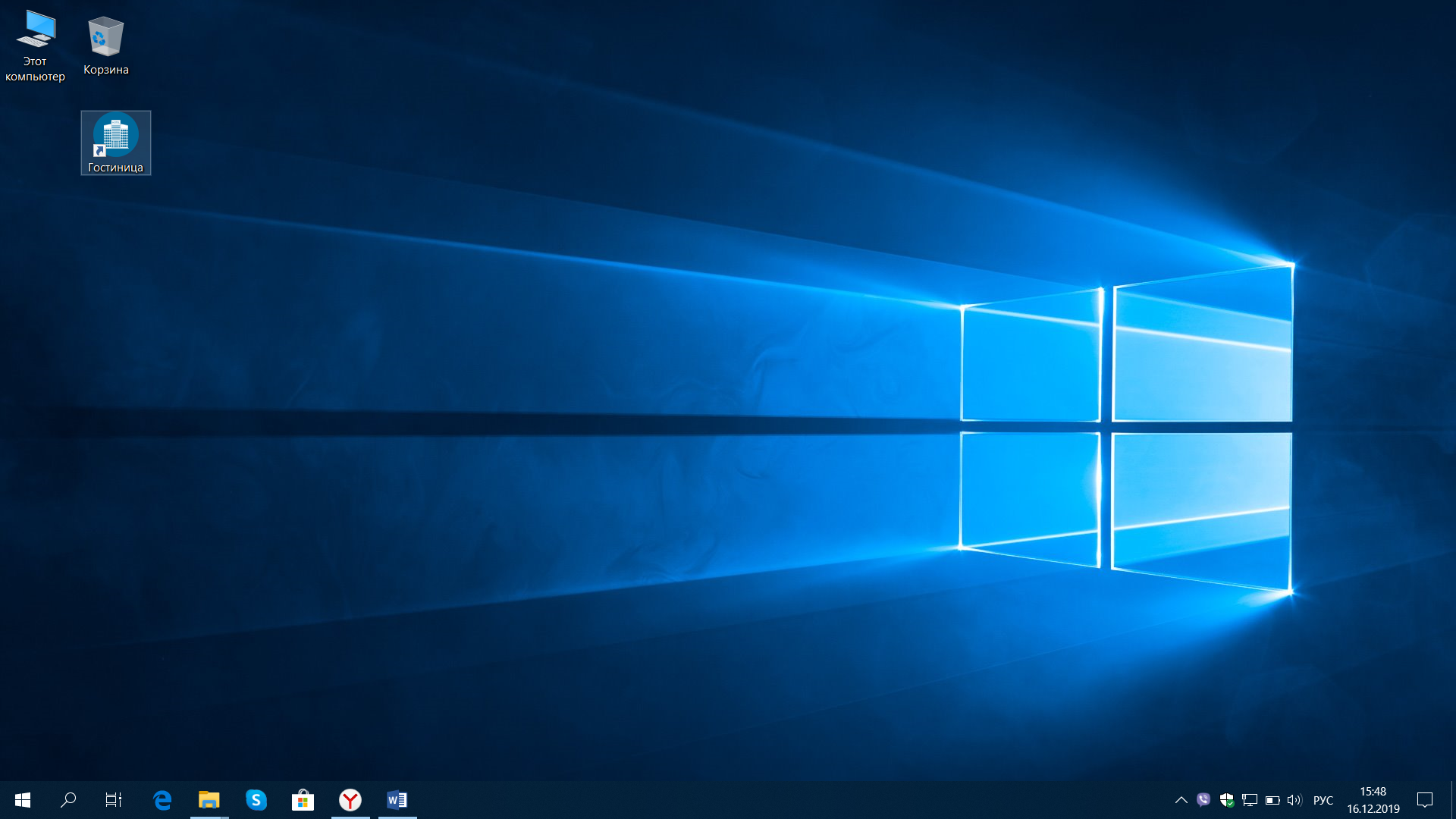


Рисунок 4 - запуск приложения «Гостиница» в Visual Studio

Далее открывается Visual Studio, и для запуска приложения необходимо нажать на кнопку «Пуск» в Visual Studio. Запуск представлен на рисунке 5.

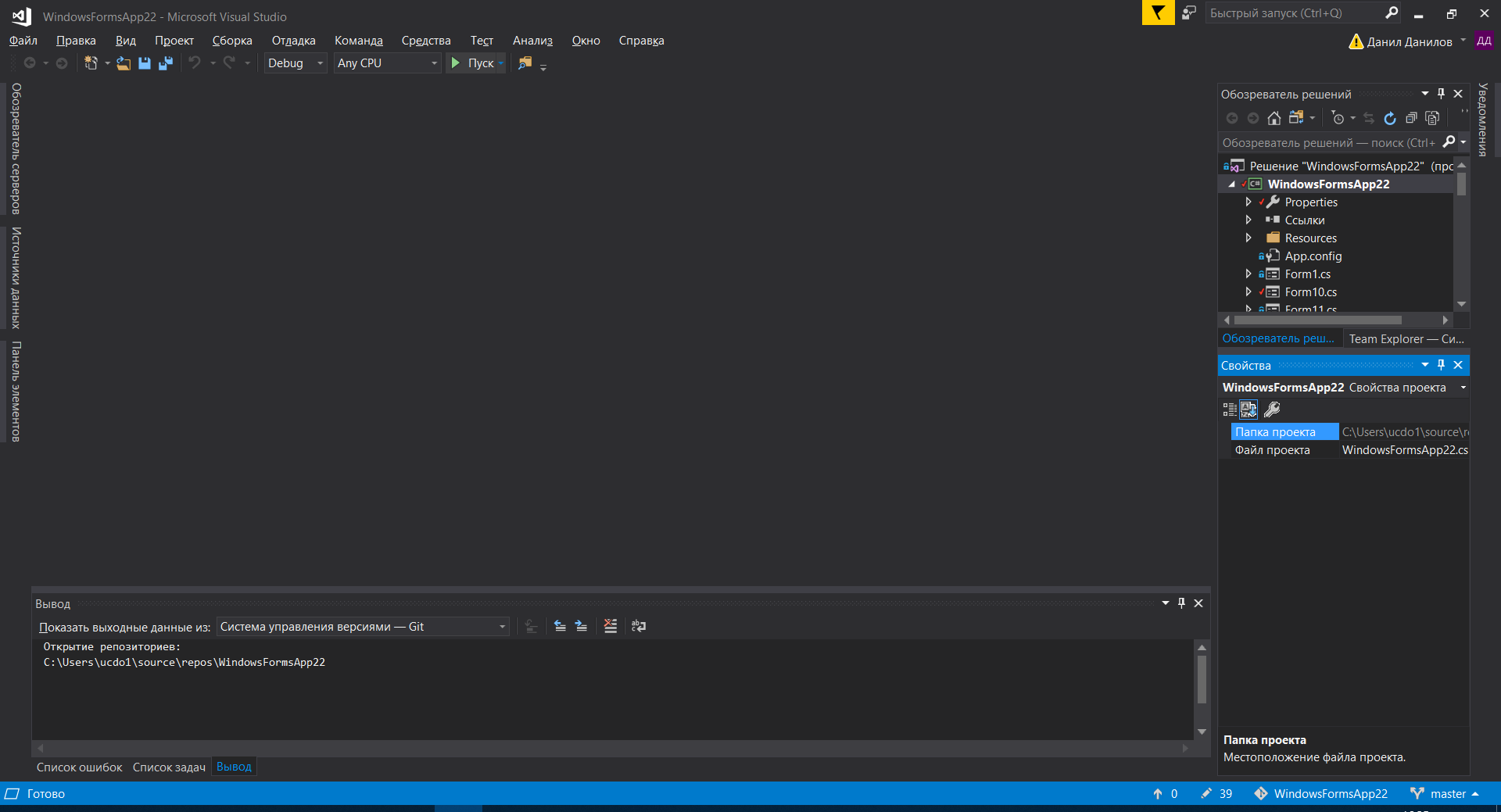


Рисунок 5 - запуск программы «Гостиница» в Visual Studio

После запуска приложения, пользователя встречает главное окно программы. Где пользователь должен выбрать свой тип учётной записи.

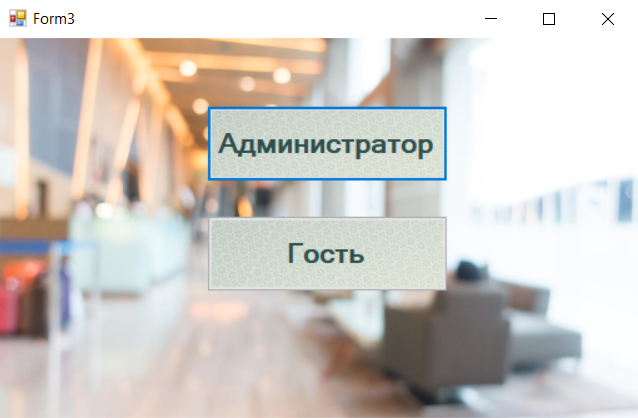


Рисунок 6 - главное окно программы

При нажатии на одну из кнопок откроется либо «окно входа Администратора», либо «окно входа Гостя». «Окно входа Администратора» представлено на рисунке 7, а «окно входа Гостя» на рисунке 8.

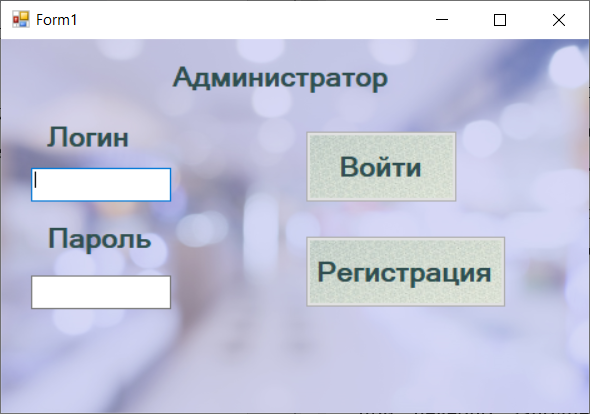


Рисунок 7 - окно входа администратора

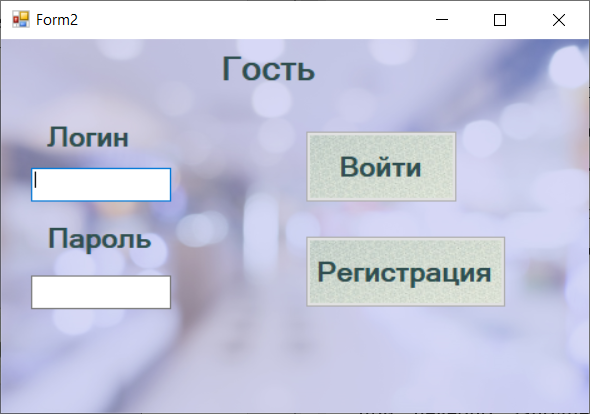


Рисунок 7 - окно входа гостя.

При входе от имени администратора пользователя встречает окно администрирования. В нём имеется возможность просматривать, добавлять, удалять и изменять данные, а также формировать отчёты о сотрудниках, клиентах и номерах. Окно администрирования представлено на рисунке 8. Пример отчёта представлен на рисунке 9.

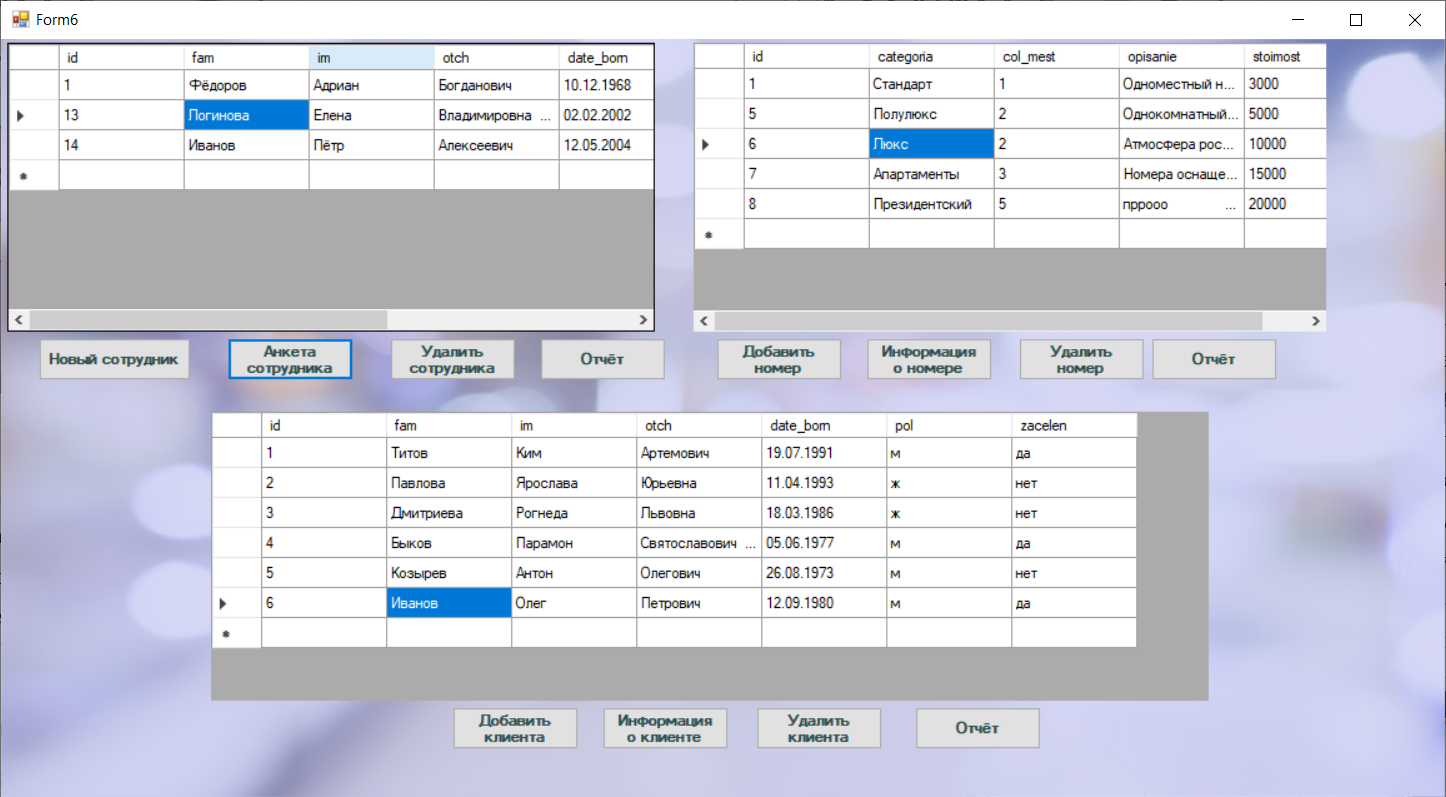


Рисунок 8 - окно администрирования.

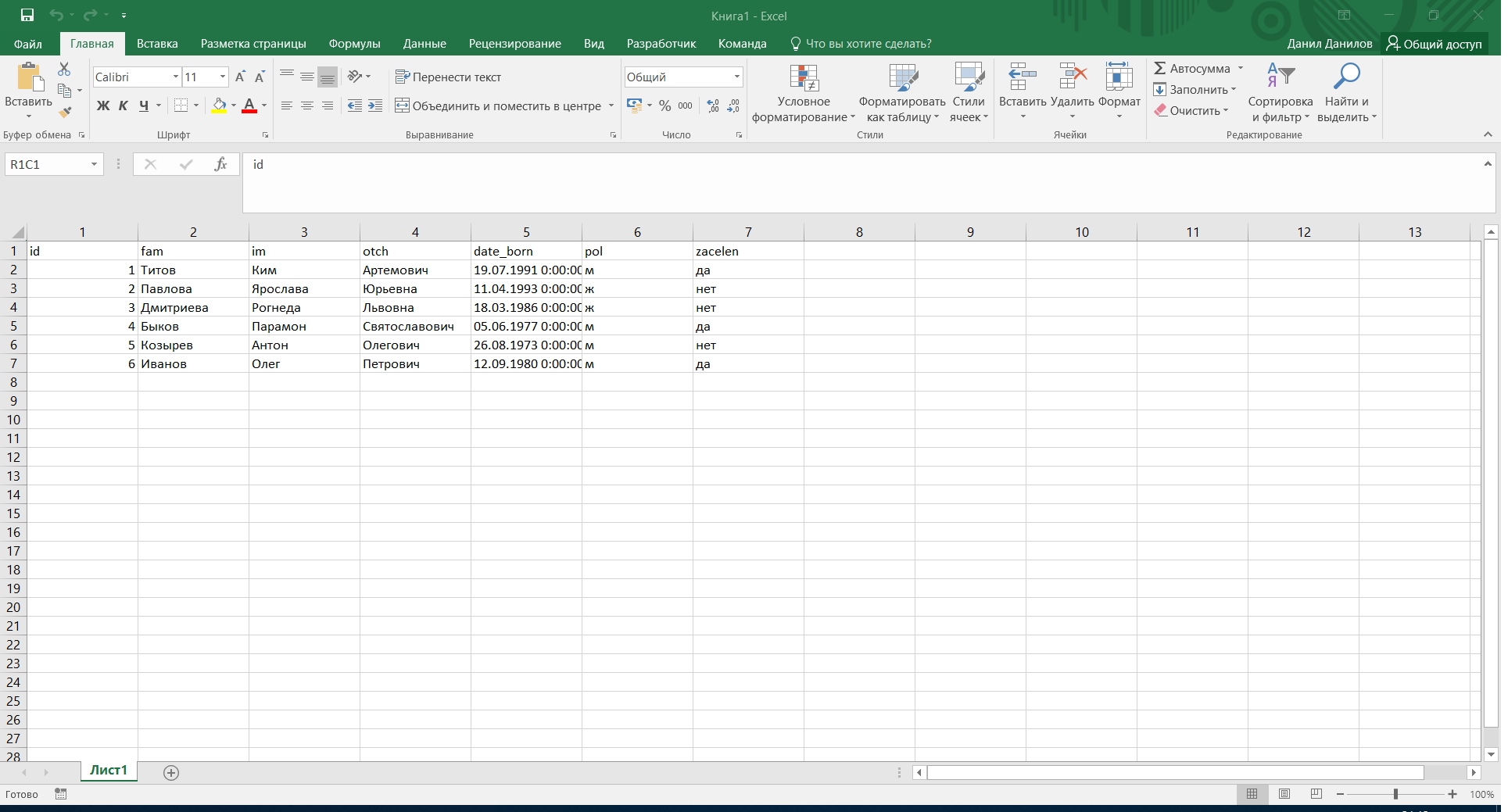


Рисунок 9 - отчёт по клиентам.

При входе от имени гостя пользователя встречает анкета того гостя логин и пароль которого, он ввёл при входе. Пример анкеты гостя представлен на рисунке 10.

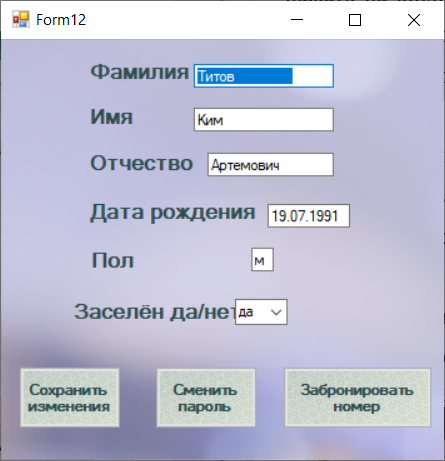


Рисунок 10 - анкета гостя.

С этой анкеты гость может забронировать номер и сменить пароль, а также при необходимости изменить свои личные данные. окно смены пароля представлено на рисунке 11.

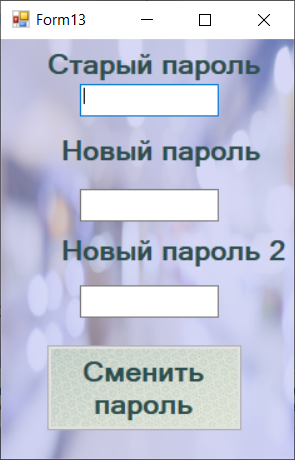


Рисунок 11 - окно смены пароля.

При нажатии на кнопку «Забронировать номер» у гостя откроется окно, в котором он сможет указать необходимое количество спальных мест в номере, выбрать категорию номера, дату заезда и выезда, а также узнать финальную цену бронирования. Пример бронирования номера представлен на рисунке 12.

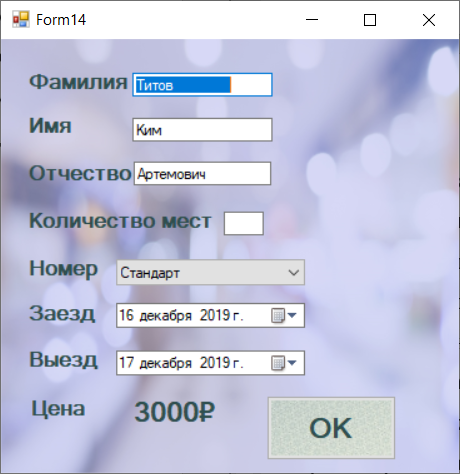


Рисунок 12 - окно бронирования номера

# ЗАКЛЮЧЕНИЕ

В результате курсовой работы был разработан интерфейс программного продукта для гостиничного бизнеса было проведено тестирование и отладка программного обеспечения.

Разработанный программный продукт позволит упростить учёт сотрудников, клиентов и номеров, а также автоматизировать формирование отчётов.

Данный программный продукт в первую очередь подойдёт небольшим хостелам, которым нерационально использовать дорогостоящие программные решения, которые достаточно требовательны к аппаратному обеспечению и сложны в использовании, что требует от владельцев хостелов дополнительных капиталовложений на обучение персонала.

В дальнейшем планируется реализовать просмотр нескольких фотографий номера, а также формирование отчёта по уже заселённым клиентам.

Цель достигнута, задачи выполнены.

# СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Новожилов, О. П. Архитектура компьютерных систем в 2 ч. Часть 1 : учебное пособие для среднего профессионального образования / О. П. Новожилов. — Москва : Издательство Юрайт, 2018. — 276 с. — (Профессиональное образование). — ISBN 978-5-534-10299-4. — Режим доступа : www.biblio-online.ru/book/arhitektura-kompyuternyh-sistem-v-2-ch-chast-1-442490
2. Новожилов, О. П. Архитектура компьютерных систем в 2 ч. Часть 2 : учебное пособие для среднего профессионального образования / О. П. Новожилов. — Москва : Издательство Юрайт, 2018. — 246 с. — (Профессиональное образование). — ISBN 978-5-534-10301-4. — Режим доступа : www.biblio-online.ru/book/arhitektura-kompyuternyh-sistem-v-2-ch-chast-2-442491
3. Осокин, А. Н. Теория информации : учебное пособие для среднего профессионального образования / А. Н. Осокин, А. Н. Мальчуков. — Москва : Издательство Юрайт, 2018. — 205 с. — (Профессиональное образование). — ISBN 978-5-534-11417-1. — Режим доступа : www.biblio-online.ru/book/teoriya-informacii-445263
4. Миленина, С. А. Электротехника, электроника и схемотехника : учебник и практикум для среднего профессионального образования / С. А. Миленина, Н. К. Миленин ; под редакцией Н. К. Миленина. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2018. — 406 с. — (Профессиональное образование). — ISBN 978-5-534-04676-2. — Режим доступа : www.biblio-online.ru/book/elektrotehnika-elektronika-i-shemotehnika-433455

**ПРИЛОЖЕНИЯ**

Приложение Б

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

{

Form7 f = new Form7();

f.Show();

this.Hide();

}

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

{

int k = dataGridView1.CurrentRow.Index;

Program.id\_user = Convert.ToInt32(dataGridView1[0, k].Value);

SqlConnection conn = new SqlConnection(Program.st\_connect);

conn.Open();

string s = "delete from sotrudniki where id="

+ Program.id\_user.ToString();

SqlCommand comm = new SqlCommand(s, conn);

comm.ExecuteScalar();

conn.Close();

dataGridView1.DataSource.ToString();

this.Hide();

Form6 f = new Form6();

f.Show();

}

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

{

}

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

{

}

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

{

int k = dataGridView1.CurrentRow.Index;

Program.id\_user = Convert.ToInt32(dataGridView1[0, k].Value);

SqlConnection conn = new SqlConnection(Program.st\_connect);

conn.Open();

string s = "select \* from sotrudniki where id="

+ Program.id\_user.ToString();

SqlCommand comm = new SqlCommand(s, conn);

SqlDataReader read = comm.ExecuteReader();

read.Read();

Form8 f = new Form8();

f.textBox1.Text = read.GetString(1);

f.textBox2.Text = read.GetString(2);

f.textBox3.Text = read.GetString(3);

f.textBox4.Text = read.GetDateTime(4).ToString();

f.textBox5.Text = read.GetString(5);

f.textBox6.Text = read.GetDouble(6).ToString();

string p = read.GetString(7);

f.pictureBox1.Load(p);

f.Show();

}

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

{

Form10 f = new Form10();

f.Show();

}

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

{

int k = dataGridView2.CurrentRow.Index;

Program.id\_user = Convert.ToInt32(dataGridView2[0, k].Value);

SqlConnection conn = new SqlConnection(Program.st\_connect);

conn.Open();

string s = "select \* from nomera where id="

+ Program.id\_user.ToString();

SqlCommand comm = new SqlCommand(s, conn);

SqlDataReader read = comm.ExecuteReader();

read.Read();

Form11 f = new Form11();

f.textBox1.Text = read.GetString(1);

f.textBox2.Text = read.GetInt32(2).ToString();

f.textBox3.Text = read.GetString(3);

f.textBox4.Text = read.GetDouble(4).ToString();

string p = read.GetString(5);

f.pictureBox1.Load(p);

f.Show();

}

private void button8\_Click(object sender, EventArgs e)

{

int k = dataGridView1.CurrentRow.Index;

Program.id\_user = Convert.ToInt32(dataGridView1[0, k].Value);

SqlConnection conn = new SqlConnection(Program.st\_connect);

conn.Open();

string s = "delete from nomera where id="

+ Program.id\_user.ToString();

SqlCommand comm = new SqlCommand(s, conn);

comm.ExecuteScalar();

conn.Close();

this.Hide();

Form6 f = new Form6();

f.Show();

}

private void button10\_Click\_1(object sender, EventArgs e)

{

}

private void button9\_Click(object sender, EventArgs e)

{

}

private void button14\_Click(object sender, EventArgs e)

{

int k = dataGridView3.CurrentRow.Index;

Program.id\_user = Convert.ToInt32(dataGridView3[0, k].Value);

SqlConnection conn = new SqlConnection(Program.st\_connect);

conn.Open();

string s = "select \* from klienti where id="

+ Program.id\_user.ToString();

SqlCommand comm = new SqlCommand(s, conn);

SqlDataReader read = comm.ExecuteReader();

read.Read();

Form12 f = new Form12();

f.textBox1.Text = read.GetString(1);

f.textBox2.Text = read.GetString(2);

f.textBox3.Text = read.GetString(3);

f.textBox4.Text = read.GetDateTime(4).ToString();

f.textBox5.Text = read.GetString(5);

f.comboBox1.SelectedItem = read.GetString(6);

f.Show();

}

private void button15\_Click(object sender, EventArgs e)

{

Form5 f = new Form5();

f.Show();

this.Hide();

}

private void dataGridView2\_CellContentClick(object sender, DataGridViewCellEventArgs e)

{

}

private void button13\_Click(object sender, EventArgs e)

{

int k = dataGridView3.CurrentRow.Index;

Program.id\_user = Convert.ToInt32(dataGridView3[0, k].Value);

SqlConnection conn = new SqlConnection(Program.st\_connect);

conn.Open();

string s = "delete from klienti where id="

+ Program.id\_user.ToString();

SqlCommand comm = new SqlCommand(s, conn);

comm.ExecuteScalar();

conn.Close();

this.Hide();

Form6 f = new Form6();

f.Show();

}

private void button1\_Click\_1(object sender, EventArgs e)

{

{

Microsoft.Office.Interop.Excel.Application myExcel =

new Microsoft.Office.Interop.Excel.Application();

myExcel.Application.Workbooks.Add(Type.Missing);

myExcel.Columns.ColumnWidth = 15;

for (int i = 0; i < dataGridView2.ColumnCount; i++)

{

myExcel.Cells[1, i + 1] = dataGridView2.Columns[i].HeaderText;

}

for (int i = 0; i < dataGridView2.ColumnCount; i++)

{

for (int j = 0; j < dataGridView2.RowCount - 1; j++)

{

myExcel.Cells[j + 2, i + 1]

= (dataGridView2[i, j].Value).ToString();

}

}

myExcel.Visible = true;

}

}

private void button2\_Click\_1(object sender, EventArgs e)

{

// создаем виртуальный обьект эксель

Microsoft.Office.Interop.Excel.Application myExcel =

new Microsoft.Office.Interop.Excel.Application();

//создаем книгу в обьекте эксель

myExcel.Application.Workbooks.Add(Type.Missing);

//Настраиваем ячейки

myExcel.Columns.ColumnWidth = 15;

//Пишем заголовки ячеек

for (int i = 0; i < dataGridView3.ColumnCount; i++)

{

myExcel.Cells[1, i + 1] = dataGridView3.Columns[i].HeaderText;

}

for (int i = 0; i < dataGridView3.ColumnCount; i++)

{

for (int j = 0; j < dataGridView3.RowCount - 1; j++)

{

myExcel.Cells[j + 2, i + 1]

= (dataGridView3[i, j].Value).ToString();

}

}

myExcel.Visible = true;

}

private void button9\_Click\_1(object sender, EventArgs e)

{

{//создаем виртуальный обьект эксель

Microsoft.Office.Interop.Excel.Application myExcel =

new Microsoft.Office.Interop.Excel.Application();

//создаем книгу в обьекте эксель

myExcel.Application.Workbooks.Add(Type.Missing);

//Настраиваем ячейки

myExcel.Columns.ColumnWidth = 15;

//Пишем заголовки ячеек

for (int i = 0; i < dataGridView1.ColumnCount; i++)

{

myExcel.Cells[1, i + 1] = dataGridView1.Columns[i].HeaderText;

}

for (int i = 0; i < dataGridView1.ColumnCount; i++)

{

for (int j = 0; j < dataGridView1.RowCount - 1; j++)

{

myExcel.Cells[j + 2, i + 1]

= (dataGridView1[i, j].Value).ToString();

}

}

myExcel.Visible = true;