|  |
| --- |
| **МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ** |
| **ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТЕХНОЛОГИЙ И УПРАВЛЕНИЯ ИМЕНИ К.Г. РАЗУМОВСКОГО (ПЕРВЫЙ КАЗАЧИЙ УНИВЕРСИТЕТ)»**  **(ФГБОУ ВО «МГУТУ ИМ. К.Г. РАЗУМОВСКОГО (ПКУ)»)** |
|  |
| **УНИВЕРСИТЕТСКИЙ КОЛЛЕДЖ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ** |

|  |
| --- |
| **КУРСОВОЙ ПРОЕКТ** |
|  |
| по междисциплинарному курсу МДК.02.02. Технология разработки и защиты баз данных |
|  |
| на тему: Разработка информационной системы для  отеля. |
|  |
| студента группы ПКС-304  специальности 09.02.03 Программирование в компьютерных системах |
| Морозова Игоря Михайловича |

|  |  |  |
| --- | --- | --- |
| Студент |  | И.М. Морозов |
| Руководитель курсового проекта |  | Е.А. Ларионова |
| Председатель ПЦК специальности 09.02.03 Программирование в компьютерных системах |  | А.И. Глускер |

|  |  |  |
| --- | --- | --- |
| Дата защиты «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_ 2020 г. | | |
| Оценка:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | | |
| Заведующий отделением № 3 |  | Е.Л. Гусева |

# ОГЛАВЛЕНИЕ

[ОГЛАВЛЕНИЕ 2](#_Toc40996261)

[ВВЕДЕНИЕ 4](#_Toc40996262)

[1 ОСНОВНАЯ ЧАСТЬ 8](#_Toc40996263)

[1.1 Назначение и цели создания системы 8](#_Toc40996264)

[1.2 Обзор и анализ предметной области 8](#_Toc40996265)

[1.3 Жизненный цикл базы данных 9](#_Toc40996266)

[1.4 Выбор и характеристика СУБД 9](#_Toc40996267)

[1.5 Выбор и характеристика среды разработки 10](#_Toc40996268)

[1.6 Выбор и характеристика языка программирования 11](#_Toc40996269)

[2 СПЕЦИАЛЬНАЯ ЧАСТЬ 12](#_Toc40996270)

[2.1 Постановка задачи 12](#_Toc40996271)

[2.2 Архитектура информационной системы 12](#_Toc40996272)

[2.2.1 Основные этапы проектирования приложения 12](#_Toc40996273)

[2.2.3 Архитектура клиент-сервер 13](#_Toc40996274)

[2.2.4 Навигационная схема приложения 13](#_Toc40996275)

[2.2.5 Архитектура классов приложения 14](#_Toc40996276)

[2.3 Логическая модель базы данных 15](#_Toc40996277)

[2.4 Нормализация таблиц 16](#_Toc40996278)

[2.5 Описание таблиц 17](#_Toc40996279)

[2.6 Разработка приложения 21](#_Toc40996280)

[2.6.1 Диаграмма вариантов использования 21](#_Toc40996281)

[2.6.2 Решение главной задачи проекта 22](#_Toc40996282)

[2.6.4 Защита информационной системы 23](#_Toc40996283)

[2.6.5 Скриншоты интерфейса 23](#_Toc40996284)

[2.7 Инструкция пользователю 27](#_Toc40996285)

[2.7.1 Общие сведения об информационной системе 27](#_Toc40996286)

[2.7.2 Требования к техническим средствам 27](#_Toc40996287)

[2.7.3 Требования к программным средствам 27](#_Toc40996288)

[2.7.4 Настройка информационной системы 28](#_Toc40996289)

[2.7.5 Формы ввода 28](#_Toc40996290)

[2.7.6 Отчеты 29](#_Toc40996291)

[ЗАКЛЮЧЕНИЕ 30](#_Toc40996292)

[СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ 31](#_Toc40996293)

[ПРИЛОЖЕНИЕ А SQL скрипты на создание и заполнение базы данных 32](#_Toc40996294)

[ПРИЛОЖЕНИЕ Б Код программы 46](#_Toc40996295)

# ВВЕДЕНИЕ

Темой проекта является разработка информационной системы для

отеля. В современном мире, мы минимум раз в жизни точно сталкиваемся с проблемой снятия номера в отеле. Есть множество компаний и частных предпринимателей, которые помогают вам осуществить снятие номера в отеле. У некоторых компаний аренда номера происходит через сайт, у некоторых через приложение, которое нужно установить на свое устройство, будь то мобильный телефон или ПК. Приложение для аренды номера облегчают покупателю труд аренду номера, так как в них обычно более интуитивно понятный интерфейс, помогающий пользователю найти нужный номер для сна.

Арендовать номер в настоящее время не составит труда, достаточно найти сайт, на котором можно это сделать в пару кликов. Отталкиваясь от этой идеи мне стало интересно разработать приложение, которое позволяет арендовать номера, несмотря на трудности выбора отеля.

Основные задачи, которые необходимо выполнить для того, чтобы разработать данное приложение:

* подробное изучение функциональных возможностей WPF.Net Framework,
* освоение тонкостей проектирования баз данных,
* изучение Entity Framework,
* разработка функций для работы с базой данных,
* создание функций для работы с данными внутри приложения,
* проектирование интерфейса программы,
* написание функций для корректного сохранения и обновления данных,
* создание функций для корректного отображения содержимого страниц,
* проектирование архитектуры и логики приложения.

Используемые в разработке паттерны и методы:

* объектно-ориентированное программирование,
* моделирование.

Обоснование структуры курсового проекта:

1. введение: в данном разделе описываются цели и задачи проекта, а также его актуальность, так как изначально необходимо определить, что предстоит изучать и делать в проекте,
2. основная часть: этот раздел включает в себя анализ предметной области, для которой разрабатывается данное приложение, назначения и цели разработки, а также жизненный цикл базы данных,
3. специальная часть: в данном разделе описаны архитектурные решения, которые применяются в разработке, а также процесс разработки программного обеспечения и базы данных,
4. заключение: вывод о проделанной работе и полученных навыках за время разработки приложения,
5. список используемых источников: этот раздел содержит в себе перечень ресурсов, которые были использованы для получения необходимой информации во время разработки.

# ОСНОВНАЯ ЧАСТЬ

* 1. Назначение и цели создания системы

Приложение создается для разработки информационной системы

авиакомпании. База данных приложения содержит информацию о пользователях и номерах в отеле.

Цель: создать приложение средствами позволяющее снять номер в отеле и после этого сдать номер администратору.

Назначение системы: обеспечить простое снятие номер.

Базовые сущности этой предметной области: пользователь и номер в отеле

В системе должны существовать две вида пользователей: администратор, клиент

* 1. Обзор и анализ предметной области

Предметной областью разработки является отель. В этой отрасли существует обширный набор вводных параметров. В области некоторых параметров есть данные о номерах в отеле и клиентах.

Список основных сущностей:

* Номер в отеле – номер в отеле, который пользователь может снять,
* пользователь – пользователь или администратор системы,
  1. Жизненный цикл базы данных

Жизненный цикл – это процесс проектирования, реализации и управления базой данных. Он состоит из:

1. Анализа предметной области, который включает в себя планирование разработки (15.02.2020 – 25.02.2020), определение требований (15.02.2020 – 25.02.2020), сбор и анализ требований (15.02.2020 – 25.02.2020),
2. Проектирования базы данных, которое состоит из создания концептуальной модели (26.02.2020), логической модели (26.02.2020), физической модели (27.02.2020),
3. Реализации, в процессе которой пишутся скрипты на создание базы данных, связей сущностей в базе данных, а также разрабатывается интерфейс приложения и его функции (01.03.2020 – 05.05.2020).
   1. Выбор и характеристика СУБД

Для разработки базы данных была выбрана среда Microsoft SQL Server Management Studio 2018, так как она является удобной для использования и не требует долгого освоения тонкостей для создания полноценной базы данных под программу, а также с ней можно работать, используя Entity Framework. Эта СУБД использует язык SQL (T-SQL) для написания необходимых сущностей и заполнения их данными.

При сравнении SQL Server с MySQL можно выделить несколько главных различий:

* первый предназначен для работы с платформой .Net, а второй для работы с PHP,
* синтаксис,
* данные SQL Server хранятся в единой системе Microsoft, в то время как данные MySQL хранятся на движках.

Весомыми плюсами SQL Server являются простота понимания принципов работы, синтаксиса, а также возможность создавать объекты баз данных либо с помощью скрипта, либо вручную, с помощью конструктора.

* 1. Выбор и характеристика среды разработки

Для разработки приложения была выбрана среда программирования Microsoft Visual Studio 2019, потому что эта среда разработки обеспечивает возможность создания приложений с удобным интерфейсом с помощью языка C# и системы построения клиентских приложений WPF.Net Framework.

Microsoft Visual Studio 2019 – продукт компании Microsoft, включающий в себя интегрированную среду разработки программного обеспечения, а также ряд других инструментальных средств.

Некоторые преимущества:

* удобная отладка и поиск ошибок,
* возможность создания специальной библиотеки для тестирования,
* доступность NuGet пакетов, необходимых для работы приложения,
* подробная документация об ошибках, располагающаяся на сайте компании Microsoft,
* интеллектуальность системы, предупреждающая об ошибках и помогающая упростить и улучшить код.

Данную среду разработки можно сравнить с Visual Studio Code. Она является менее удобной, так как не представляет такого обширного количества средств для отладки и тестирования.

* 1. Выбор и характеристика языка программирования

Для написания кода приложения был выбран язык C#, так как он является предпочтительным для меня, а также с целью повышения навыков его использования в различных целях.

C# – объектно-ориентированный язык программирования от компании Microsoft, предназначенный для разработки приложений для платформы Microsoft.Net.

Некоторые преимущества:

* поддерживается ООП,
* удобный синтаксис,

В сравнении с языком Visual Basic, C# является более удобным и лаконичным, так как был изначально спроектирован под платформу .Net.

# 2 СПЕЦИАЛЬНАЯ ЧАСТЬ

2.1 Постановка задачи

Главная задача проекта – спроектировать средствами MS SQL SERVER базу данных и создать приложение средствами Microsoft Visual Studio 2019 и Microsoft SQL Server Management Studio 18, позволяющее осуществлять снятие номера в отеле, редактирование данных в базе данных, прием снятых номеров постояльцев.

Предусмотреть чтобы:

* постоялец не мог снять номер, в котором он уже поселился.
* постоялец не мог снять номер, в котором поселился другой человек.
* В системе должны существовать три вида пользователей: пользователь, администратор.

Пассажир может выполнять следующие действия:

* Снять номер в отеле.

Администратор может выполнять следующие действия:

* Редактировать базу данных
* Принимать снятые комнаты постояльцами:
* Добавление новых комнат

2.2 Архитектура информационной системы

2.2.1 Основные этапы проектирования приложения

Первым пунктом разработки являлось создание первичной версии базы данных для последующей доработки.

После успешного завершения проектирования базы данных нужно было приступать к разработке окна и страниц приложения.

Первой была создана страница приветствия с возможностью перехода к авторизации, либо регистрации.

После этого было разработано меню программы, позволяющее авторизованным пользователям просматривать главное меню приложения, в котором находится либо панель администратора или панель постояльца.

На последнем этапе были созданы различия между правами доступа у пользователей. Следующим шагом была доработка форм ввода данных с целью защиты от неверного ввода информации.

2.2.3 Архитектура клиент-сервер

Клиент-сервер – архитектура, в которой задачи или сетевая нагрузка распределены между поставщиками услуг, которые называются серверами, и заказчиками услуг – клиентами. Серверы ожидают запрос от клиента и передают им свои ресурсы в виде данных. Модель данной архитектуры легко представить графически (Рисунок 2).



Рисунок 2 - модель архитектуры клиент-сервер.

2.2.4 Навигационная схема приложения

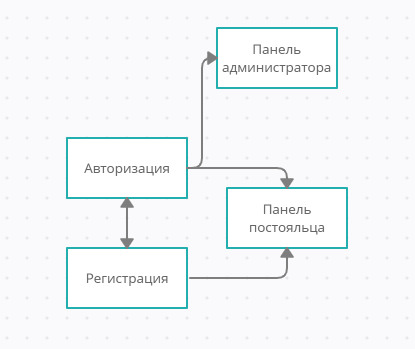


Рисунок 3 – навигационная схема приложения

2.2.5 Архитектура классов приложения

Для создания оптимальной архитектуры в среде Microsoft Visual Studio 2019 я создал два решения – приложение WPF.Net Framework с названием Hotel. Классы Hotel отвечают за интерфейс приложения и его визуальные функции, такие как:

* создание страниц,
* корректное отображение элементов на странице,
* навигация,
* сохранение экземпляров страниц и других необходимых элементов с целью предотвращения создания дубликатов, а также обеспечения быстрого доступа,
* обработка событий, вызванных пользователем,
* проверка вводимых пользователем данных,
* отображение полученных данных их базы.

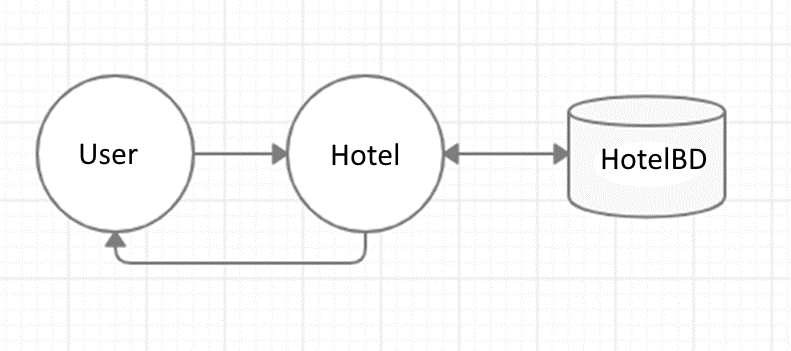


Рисунок 4 – модель архитектуры приложения

2.3 Логическая модель базы данных

Логическая модель базы данных, используемой в приложении (Рисунок 5).

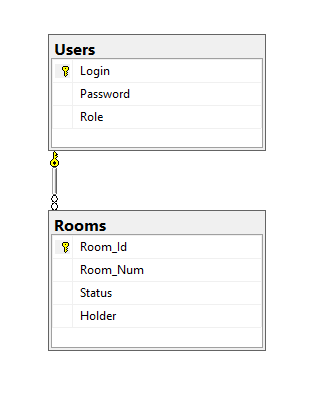


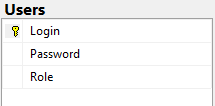
Рисунок 5 – логическая модель базы данных

2.4 Нормализация таблиц

Первая нормальная форма – все атрибуты атомарны.

Вторая нормальная форма – выполнены условия первой нормальной формы, а также каждый неключевой атрибут неприводимо зависит от первичного ключа. Неприводимость – отсутствие атрибутов, от которых можно вывести функциональную зависимость.

Пример сущностей во второй нормальной форме (Рисунок 6).



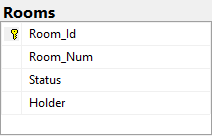


Рисунок 6 – пример второй нормальной формы

Третья нормальная форма – соблюдены условия второй нормальной формы, а также каждый неключевой атрибут нетранзитивно зависит от первичного ключа.

База данных приведена к третьей нормальной форме.

2.5 Описание таблиц

В этом разделе описаны названия таблиц, их атрибуты, свойства и типы атрибутов, первичные и внешние ключи.

Определения:

* FK – внешний ключ,
* PK – первичный ключ,
* NULL – разрешены нулевые значения.

Таблица 1 – Users

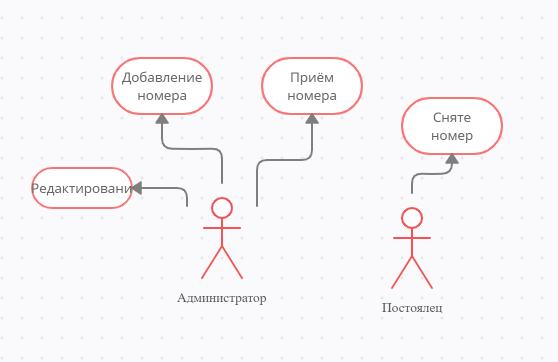
|  |  |  |
| --- | --- | --- |
| Название | Тип | Дополнительно |
| Login | varchar(40) | PK |
| Password | varchar(40) | Unchecked |
| Role | int | Unchecked |

Таблица 2 – Rooms

|  |  |  |
| --- | --- | --- |
| Название | Тип | Дополнительно |
| Room\_Id | int | PK |
| Room\_Num | varchar(10) |  |
| Status | varchar(50) |  |
| Holder | varchar(40) | NULL,FK |

2.6 Разработка приложения

2.6.1 Диаграмма вариантов использования

Рисунок 7 – диаграмма вариантов использования

2.6.2 Решение главной задачи проекта

Главная задача проекта – снятие номера в отеле.

Необходимо:

* определить роль текущего пользователя,
* в соответствии с этой ролью отобразить список доступной для просмотра и редактирования информации
* при выборе номера и нажатия кнопки заселиться заселить пользователя.

Скрипт снятие номера:   
private void Zas\_Click(object sender, RoutedEventArgs e)

{

Rooms r = (Rooms)RoomList.SelectedItem;

Rooms d = bd.Rooms.Find(r.Room\_Id);

d.Users = user;

d.Status = "занята";

bd.SaveChanges();

Refresh();

}

2.6.4 Защита информационной системы

В разрабатываемом приложении предусмотрена защита всех форм ввода от неправильной и некорректной информации. При попытке сохранения неверных данных пользователю будет выведено сообщение об ошибке. При вводе корректных данных все будет успешно сохранено.

2.6.5 Скриншоты интерфейса

Все скриншоты сделаны с аккаунта администратора.

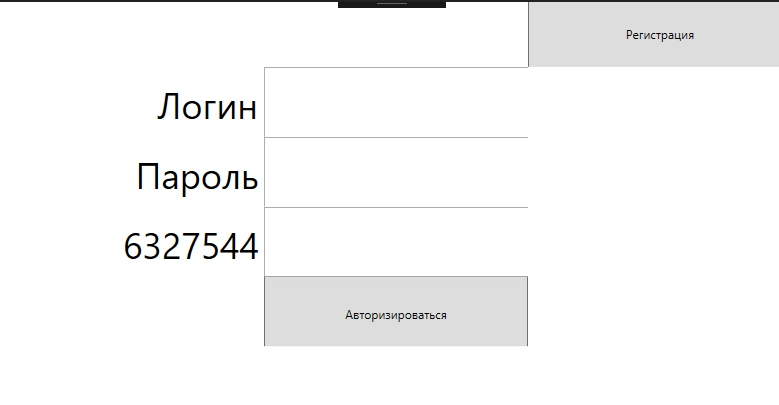


Рисунок 10 – страница авторизации

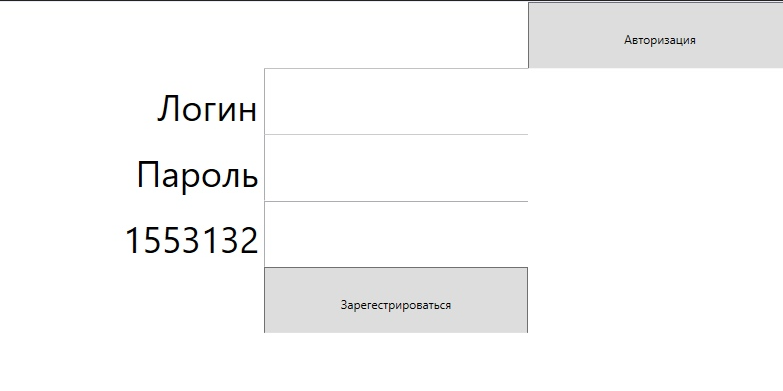


Рисунок 11 – страница регистрации

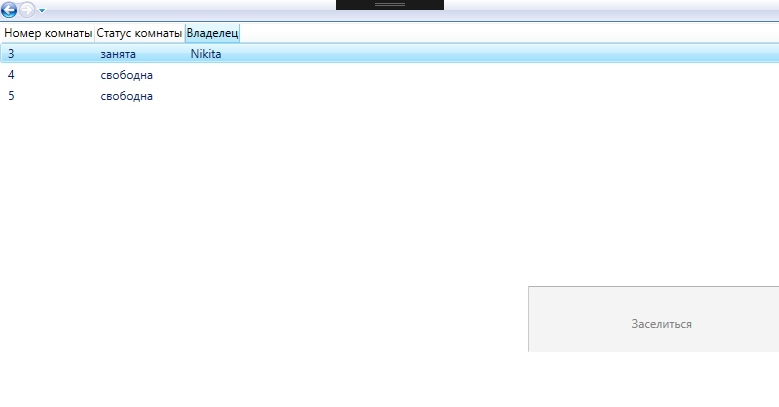


Рисунок 12 – панель постояльца

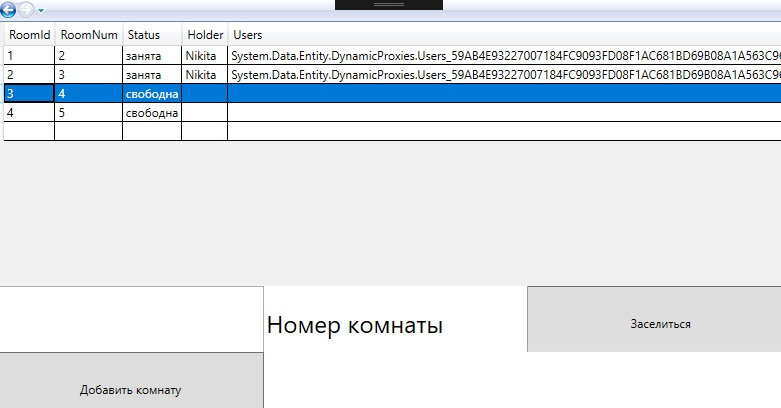


Рисунок 13 – панель администратора

2.7 Инструкция пользователю

Для начала использования приложения следует пройти регистрацию, выбрав одну из ролей. После завершения процесса регистрации пользователю будут доступны все возможности, которые обеспечивает его роль. Чтобы повторно войти в приложение следует пройти авторизацию. При вводе некорректных данных будет выведено сообщение об ошибке.

2.7.1 Общие сведения об информационной системе

Информационная система разработана для авиакомпании. Язык интерфейса – русский.

2.7.2 Требования к техническим средствам

Необходимо иметь локальный компьютер с установленной Microsoft SQL Management Studio, в которой необходимо создать базу данных приложения используя скрипт.

2.7.3 Требования к программным средствам

Для использования программы на локальном компьютере пользователя должны быть установлены Microsoft SQL Management Studio 2017 или выше, а так же сама программа и база данных, приложенная к ней.

2.7.4 Настройка информационной системы

Информационная система настроена для интуитивного взаимодействия с пользователем. Интерфейс будет дорабатываться и упрощаться в соответствии с критикой и в целях достижения максимального удобства.

2.7.5 Формы ввода

В приложении присутствуют такие формы ввода как:

* регистрация. Содержит поля логин, пароль. требуется ввести капчу.
* авторизация. Содержит поля логин и пароль, требуется ввести капчу
* Панель администратора. Содержит таблицу для редактирования базы данных, поля для ввода номера добавляемого номера в отеле, а также 2 кнопки: кнопка для выписки из номера, кнопка для создания номера
* Панель постояльца. Содержит таблицу и кнопку для снятия номера

2.7.6 Отчеты

# ЗАКЛЮЧЕНИЕ

Была проведена работа в ходе которой было спроектировано и создано приложение для снятия номера в отеле, а также возможности добавлять информацию для осуществления этого. В ходе разработки были получены практические навыки написания систем снятия номера в отеле. Разработанная система может быть усовершенствована с целью охвата большего количества полезных постояльцу, отелю функций. Были выполнены следующие цели проекта:

* улучшение навыков проектирования баз данных и архитектуры приложений,
* создание базы данных для системы,
* написание информационной системы для авиакомпании.

Подводя итоги необходимо сказать, что приложение и база данных были успешно выполнены, протестированы и готовы к работе. Во время выполнения проекта возникало множество проблем, которые были успешно решены.

# СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Руководство компании Microsoft по .NET Framework [Электронный ресурс]. – 2020 – URL: https://docs.microsoft.com/ru-ru/dotnet/framework/ (дата обращения 15.04.2020),
2. Руководство компании Microsoft по Entity Framework [Электронный ресурс]. – 2020 – URL: https://docs.microsoft.com/ru-ru/ef/ (дата обращения 05.03.2020),

# ПРИЛОЖЕНИЕ А SQL скрипты на создание и заполнение базы данных

Все скрипты на создание и заполнение базы данных доступны по ссылке на GitHub <https://github.com/Ireegorik/KyrsovoeBD/tree/master/Documents>.

USE [master]

GO

/\*\*\*\*\*\* Object: Database [HotelBD] Script Date: 30.01.2021 19:33:39 \*\*\*\*\*\*/

CREATE DATABASE [HotelBD]

CONTAINMENT = NONE

ON PRIMARY

( NAME = N'HotelBD', FILENAME = N'D:\Microsoft SQL Server\MSSQL14.SQLEXPRESS\MSSQL\DATA\HotelBD.mdf' , SIZE = 8192KB , MAXSIZE = UNLIMITED, FILEGROWTH = 65536KB )

LOG ON

( NAME = N'HotelBD\_log', FILENAME = N'D:\Microsoft SQL Server\MSSQL14.SQLEXPRESS\MSSQL\DATA\HotelBD\_log.ldf' , SIZE = 8192KB , MAXSIZE = 2048GB , FILEGROWTH = 65536KB )

GO

ALTER DATABASE [HotelBD] SET COMPATIBILITY\_LEVEL = 140

GO

IF (1 = FULLTEXTSERVICEPROPERTY('IsFullTextInstalled'))

begin

EXEC [HotelBD].[dbo].[sp\_fulltext\_database] @action = 'enable'

end

GO

ALTER DATABASE [HotelBD] SET ANSI\_NULL\_DEFAULT OFF

GO

ALTER DATABASE [HotelBD] SET ANSI\_NULLS OFF

GO

ALTER DATABASE [HotelBD] SET ANSI\_PADDING OFF

GO

ALTER DATABASE [HotelBD] SET ANSI\_WARNINGS OFF

GO

ALTER DATABASE [HotelBD] SET ARITHABORT OFF

GO

ALTER DATABASE [HotelBD] SET AUTO\_CLOSE OFF

GO

ALTER DATABASE [HotelBD] SET AUTO\_SHRINK OFF

GO

ALTER DATABASE [HotelBD] SET AUTO\_UPDATE\_STATISTICS ON

GO

ALTER DATABASE [HotelBD] SET CURSOR\_CLOSE\_ON\_COMMIT OFF

GO

ALTER DATABASE [HotelBD] SET CURSOR\_DEFAULT GLOBAL

GO

ALTER DATABASE [HotelBD] SET CONCAT\_NULL\_YIELDS\_NULL OFF

GO

ALTER DATABASE [HotelBD] SET NUMERIC\_ROUNDABORT OFF

GO

ALTER DATABASE [HotelBD] SET QUOTED\_IDENTIFIER OFF

GO

ALTER DATABASE [HotelBD] SET RECURSIVE\_TRIGGERS OFF

GO

ALTER DATABASE [HotelBD] SET DISABLE\_BROKER

GO

ALTER DATABASE [HotelBD] SET AUTO\_UPDATE\_STATISTICS\_ASYNC OFF

GO

ALTER DATABASE [HotelBD] SET DATE\_CORRELATION\_OPTIMIZATION OFF

GO

ALTER DATABASE [HotelBD] SET TRUSTWORTHY OFF

GO

ALTER DATABASE [HotelBD] SET ALLOW\_SNAPSHOT\_ISOLATION OFF

GO

ALTER DATABASE [HotelBD] SET PARAMETERIZATION SIMPLE

GO

ALTER DATABASE [HotelBD] SET READ\_COMMITTED\_SNAPSHOT OFF

GO

ALTER DATABASE [HotelBD] SET HONOR\_BROKER\_PRIORITY OFF

GO

ALTER DATABASE [HotelBD] SET RECOVERY SIMPLE

GO

ALTER DATABASE [HotelBD] SET MULTI\_USER

GO

ALTER DATABASE [HotelBD] SET PAGE\_VERIFY CHECKSUM

GO

ALTER DATABASE [HotelBD] SET DB\_CHAINING OFF

GO

ALTER DATABASE [HotelBD] SET FILESTREAM( NON\_TRANSACTED\_ACCESS = OFF )

GO

ALTER DATABASE [HotelBD] SET TARGET\_RECOVERY\_TIME = 60 SECONDS

GO

ALTER DATABASE [HotelBD] SET DELAYED\_DURABILITY = DISABLED

GO

ALTER DATABASE [HotelBD] SET QUERY\_STORE = OFF

GO

USE [HotelBD]

GO

/\*\*\*\*\*\* Object: Table [dbo].[Rooms] Script Date: 30.01.2021 19:33:39 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Rooms](

[Room\_Id] [int] NOT NULL,

[Room\_Num] [varchar](10) NOT NULL,

[Status] [varchar](50) NOT NULL,

[Holder] [varchar](40) NULL,

CONSTRAINT [PK\_Rooms] PRIMARY KEY CLUSTERED

(

[Room\_Id] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

/\*\*\*\*\*\* Object: Table [dbo].[Users] Script Date: 30.01.2021 19:33:39 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Users](

[Login] [varchar](40) NOT NULL,

[Password] [varchar](40) NOT NULL,

[Role] [int] NOT NULL,

CONSTRAINT [PK\_users] PRIMARY KEY CLUSTERED

(

[Login] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

INSERT [dbo].[Rooms] ([Room\_Id], [Room\_Num], [Status], [Holder]) VALUES (1, N'2', N'занята', N'Nikita')

INSERT [dbo].[Rooms] ([Room\_Id], [Room\_Num], [Status], [Holder]) VALUES (2, N'3', N'занята', N'Nikita')

INSERT [dbo].[Rooms] ([Room\_Id], [Room\_Num], [Status], [Holder]) VALUES (3, N'4', N'свободна', NULL)

INSERT [dbo].[Rooms] ([Room\_Id], [Room\_Num], [Status], [Holder]) VALUES (4, N'5', N'свободна', NULL)

GO

INSERT [dbo].[Users] ([Login], [Password], [Role]) VALUES (N'1', N'1', 1)

INSERT [dbo].[Users] ([Login], [Password], [Role]) VALUES (N'Nikita', N'0', 0)

GO

ALTER TABLE [dbo].[Rooms] WITH CHECK ADD CONSTRAINT [FK\_Rooms\_Users] FOREIGN KEY([Holder])

REFERENCES [dbo].[Users] ([Login])

ON UPDATE CASCADE

ON DELETE CASCADE

GO

ALTER TABLE [dbo].[Rooms] CHECK CONSTRAINT [FK\_Rooms\_Users]

GO

USE [master]

GO

ALTER DATABASE [HotelBD] SET READ\_WRITE

GO

# ПРИЛОЖЕНИЕ Б Код программы

Весь код приложения доступен по ссылке на GitHub <https://github.com/Ireegorik/KyrsovoeBD>.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

using static Hotel.Holder;

namespace Hotel

{

/// <summary>

/// Логика взаимодействия для MainWindow.xaml

/// </summary>

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

MainFrame = MF;

MainFrame.Navigate(new Page.Auth());

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Controls;

namespace Hotel

{

static class Holder

{

public static Frame MainFrame;

public static HotelBDEntities bd = new HotelBDEntities();

public static Users user;

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

using static Hotel.Holder;

namespace Hotel.Page

{

/// <summary>

/// Логика взаимодействия для TakeRoom.xaml

/// </summary>

public partial class TakeRoom : System.Windows.Controls.Page

{

public TakeRoom()

{

InitializeComponent();

Refresh();

}

private void RoomList\_SelectionChanged(object sender, SelectionChangedEventArgs e)

{

Rooms r = (Rooms)RoomList.SelectedItem;

if(r.Status == "свободна")

{

Zas.IsEnabled = true;

}

else

{

Zas.IsEnabled = false;

}

}

void Refresh()

{

List<Rooms> rooms = new List<Rooms>();

foreach (Rooms r in bd.Rooms.ToList())

{

if (r.Status == "свободна") rooms.Add(r); else if (r.Holder == user.Login) rooms.Add(r);

}

RoomList.ItemsSource = rooms;

}

private void Zas\_Click(object sender, RoutedEventArgs e)

{

Rooms r = (Rooms)RoomList.SelectedItem;

Rooms d = bd.Rooms.Find(r.Room\_Id);

d.Users = user;

d.Status = "занята";

bd.SaveChanges();

Refresh();

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

using static Hotel.Holder;

namespace Hotel.Page

{

/// <summary>

/// Логика взаимодействия для Reg.xaml

/// </summary>

public partial class Reg : System.Windows.Controls.Page

{

public Reg()

{

InitializeComponent();

Random random = new Random();

Captch.Content = random.Next(9999999).ToString();

}

private void auths(object sender, RoutedEventArgs e)

{

MainFrame.Content = new Page.Auth();

}

private void regis(object sender, RoutedEventArgs e)

{

if ((string)Captch.Content == CaptchaT.Text)

{

if (bd.Users.Find(Login.Text) == null)

{

user = new Users { Login = Login.Text, Password = Password.Text, Role = 0 };

bd.Users.Add(user);

bd.SaveChanges();

MainFrame.Content = new Page.TakeRoom();

}

}

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

using static Hotel.Holder;

namespace Hotel.Page

{

/// <summary>

/// Логика взаимодействия для Auth.xaml

/// </summary>

public partial class Auth : System.Windows.Controls.Page

{

public Auth()

{

InitializeComponent();

Random random = new Random();

Captch.Content = random.Next(9999999).ToString();

}

private void Regis(object sender, RoutedEventArgs e)

{

MainFrame.Content = new Page.Reg();

}

private void auths(object sender, RoutedEventArgs e)

{

if ((string)Captch.Content == CaptchaT.Text)

{

Users u = bd.Users.Find(Login.Text);

if ((u != null)&&(u.Password ==Password.Text))

{

if (u.Role == 0) { user = u; MainFrame.Content = new Page.TakeRoom(); } else { user = u; MainFrame.Content = new Page.AddRoom();}

}

}

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

using static Hotel.Holder;

namespace Hotel.Page

{

/// <summary>

/// Логика взаимодействия для AddRoom.xaml

/// </summary>

public partial class AddRoom : System.Windows.Controls.Page

{

public AddRoom()

{

InitializeComponent();

DataRoom.ItemsSource = bd.Rooms.ToList();

}

private void RoomList\_SelectionChanged(object sender, SelectionChangedEventArgs e)

{

Zas.IsEnabled = true;

}

private void Addrooms(object sender, RoutedEventArgs e)

{

if (bd.Rooms.Count() > 0)

{

int id = bd.Rooms.ToList()[bd.Rooms.Count() - 1].Room\_Id + 1;

bd.Rooms.Add(new Rooms { Room\_Num = NumR.Text, Status = "свободна", Room\_Id = id});

}

else

bd.Rooms.Add(new Rooms { Room\_Num = NumR.Text, Status = "свободна", Room\_Id = 0 });

bd.SaveChanges();

DataRoom.ItemsSource = bd.Rooms.ToList();

}

private void Visel(object sender, RoutedEventArgs e)

{

if(DataRoom.SelectedItem != null)

{

bd.Rooms.Find((Rooms)DataRoom.SelectedItem).Status = "свободна";

}

}

}

}

XAML

<Application x:Class="Hotel.App"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:local="clr-namespace:Hotel"

StartupUri="MainWindow.xaml">

<Application.Resources>

</Application.Resources>

</Application>

<Window x:Class="Hotel.MainWindow"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:local="clr-namespace:Hotel"

mc:Ignorable="d"

Title="MainWindow" Height="450" Width="800">

<Grid>

<Frame Name="MF" Content="Frame" Height="419" Width="792" VerticalContentAlignment="Stretch" HorizontalContentAlignment="Stretch"/>

</Grid>

</Window>

<Page x:Class="Hotel.Page.Auth"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

mc:Ignorable="d"

d:DesignHeight="450" d:DesignWidth="800"

Title="Auth">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="100\*"/>

<RowDefinition Height="100\*"/>

<RowDefinition Height="100\*"/>

<RowDefinition Height="100\*"/>

<RowDefinition Height="100\*"/>

<RowDefinition Height="100\*"/>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="100\*"/>

<ColumnDefinition Width="100\*"/>

<ColumnDefinition Width="100\*"/>

</Grid.ColumnDefinitions>

<Label Content="Логин" Grid.Row="1" Height="75" Width="262" VerticalContentAlignment="Center" HorizontalContentAlignment="Right" FontSize="36"/>

<TextBox Name="Login" Grid.Column="1" Height="75" Grid.Row="1" TextWrapping="Wrap" Text="" Width="266"/>

<TextBox Name="Password" Grid.Column="1" Height="75" Grid.Row="2" TextWrapping="Wrap" Text="" Width="266"/>

<TextBox Name="CaptchaT" Grid.Column="1" Height="75" Grid.Row="3" TextWrapping="Wrap" Text="" Width="266"/>

<Label Content="Пароль" Grid.Row="2" Height="75" Width="262" VerticalContentAlignment="Center" HorizontalContentAlignment="Right" FontSize="36" Margin="2,0,3,0"/>

<Label Name="Captch" Content="" Grid.Row="3" Height="75" Width="262" VerticalContentAlignment="Center" HorizontalContentAlignment="Right" FontSize="36" Margin="2,0,3,0"/>

<Button Content="Авторизироваться" Grid.Row="4" Height="75" Grid.Column="1" Click="auths"/>

<Button Content="Регистрация" Grid.Column="2" Height="75" Click="Regis"/>

</Grid>

</Page>

<Page x:Class="Hotel.Page.Reg"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:local="clr-namespace:Hotel.Page"

mc:Ignorable="d"

d:DesignHeight="450" d:DesignWidth="800"

Title="Reg">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="100\*"/>

<RowDefinition Height="100\*"/>

<RowDefinition Height="100\*"/>

<RowDefinition Height="100\*"/>

<RowDefinition Height="100\*"/>

<RowDefinition Height="100\*"/>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="100\*"/>

<ColumnDefinition Width="100\*"/>

<ColumnDefinition Width="100\*"/>

</Grid.ColumnDefinitions>

<Label Content="Логин" Grid.Row="1" Height="75" Width="262" VerticalContentAlignment="Center" HorizontalContentAlignment="Right" FontSize="36"/>

<TextBox Name="Login" Grid.Column="1" Height="75" Grid.Row="1" TextWrapping="Wrap" Text="" Width="266"/>

<TextBox Name="Password" Grid.Column="1" Height="75" Grid.Row="2" TextWrapping="Wrap" Text="" Width="266"/>

<TextBox Name="CaptchaT" Grid.Column="1" Height="75" Grid.Row="3" TextWrapping="Wrap" Text="" Width="266"/>

<Label Content="Пароль" Grid.Row="2" Height="75" Width="262" VerticalContentAlignment="Center" HorizontalContentAlignment="Right" FontSize="36" Margin="2,0,3,0"/>

<Label Name="Captch" Content="" Grid.Row="3" Height="75" Width="262" VerticalContentAlignment="Center" HorizontalContentAlignment="Right" FontSize="36" Margin="2,0,3,0"/>

<Button Content="Зарегестрироваться" Grid.Row="4" Height="75" Grid.Column="1" Click="regis"/>

<Button Content="Авторизация" Grid.Column="2" Height="75" Click="auths"/>

</Grid>

</Page>

<Page x:Class="Hotel.Page.TakeRoom"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:local="clr-namespace:Hotel.Page"

mc:Ignorable="d"

d:DesignHeight="450" d:DesignWidth="800"

Title="TakeRoom">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="100\*"/>

<RowDefinition Height="100\*"/>

<RowDefinition Height="100\*"/>

<RowDefinition Height="100\*"/>

<RowDefinition Height="100\*"/>

<RowDefinition Height="100\*"/>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="100\*"/>

<ColumnDefinition Width="100\*"/>

<ColumnDefinition Width="100\*"/>

</Grid.ColumnDefinitions>

<ListView Name="RoomList" HorizontalAlignment="Left" Height="300" Grid.RowSpan="4" VerticalAlignment="Top" Width="800" Grid.ColumnSpan="3" SelectionChanged="RoomList\_SelectionChanged">

<ListView.View>

<GridView>

<GridView.ColumnHeaderContextMenu>

<ContextMenu/>

</GridView.ColumnHeaderContextMenu>

<GridViewColumn Header="Номер комнаты" DisplayMemberBinding="{Binding Path=Room\_Num}"/>

<GridViewColumn Header="Статус комнаты" DisplayMemberBinding="{Binding Path=Status}"/>

<GridViewColumn Header="Владелец" DisplayMemberBinding="{Binding Path=Holder}"/>

</GridView>

</ListView.View>

</ListView>

<Button Click="Zas\_Click" Name="Zas" Content="Заселиться" Grid.Column="1" HorizontalAlignment="Left" Margin="266.333,0,0,0" Grid.Row="4" VerticalAlignment="Top" Width="267" Grid.ColumnSpan="2" Height="75" IsEnabled="False"/>

</Grid>

</Page>

<Page x:Class="Hotel.Page.AddRoom"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:local="clr-namespace:Hotel.Page"

mc:Ignorable="d"

d:DesignHeight="450" d:DesignWidth="800"

Title="AddRoom">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="100\*"/>

<RowDefinition Height="100\*"/>

<RowDefinition Height="100\*"/>

<RowDefinition Height="100\*"/>

<RowDefinition Height="100\*"/>

<RowDefinition Height="100\*"/>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="100\*"/>

<ColumnDefinition Width="100\*"/>

<ColumnDefinition Width="100\*"/>

</Grid.ColumnDefinitions>

<Button Click="Visel" Name="Zas" Content="Заселиться" Grid.Column="1" HorizontalAlignment="Left" Margin="266.333,0,0,0" Grid.Row="4" VerticalAlignment="Top" Width="267" Grid.ColumnSpan="2" Height="75"/>

<DataGrid Name="DataRoom" HorizontalAlignment="Left" Height="300" Grid.RowSpan="4" VerticalAlignment="Top" Width="800" Grid.ColumnSpan="3"/>

<TextBox Name="NumR" HorizontalAlignment="Left" Height="75" Grid.Row="4" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="267" Grid.ColumnSpan="2"/>

<Button Click="Addrooms" Content="Добавить комнату" HorizontalAlignment="Left" Grid.Row="5" VerticalAlignment="Top" Width="267" Grid.ColumnSpan="2" Height="75"/>

<Label Content="Номер комнаты" Grid.Column="1" HorizontalAlignment="Left" Margin="0.333,0,0,0" Grid.Row="4" VerticalAlignment="Top" Height="75" Width="266" VerticalContentAlignment="Center" FontSize="24"/>

</Grid>