Министерство образования Пензенской области

Государственное автономное профессиональное образовательное учреждение Пензенской области «Пензенский колледж информационных и промышленных технологий (ИТ-колледж)»

Учебный комплекс информационных технологий

**КУРСОВОЙ ПРОЕКТ**

по МДК 11.01

«Технология разработки и защиты баз данных»

**Тема:** **«Система бронирования и аренды номеров отелей»**

|  |  |
| --- | --- |
| **Допускаю**  курсовой проект к защите  на промежуточной аттестации  «\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2021 г.  Зам. директора по ООП  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Волобуева Е.А. | Выполнил: Абрамов Олег Евгеньевич |
|  |
| Обучающийся 3 курса, группы 19ИТ17 |
| По специальности 09.02.07  Информационные системы и программирование, |
| очной формы обучения |
| Руководитель курсового проекта |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_Лукьянова Галина Степановна |
|  |
| Оценка\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

Пенза, 2021 г.

Министерство образования Пензенской области

ГАПОУ ПО

«Пензенский колледж информационных и промышленных технологий

(ИТ-колледж)»

**РАССМОТРЕНО УТВЕРЖДЕНО**

Заседанием цикловой комиссии Председатель цикловой комиссии

Протокол №2 от «\_\_\_» \_\_\_\_\_\_2021 г. \_\_\_\_\_\_\_\_\_\_\_\_Н.А. Стенькина

« » \_\_\_\_\_\_\_\_\_\_ 2021 г.

**ЗАДАНИЕ**

на курсовое проектирование по дисциплине МДК 11.01

**«Поддержка и тестирование программного обеспечения»**

Студента гр. 19ИТ17

Абрамов Олег Евгеньевич

Тема: **Система бронирования и аренды номеров отелей**

**Содержание курсового проекта**

Курсовой проект включает в себя пояснительную записку

**Пояснительная записка**

**КП.09.02.07.ПКИПТ.335 - С.39.21 ПЗ**

Введение

Общий раздел

Постановка задачи

Выбор и обоснование средств и методов разработки

Специальный раздел

Системный анализ предметной области

Проектирование инфологической (концептуальной) модели предметной области

Преобразование инфологический (концептуальной) модели предметной области в реляционную модель данных

Проектирование физической структуры базы данных

Описание реализации проекта в среде СУБД

Разработка интерфейса базы данных и приложения в среде программирования

Тестирование разработанного приложения

Руководство пользователя

Преподаватель Лукьянова Г.С

**График выполнения курсового проекта**

|  |  |  |  |
| --- | --- | --- | --- |
| **Этапы работы** | **Последовательность выполнение проекта** | **Срок выполнения (неделя)** | **Примечание** |
| **Пояснительная записка (разделы)** |
| 1 | Введение | 1 | 2 страницы |
| 2 | Общий раздел | 1 | 2 страницы |
| 3 | Постановка задачи | 1 | 1 страница |
| 4 | Описание программных и аппаратных средств | 1 | 1 страница |
| 5 | Специальный раздел | 1 |  |
| 6 | Системный анализ предметной области | 2 | 2 страницы |
| 7 | Проектирование инфологической (концептуальной) модели предметной области | 1 | 2 страницы |
| 8 | Преобразование инфологический (концептуальной) модели предметной области в реляционную модель данных | 1 | 2 страницы |
| 9 | Проектирование физической структуры базы данных | 2 | 2 страницы |
| 10 | Описание реализации проекта в среде СУБД | 2 | 4 страниц |
| 11 | Разработка интерфейса базы данных и приложения в среде программирования | 1 | 10 страниц |
| 12 | Тестирование разработанного приложения | 1 | 9 страница |
| 13 | Руководство пользователя | 1 | 3 страницы |

Дата выдачи задания: «1» 10 2021 г.

Срок выполнения: « » 2021 г.

Дата защиты: « » 2021 г.

Задание получено: «1» 10 2021 г.

\_\_\_\_\_\_\_\_\_\_\_\_\_/\_\_\_\_\_\_\_\_\_\_\_\_\_/

**Содержание**

Изм.

Лист

№ докум.

Подпись

Дата

Лист

3

КП.09.02.07.ПКИПТ.335 - С.39.21 ПЗ

Разраб.

Абрамов О.Е.

Провер.

Лукьянова Г.С.

Т. контр

Н. контр.

Ж.Е.

Утвержд.

Стенькина Н.А.

Система бронирования и аренды номеров отелей

Пояснительная записка

Лит.

Листов

51

**19ИТ17**

Введение 4

1. Общий раздел 6
   1. Постановка задачи 6

1.2 Описание программных и аппаратных средств разработки 7

1. Специальный раздел 8
   1. Системный анализ предметной области 8
   2. Проектирование инфологической (концептуальной) модели

предметной области 10

* 1. Преобразование инфологический (концептуальной) модели

предметной области в реляционную модель данных 12

2.4 Проектирование физической структуры баз данных 14

2.5 Описание реализации проекта в среде СУБД 16

2.6 Разработка интерфейса базы данных и приложения

в среде программирования 20

2.7 Тестирование разработанного приложения 30

2.8 Руководство пользователя 39

Заключение 42

Список использованной литературы 43

Приложение А – ER-диаграмма

Приложение Б – Диаграмма вариантов использования

Приложение В – Листинг с исходным кодом и результаты

**Введение**

Изм.

Лист

№ докум.

Подпись

Дата

Лист

4

КП.09.02.07.ПКИПТ.335 - С.39.21 ПЗ

Планируя собственное путешествие, необходимо позаботиться обо всех возможных мелочах, дабы сделать поездку максимально комфортной и интересной. Особое значение имеет система бронирования и аренды номеров отелей, в качестве гарантии успешного пребывания в отеле. Заблаговременное бронирование или аренда отеля становится не чем иным, как наиболее явственным и живым доказательством возможности быть полностью уверенным в том, что по приезду гостя не будут подстерегать никакие трудности и неприятности. Ведь именно крыша над головой имеет максимальное значение. Из года в год все больше туристских фирм и отдельных граждан используют системы бронирования гостиниц. Такие системы позволяют бронировать места в гостиницах всего мира, получать свежую информацию об условиях бронирования, ценах, скидках за минимальное время, не выходя из офиса или дома. В настоящее время система бронирования и аренды номеров отелей является актуальной из-за растущего спроса на жилье в отпуске и появления новых мест курортов. Более того, уже существующее программное обеспечение может быть усовершенствовано и должно совершенствоваться для оптимизации работы предприятия.

Целью данного курсового проекта является создание программного продукта для автоматизации бронирования и аренды номеров отелей. Для достижения цели необходимо решить следующие задачи:

* изучить научно-техническую и справочную литературу по данной предметной области;
* спроектировать логику, дизайн приложения, написать программный код, создать базу данных и связать его с приложением;
* протестировать программный продукт;
* составить необходимую документацию, в т.ч. руководство пользователя, а также составить все необходимые диаграммы.

Программный продукт может быть полезен для предприятий, деятельность которых связана с бронированием и арендой номеров отелей, а также для пользователя, которому необходимо забронировать или арендовать номер в отеле. Пользуясь данным приложением, пользователи смогут забронировать или арендовать номер в отеле у себя дома или в любом удобном им месте, а предприятие отслеживать процесс бронирования или аренды номера.

Изм.

Лист

№ докум.

Подпись

Дата

Лист

5

КП.09.02.07.ПКИПТ.335 - С.39.21 ПЗ

В качестве целевой аудитории программного продукта выступают пользователи сети-Internet, которым необходимо забронировать или арендовать номер в отеле.

**1 Общий раздел**

Изм.

Лист

№ докум.

Подпись

Дата

Лист

6

КП.09.02.07.ПКИПТ.335 - С.39.21 ПЗ

* 1. **Постановка задачи**

Целью курсового проекта является разработка базы данных и приложения «Система бронирования и аренды номеров отелей». База данных должна содержать данные о клиентах, бронировании и аренды номеров отелей. Данные должны быть представлены в виде реляционной модели. База данных должна проектироваться в специальной среде – системе управления базами данных. В данном случае выбрана СУБД Microsoft SQL Server 2019. Приложение для работы с базой данных будет выполнено в виде многооконной программы. Для создания приложения будет использована интегрированная среда разработки Microsoft Visual Studio 2019.

Разработанная база данных и приложение к ней будет выполнять функцию автоматизации бронирования и аренды номеров отелей. База данных будет содержать в себе информацию о клиентах, бронировании и аренде номеров отелей.

Приложение и база данных должны быть защищены от непредвиденных сбоев и несанкционированного доступа, так как в базе будут храниться личные данные клиентов, в т.ч. паспортные данные и номера телефонов телефоны. Нужно очень хорошо защитить эти данные от утечки.

Таким образом, целью курсового проекта является разработка базы данных отеля и приложения для работы с ней. Данный программный продукт позволит автоматизировать процесс бронирования и аренды номера в отеле.

**1.2 Описание программных и аппаратных средств разработки**

Изм.

Лист

№ докум.

Подпись

Дата

Лист

7

КП.09.02.07.ПКИПТ.335 - С.39.21 ПЗ

Для разработки проекта «Система бронирования и аренды номеров отелей» использовались следующие аппаратные средства:

* процессор Intel Core i3 9990;
* видеокарта GTX 1050 Ti;
* оперативная память 8гб;
* накопитель HDD 1024 гб.

Также для разработки данного проекта были использованы следующие программные средства:

* операционная система Windows 10 Home x64;
* среда разработки MS Visual Studio Community 2019 – в этой среде создавался интерфейс программы и разрабатываться код. Помимо этого, производилась отладка и тестирование программы;
* браузер Google Chrome. Использовался для поиска информации по теме проекта.
* Microsoft Office Word 2010 – текстовый редактор, в котором выполнялось оформление курсового проекта;
* Draw.IO Desktop – десктопный инструмент для создания диаграмм с бесчисленным количеством различных заготовок, в т.ч. и UML.
* Microsoft SQL Server 2019 – программное обеспечение для построения и управления базой данных (СУБД).

Для конечного пользователя минимальные системные требования выглядят следующим образом:

* процессор Intel Celeron G530 или аналог от AMD;
* видеокарта Nvidia Geforce 5600 GT / ATI Radeon HD 7220
* оперативная память не менее 2048 мб;
* свободное место на накопителе не менее 200 мб;
* операционная система Windows 10 x64 или выше.

**2 Специальный раздел**

Изм.

Лист

№ докум.

Подпись

Дата

Лист

8

КП.09.02.07.ПКИПТ.335 - С.39.21 ПЗ

**2.1 Системный анализ предметной области**

С развитием научно-технического прогресса и, в частности, цифровых технологий все больше информации хранится не на бумажных носителях, которые имеют довольно ограниченную вместимость, а на электронных носителях, которые позволяют хранить намного большие объемы данных, более удобно структурировать их и при этом занимать меньшее пространство. Здесь очень кстати подходят базы данных, которые хранятся на удаленных серверах, и при этом позволяют пользователю просматривать и управлять данными, которые физически располагаются на большом расстоянии. Преимуществом базы данных является то, что информация в ней структурируется и распределяется по таблицам, что намного удобнее при поиске информации и работы с ней. Однако для базы данных нужно программное обеспечение, разработанное специально для работы с ней. Это могут быть как СУБД – системы управления базами данных, так и приложения, выполняющие роль связующего между пользователем и базой.

Исходя из цели данного курсового проекта, должна быть спроектирована база данных, содержащая информацию о клиентах и номерах в отеле. Это и будут таблицы будущей базы данных, в которых будет храниться эта инфомарция.

Информация о клиенте:

* идентификационный номер клиента;
* ФИО клиента;
* номер телефона клиента;
* паспортные данные клиента.

Информация о номерах:

* идентификационный номер клиента;
* тип номера;
* спальных мест;
* дней проживания;
* бронирование/аренда;
* сумма.

Изм.

Лист

№ докум.

Подпись

Дата

Лист

9

КП.09.02.07.ПКИПТ.335 - С.39.21 ПЗ

Информация о персонале:

* идентификационный номер работника;
* ФИО работника;
* номер телефона работника;
* заработанная плата работника.

Далее разрабатывается приложение, которое будет работать с описанной базой данных. Так как оно будет создано для клиентов отеля и персонала отеля, то оно должно выполнять функции программного обеспечения отеля.

Поэтому к нему предъявлены следующие функциональные требования:

* просмотр данных;
* работа с данными (в т.ч. редактирование, добавление и удаление);
* бронирование номера;
* аренда номера;
* расчет стоимости проживания в номере.

**2.2 Проектирование инфологической (концептуальной) модели предметной области**

Изм.

Лист

№ докум.

Подпись

Дата

Лист

10

КП.09.02.07.ПКИПТ.335 - С.39.21 ПЗ

Структура реляционной базы данных предполагает, что данные хранятся в таблицах, которые имеют свои атрибуты (в виде столбцов). Между таблицами в базе существуют различные реляционные связи. С помощью связей в базе происходит взаимодействие таблиц друг с другом, т.е. чтобы не работать с каждой таблицей по отдельности, а с базой в общем и целом, в ней необходимо выстроить связи.

Согласно системному анализу предметной области в базе данных существуют три объекта: клиент, номера и персонал.

Объект «Клиент» имеет следующие атрибуты:

* идентификационный номер клиента;
* номер телефона;
* ФИО;
* паспортные данные;

Объект «Номера» имеет следующие атрибуты:

* идентификационный номер клиента;
* тип номера;
* спальных мест;
* дней проживания;
* бронирование/Аренда;
* сумма.

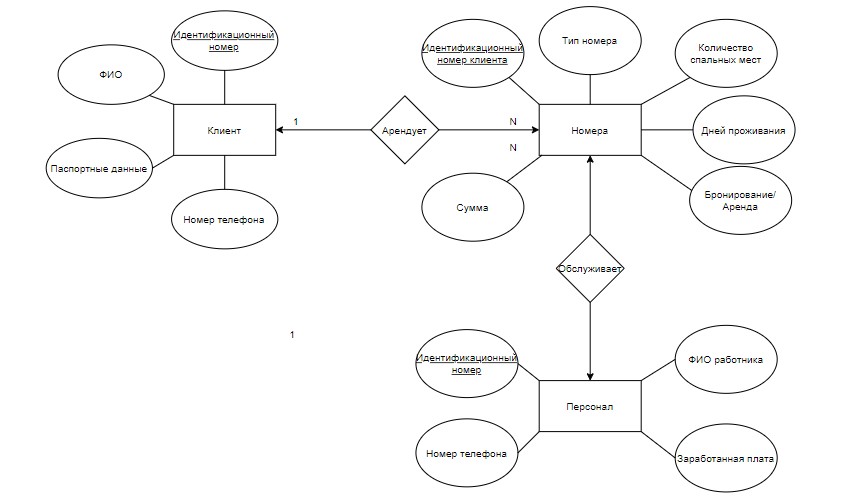
Объект «Персонал» имеет следующие атрибуты:

* идентификационный номер работника;
* ФИО работника;
* номер телефона работника;
* заработанная плата работника.

Объект «Клиент» связан с объектом «Номера» связью «Арендует». Связь один-ко-многим, т.е. один клиент может арендовать сразу несколько номеров.

Объект «Персонал» связан с объектом «Номера» связью «Обслуживает». Связь один-ко-многим, т.е. один работник может обслуживать несколько номеров.

По результатам данного раздела была разработана инфологическая (концептуальная) модель предметной области:



Изм.

Лист

№ докум.

Подпись

Дата

Лист

11

КП.09.02.07.ПКИПТ.335 - С.39.21 ПЗ

Рисунок 1 – Концептуальная модель предметной области

**2.3 Преобразование инфологический (концептуальной) модели предметной области в реляционную модель данных**

Изм.

Лист

№ докум.

Подпись

Дата

Лист

12

КП.09.02.07.ПКИПТ.335 - С.39.21 ПЗ

Для преобразования инфологической (концептуальной) модели предметной области в реляционную модель данных первым шагом необходимо составить ER-диаграмму, основанную на анализе предметной области.

Это необходимо, чтобы выделить и визуализировать основные сущности и их атрибуты, необходимые для реализации в базе данных.

Каждый объект является сущность. У каждой сущности есть свои атрибуты. Сущность «Клиент» имеет следующие атрибуты:

* идентификационный номер клиента;
* номер телефона;
* ФИО;
* паспортные данные.

Сущность «Номера» имеет следующие атрибуты:

* идентификационный номер клиента;
* тип номера;
* спальных мест;
* дней проживания;
* бронирование/Аренда;
* сумма.

Сущность «Персонал» имеет следующие атрибуты:

* идентификационный номер работника;
* ФИО работника;
* номер телефона работника;
* заработанная плата работника.

Сущности в базе данных имеют связи между собой. Сущность «клиент» связана с сущностью «номера» связью «арендует». Связь один-ко-многим, то есть один клиент может арендовать множество номеров.

Сущность «Персонал» связан с сущностью «Номера» связью «Обслуживает». Связь один-ко-многим, т.е. один работник может обслуживать несколько номеров.

Исходя из вышеизложенного, была составлена ER-диаграмма базы данных:

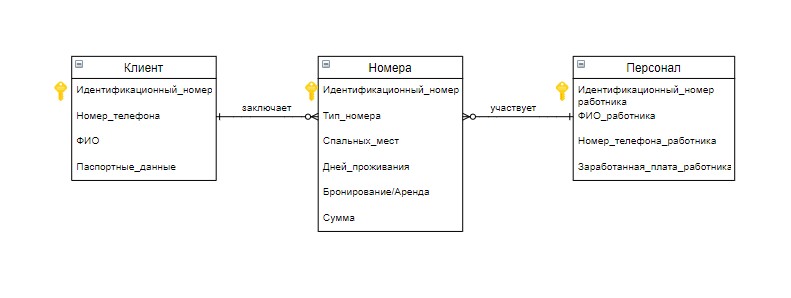
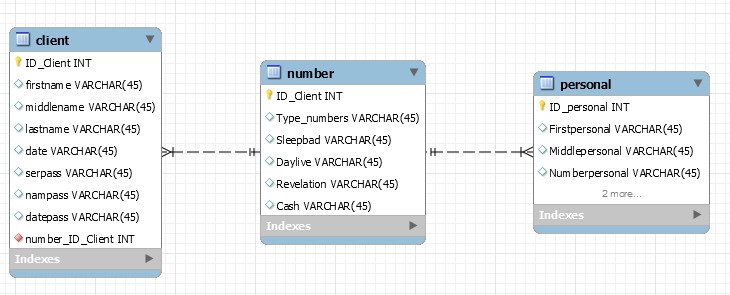


Рисунок 2 – ER-диаграмма

Для создания реляционной модели данных необходимо типизировать каждый атрибут всех сущностей. На основе этого строим реляционную модель данных :



Изм.

Лист

№ докум.

Подпись

Дата

Лист

13

КП.09.02.07.ПКИПТ.335 - С.39.21 ПЗ

Рисунок 3 – Реляционная модель данных

**2.4 Проектирование физической структуры базы данных**

В качестве системы управления базами данных выбрана Microsoft SQL Server 2019, с ней довольно просто работать и можно практически сразу после скачивания приступить к созданию баз данных.

База данных «Система бронирования и аренды номеров отелей» состоит из трех таблиц: client, number, personal.

Изм.

Лист

№ докум.

Подпись

Дата

Лист

14

КП.09.02.07.ПКИПТ.335 - С.39.21 ПЗ

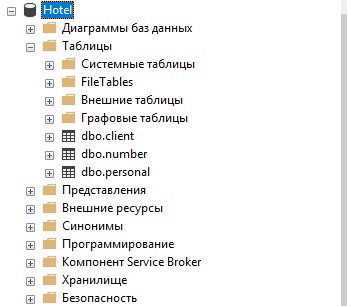


Рисунок 4 – Структура базы данных

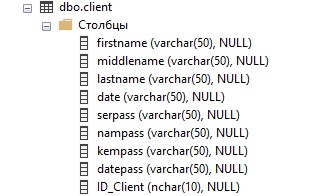


Рисунок 5 – Структура таблицы “client”

Таблица состоит из 9 столбцов. Столбец “ID\_Client” является первичным ключом.

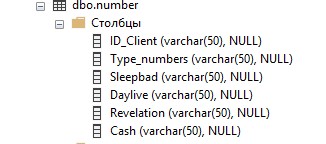
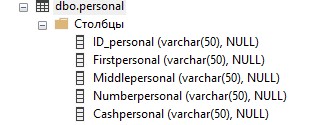


Рисунок 6 – Структура таблицы “number”

Таблица состоит из 6 столбцов. Столбец “ID\_Client” является первичным ключом.



Изм.

Лист

№ докум.

Подпись

Дата

Лист

15

КП.09.02.07.ПКИПТ.335 - С.39.21 ПЗ

Рисунок 7 – Структура таблицы “personal”

Таблица состоит из 5 столбцов. Столбец “ID\_personal” является первичным ключом.

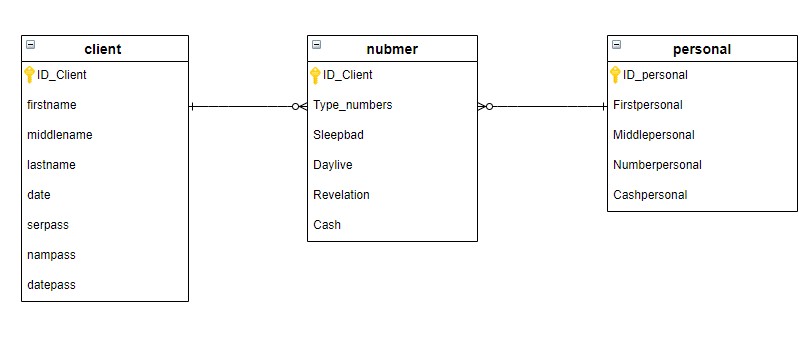


Рисунок 8 – Связи между таблицами

**2.5 Описание реализации проекта в среде СУБД**

Изм.

Лист

№ докум.

Подпись

Дата

Лист

16

КП.09.02.07.ПКИПТ.335 - С.39.21 ПЗ

Для создания базы данных использовался SQL – декларативный язык программирования для создания и управления данными в реляционной базе данных.

Для создания базы данных использовалась команда CREATE DATABASE IF NOT EXISTS `hotel` DEFAULT CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci;

Здесь прописывается название базы данных и кодировки.

Для создания таблицы “client” использовался следующий запрос: CREATE TABLE `client`

`ID\_Client` int NOT NULL,

`firstname` varchar(50) CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci NOT NULL COMMENT 'фамилия клиента ',

`middlename` varchar(50) CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci NOT NULL COMMENT 'имя клиента',

`lastname` varchar(50) CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci NOT NULL COMMENT 'отчество клиента')

`date` varchar(50) CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci NOT NULL COMMENT 'дата рождения)

`serpass` varchar(50) CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci NOT NULL COMMENT 'серия паспорта клиента')

`nampass` varchar(50) CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci NOT NULL COMMENT 'номер паспорта клиента')

`kempass` varchar(50) CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci NOT NULL COMMENT 'кем выдан паспорт клиента')

`datepass` varchar(50) CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci NOT NULL COMMENT 'дата выдачи паспорта клиента')

ENGINE=InnoDB DEFAULT CHARSET=utf8;

Для создания таблицы “number” использован запрос CREATE TABLE `nubmer` (

`ID\_Client` int NOT NULL,

`Type\_numbers` varchar(50) CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci NOT NULL COMMENT 'тип номера',

`sleepbad` varchar(50) CH1ARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci NOT NULL COMMENT 'количество спальных мест',

`Daylive` varchar(50) CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci NOT NULL COMMENT 'дней проживания',

`Revelation` varchar(50) CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci NOT NULL COMMENT 'Бронирование/Аренда') ,

Изм.

Лист

№ докум.

Подпись

Дата

Лист

17

КП.09.02.07.ПКИПТ.335 - С.39.21 ПЗ

`Cash` varchar(50) CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci NOT NULL COMMENT 'Цена проживания'.

ENGINE=InnoDB DEFAULT CHARSET=utf8;

Для создания таблицы “personal” была использована следующая команда:

CREATE TABLE `personal` (

`ID\_personal` int NOT NULL,

`firstpersonal` varchar(50) CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci NOT NULL COMMENT 'Фамилия работника',

`middlepersonal` varchar(50) CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci NOT NULL COMMENT 'Имя работника',

`numberspersonal` varchar(50) CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci NOT NULL COMMENT 'Телефон работника',

`cashpersonal` varchar(50) CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci NOT NULL COMMENT 'Заработанная плата работника',

ENGINE=InnoDB DEFAULT CHARSET=utf8;

Изм.

Лист

№ докум.

Подпись

Дата

Лист

18

КП.09.02.07.ПКИПТ.335 - С.39.21 ПЗ

Для добавления данных в таблицу “client” использовался запрос:

INSERT INTO `client` (`ID\_Client`, `firstname`,`middlename`,`lastname`, `date`,`serpass`, `nampass`,`kempass`, `datepass`) VALUES

(1, 'Иванов', 'Иван', 'Иванович', '19.03.1985', '5713455667', '8-999-612-09-86', 'ТП УМФС Рязанского района', '25.03.1999'),

(2, 'Кириллов', 'Николай', 'Васильевич', '26.05.1986', '5713454127', '8-937-622-09-46', 'ТП УМФС Владимирского района', '1.06.2000'),

(3, 'Иванов', 'Иван', 'Иванович', '19.03.1985', '5713455667', '8-999-612-09-86', 'ТП УМФС Рязанского района', '25.03.1999'),

Для добавления данных в таблицу “number” использовалась команда INSERT INTO `nubmer` (`id\_client`, `type\_numbers`, `sleepbad`, `daylive`, `revelation`, `cash`,) VALUES

(1, 'Обычный', '2', '05.06-07.06, 2дня', 'Аренда', '6000'),

(2, 'Люкс', '2', '07.07-09.07, 2дня', 'Аренда',' 9000 руб'),

(3, 'Президентский', '1', ''10.08-15.08, 5дней', 'Бронирование', '25000 руб.'),

Для добавления данных в таблицу “personal” был использован запрос INSERT INTO `personal` (`ID\_personal`, `firstpersonal`, `middlepersonal`, `numberpersonal`, `cashpersonal`) VALUES

(1, Петров, Геннадий, '9334446611', '15000');

(2, Дегтярев, Иван, '9354331411', '25000');

(3, Киров, Василий, '9613345611', '17000');

После исполнения всех запросов таблицы базы данных выглядят следующим образом:

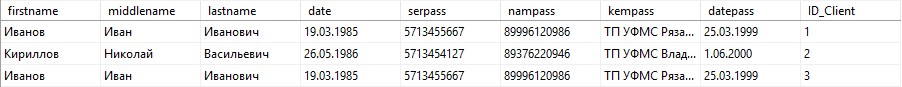


Рисунок 9 – Таблица “client”

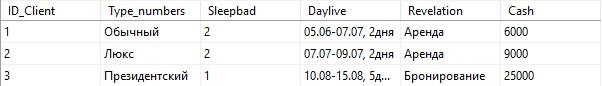


Рисунок 10 – Таблица “number”

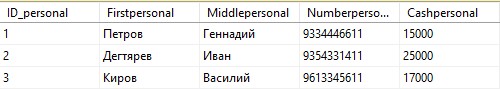


Рисунок 11 – Таблица “personal”

Изм.

Лист

№ докум.

Подпись

Дата

Лист

19

КП.09.02.07.ПКИПТ.335 - С.39.21 ПЗ

**2.6 Разработка интерфейса базы данных и приложения в среде программирования**

Изм.

Лист

№ докум.

Подпись

Дата

Лист

20

КП.09.02.07.ПКИПТ.335 - С.39.21 ПЗ

Для разработки базы данных использовалась система управления базами данных Microsoft SQL Server 2019. При разработке были использованы различные диаграммы, на основе которых составлялась база данных. Интерфейс СУБД выглядит следующим образом:

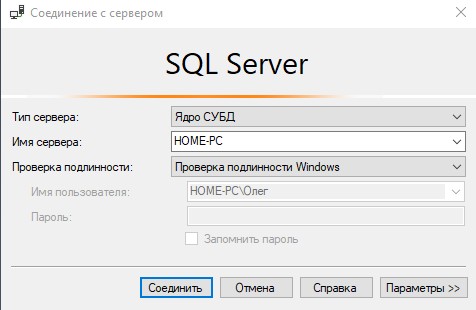


Рисунок 12 – Окно подключения к серверу

В этом окне происходит настройка подключения к серверу.

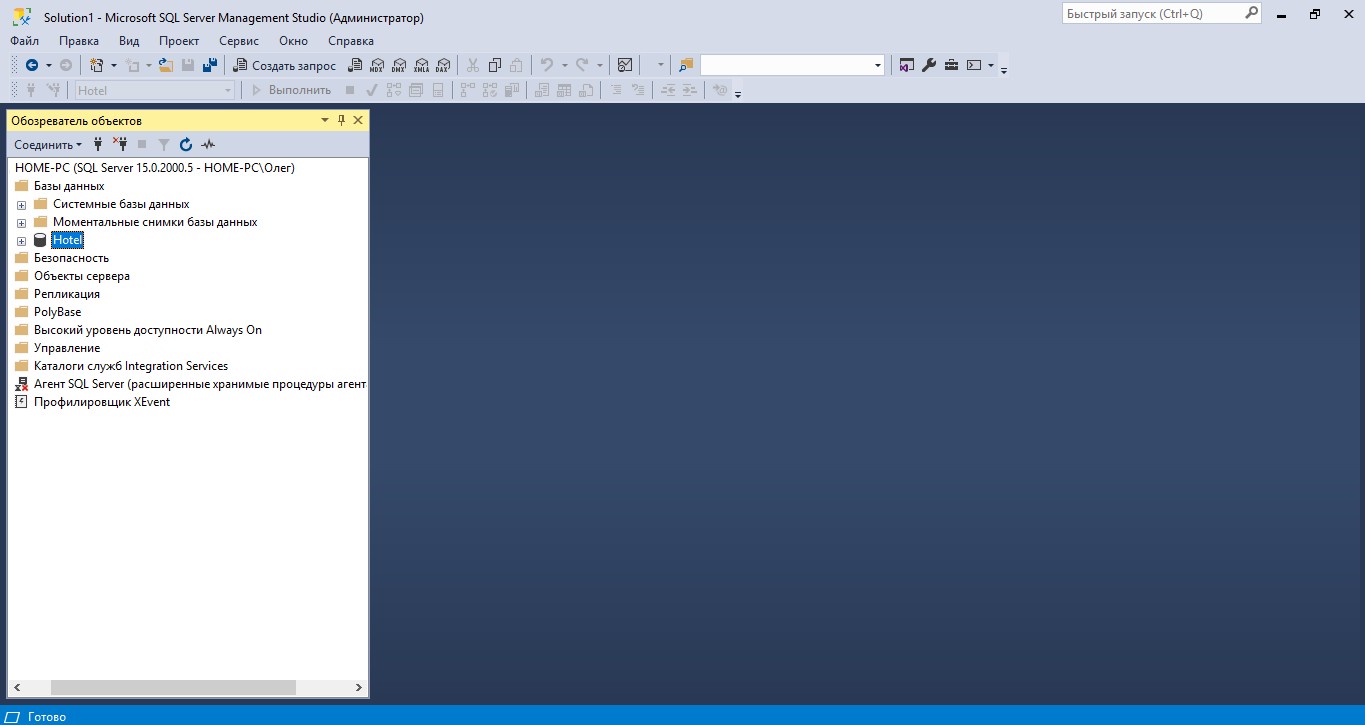


Рисунок 13 – Интерфейс СУБД

После подключения к серверу перед администратором появляется данное окно, в котором происходит вся работа над базой данных.

Изм.

Лист

№ докум.

Подпись

Дата

Лист

21

КП.09.02.07.ПКИПТ.335 - С.39.21 ПЗ

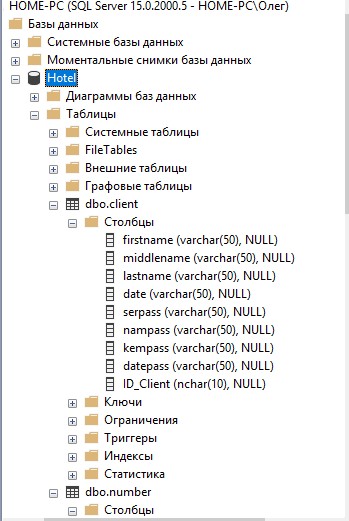


Рисунок 14 – Обозреватель объектов

Здесь содержатся все объекты СУБД.

Для того, чтобы разработать интерфейс программного продукта, необходимо сначала продумать его, представить общий план внешнего вида приложения, потом составить макеты всех его окон, страниц или вкладок, а уже после этих этапов приступить к разработке интерфейса в среде программирования.

Общий план внешнего вида представляет из себя набросок интерфейса будущего приложения без какой-либо конкретики и наименований. На рисунке 14 представлен общий вид окна главного меню:

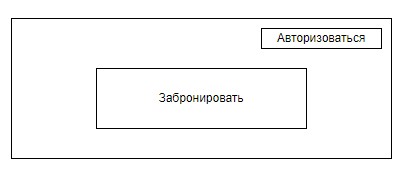
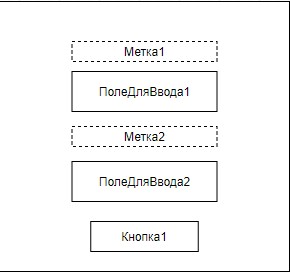


Рисунок 14 – Главное окно

После был составлен набросок окна авторизации:



Изм.

Лист

№ докум.

Подпись

Дата

Лист

22

КП.09.02.07.ПКИПТ.335 - С.39.21 ПЗ

Рисунок 15 – Общий вид окна авторизации

Далее был составлен общий вид окна взаимодействия с базой данных:

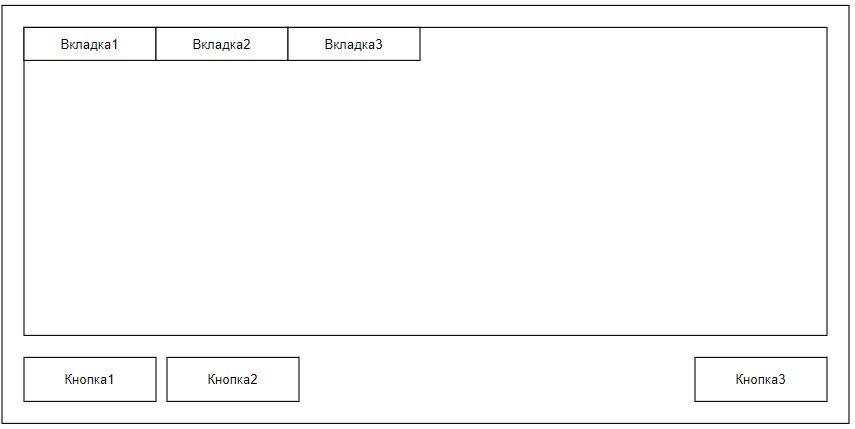


Рисунок 16 – Макет окна отображения БД

Добавление данных в таблицы базы будет происходить через отдельные окна добавления, поэтому необходимо составить макет и для них:

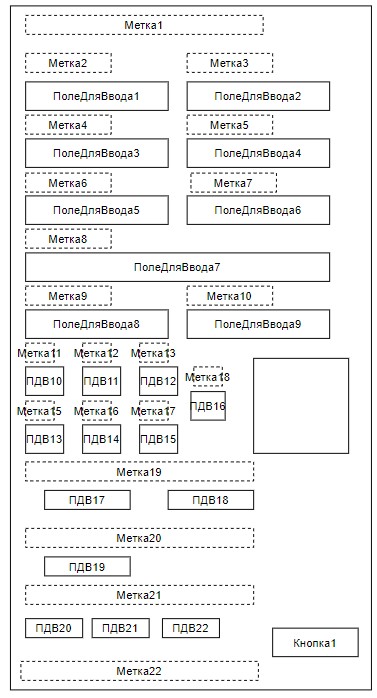


Рисунок 17 – Макет окна добавления данных

На основе предоставленных общих планов приложения и имеющегося задания уже можно составлять конкретные макеты для конкретных окон приложения.

Изм.

Лист

№ докум.

Подпись

Дата

Лист

23

КП.09.02.07.ПКИПТ.335 - С.39.21 ПЗ

Окно авторизации, как видно из вышеизображенного макета, должно быть максимально простым и иметь всего два поля для ввода, две текстовые метки для обозначения полей и кнопку. Цветовая гамма окна не должна быть резкой, яркой и т.д. Лучше всего для оформления подойдут пастельные тона любых оттенков, например, синего. Размер окна авторизации должен быть не менее 500х400 и не более 850х600 пикселей.

Первым делом был построен макет окна главного меню:

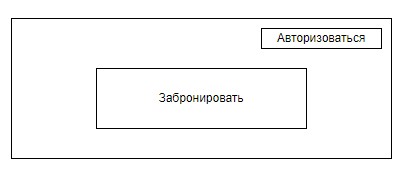


Рисунок 18 – Макет окна главного меню

Исходя из вышеперечисленных требований был построен макет окна авторизации:

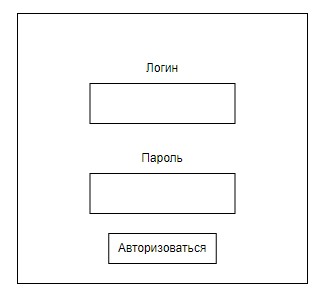


Рисунок 19 – Макет окна авторизации

Окно, в котором будет отображена база данных, тоже должно иметь минималистичный дизайн и максимальную функциональность. Основное пространство будет занимать элемент, отображающий базу данных в виде таблицы. Остальное пространство займут кнопки и вкладки. Размер окна отображения БД должен быть не менее 1000х500 пикселей. Цветовая гамма может быть схожей с цветовой гаммой окна авторизации.

Изм.

Лист

№ докум.

Подпись

Дата

Лист

25

КП.09.02.07.ПКИПТ.335 - С.39.21 ПЗ

Изм.

Лист

№ докум.

Подпись

Дата

Лист

24

КП.09.02.07.ПКИПТ.335 - С.39.21 ПЗ

Исходя из вышеизложенных требований был построен подробный макет окна отображения базы данных:

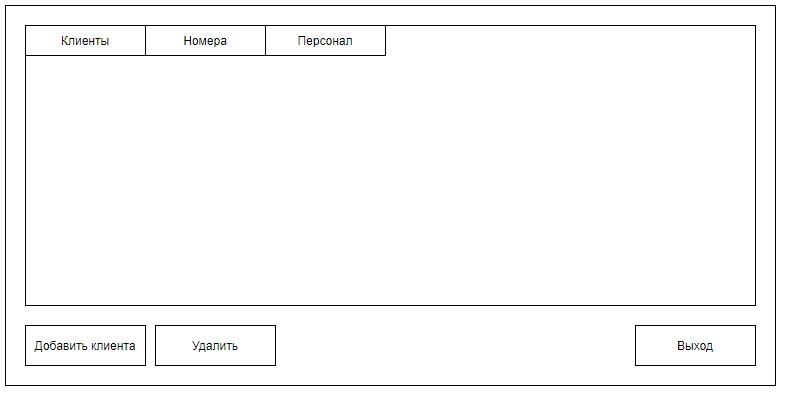


Рисунок 20 – Макет окна взаимодействия с базой данных

Окон добавления данных будет три, но они будут схожи между собой и различие будет только в количестве текстовых меток и полей для ввода. Размер окна должен варьироваться между 350х450 и 900х1000 пикселей, цветовая гамма будет выбрана такая же, как и у окна авторизации. Для проектирования хватит одного подробного макета.

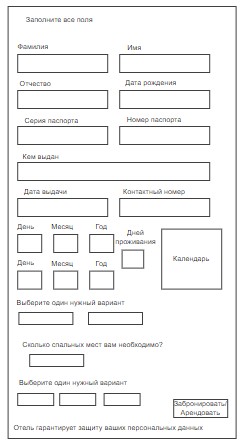


Рисунок 21 – Макет окна добавления данных

Далее необходимо по имеющимся макетам построить интерфейс приложения в среде разработки. Для выполнения данной задачи выбрана среда разработки Microsoft Visual Studio 2019. Она обладает довольно широким списком различных элементов для построения интерфейса приложения.

Изм.

Лист

№ докум.

Подпись

Дата

Лист

26

КП.09.02.07.ПКИПТ.335 - С.39.21 ПЗ

В первую очередь был построен интерфейс главного окна:

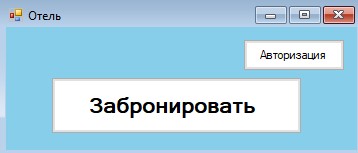
\

Рисунок 22 – Внешний вид окна главное меню

Для создания окна использовались следующие элементы:

* Label – текстовая метка;
* TextBox – поле для ввода текста;
* Button – кнопка для взаимодействия с приложением.

После главного окна был построен интерфейс окна авторизации:

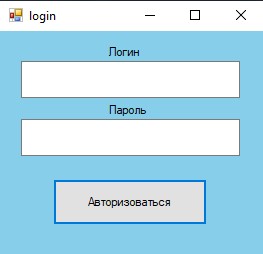
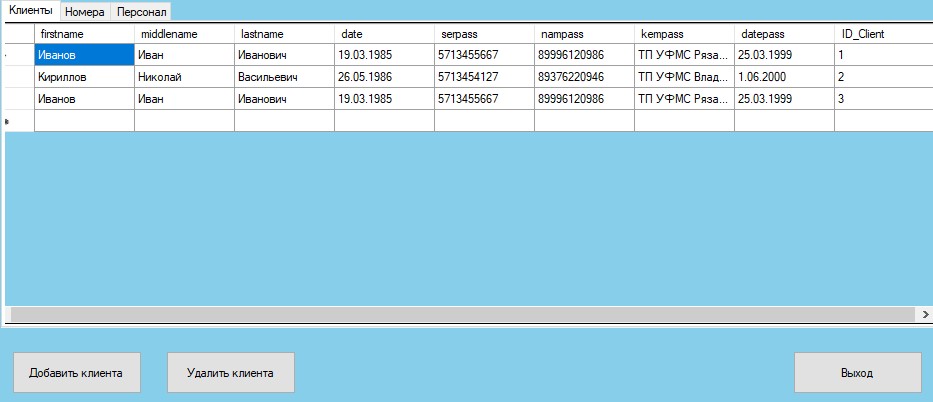


Рисунок 23 – Внешний вид окна авторизации

Для создания окна использовались следующие элементы:

* Label – текстовая метка;
* TextBox – поле для ввода текста;
* Button – кнопка для взаимодействия с приложением.

Далее было спроектировано основное окно, в котором отображается база данных:



Изм.

Лист

№ докум.

Подпись

Дата

Лист

27

КП.09.02.07.ПКИПТ.335 - С.39.21 ПЗ

Рисунок 24 – Внешний вид окна отображения БД

Окно имеет три вкладки, которые идентичны по своей структуре и различаются только содержанием.

Для создания окна использовались следующие элементы:

Изм.

Лист

№ докум.

Подпись

Дата

Лист

28

КП.09.02.07.ПКИПТ.335 - С.39.21 ПЗ

* TabPage – элемент, включающий в себя некоторое количество вкладок;
* DataGridView – таблица для отображения базы данных;
* Button – кнопка для взаимодействия с приложением.

После основного окна были спроектированы окна добавления данных в базу:

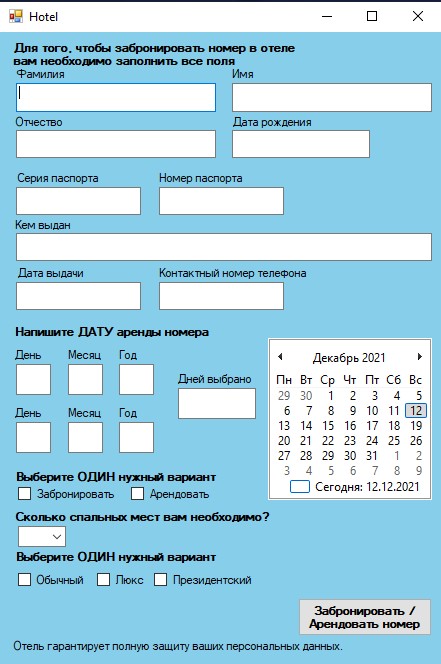


Рисунок 25 – Внешний вид окна добавления данных

Для создания окна использовались следующие элементы:

* Label – текстовая метка;

Изм.

Лист

№ докум.

Подпись

Дата

Лист

29

КП.09.02.07.ПКИПТ.335 - С.39.21 ПЗ

* TextBox – поле для ввода текста;
* ComboBox – поле с выпадающим списком;
* MonthCalendar – календарь с возможностью выбора даты;
* Button – кнопка для взаимодействия с приложением.

**2.7 Тестирование разработанного приложения**

Изм.

Лист

№ докум.

Подпись

Дата

Лист

30

КП.09.02.07.ПКИПТ.335 - С.39.21 ПЗ

Так как разработанный проект состоит из нескольких модулей, необходимо протестировать каждый модуль программы по отдельности. Для решения подобных задач больше всего подходят модульное (или unit-) и интеграционное тестирование. Данные способы тестирования может проводиться всеми возможными методами: методом черного, белого или серого ящика.

Метод черного ящика предполагает полное отсутствие у тестировщика информации о структуре программы и программном коде. Другими словами, тестирование проводится как бы от лица конечного пользователя.

Метод белого ящика подразумевает полную информированность тестировщика о структуре программы и предоставлении ему программного кода. Важно, чтобы тестировщик имел знания в выбранном языке программирования и вспомогательных средств, использованных при разработке программного продукта.

Метод серого ящика – это что-то среднее между вышеперечисленными методами, когда устройство программы известно лишь частично.

Для тестирования проекта был выбран метод белого ящика как самый подходящий для решения поставленной задачи.

Тестирование будет проводиться вручную. Средствами для тестирования являются тестовый граф, тестовые пути и обработка исключений.

В первую очередь будет проведено модульное тестирование. Это тестирование отдельных модулей программы и проверка их работоспособности дифференцировано от всех остальных частей приложения.

1. Тестируемый модуль – окно авторизации. Необходимо протестировать функцию авторизации в приложении. В базе хранятся данные о работниках отеля, которые должны иметь доступ к приложению. При авторизации приложение сравнивает введенные данные с данными в коде и либо пропускает далее, либо сообщает об ошибке.

Исходный код:

string Login = login.Text;

Изм.

Лист

№ докум.

Подпись

Дата

Лист

31

КП.09.02.07.ПКИПТ.335 - С.39.21 ПЗ

string Password = password.Text;

connector db = new connector();

DataTable table = new DataTable();

MySqlDataAdapter adapter = new MySqlDataAdapter();

MySqlCommand command = new MySqlCommand("SELECT \* FROM `user` WHERE `login` = @login AND `password` = @password", db.GetConnection());

command.Parameters.Add("@login", MySqlDbType.VarChar).Value = Login;

command.Parameters.Add("@password", MySqlDbType.VarChar).Value = Password;

adapter.SelectCommand = command;

adapter.Fill(table);

MySqlDataAdapter users = new MySqlDataAdapter($"Select role From user", db.GetConnection());

role = Convert.ToString(table.Rows[0][5]);

if (table.Rows.Count > 0)

{

ActiveForm.Hide();

general general = new general();

general.Show();

}

Else

MessageBox.Show("Пользователя не существует. Проверьте правильность введенных данных");

Для предотвращения аварийного завершения программы в коде нужно прописать специальные блоки отлова и обработки исключительных ситуаций:

try

{

string Login = login.Text;

string Password = password.Text;

connector db = new connector();

DataTable table = new DataTable();

MySqlDataAdapter adapter = new MySqlDataAdapter();

MySqlCommand command = new MySqlCommand("SELECT \* FROM `user` WHERE `login` = @login AND `password` = @password", db.GetConnection());

command.Parameters.Add("@login", MySqlDbType.VarChar).Value = Login;

command.Parameters.Add("@password", MySqlDbType.VarChar).Value = Password;

adapter.SelectCommand = command;

adapter.Fill(table);

MySqlDataAdapter users = new MySqlDataAdapter($"Select role From user", db.GetConnection());

role = Convert.ToString(table.Rows[0][5]);

if (table.Rows.Count > 0)

{

ActiveForm.Hide();

general general = new general();

general.Show();

}

else

MessageBox.Show("Пользователя не существует. Проверьте правильность введенных данных");

}

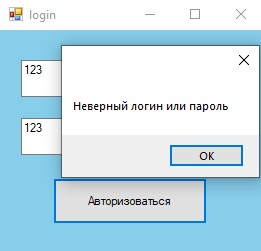
catch (Exception)

{

MessageBox.Show("Ошибка при авторизации. Проверьте правильность введенных данных");

}

Таким образом, если будут введены некорректные данные, либо одно из полей будет не заполнено, программа не завершит работу, а выведет соответствующее сообщение:



Изм.

Лист

№ докум.

Подпись

Дата

Лист

32

КП.09.02.07.ПКИПТ.335 - С.39.21 ПЗ

Рисунок 26 – Обработка исключений

Теперь вводим верные данные, которые содержатся в таблице “user”: логин – 1234, пароль – 1234:

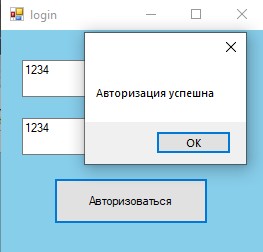
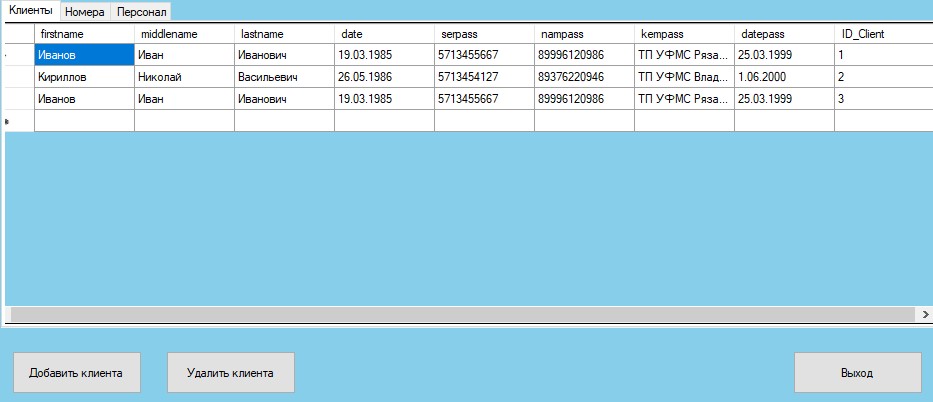


Рисунок 27 – Введены корректные данные

Как и ожидалось, открывается главное окно для просмотра и работы с базой данных.



Изм.

Лист

№ докум.

Подпись

Дата

Лист

33

КП.09.02.07.ПКИПТ.335 - С.39.21 ПЗ

Рисунок 28 – Главное окно приложения

1. Тестируемый модуль – окно добавления данных в таблицу «client». Окно содержит несколько полей, в которые пишутся данные, которые должны попасть в базу данных. При нажатии на кнопку добавить данные будут зафиксированы в базе. Внутри программы в этот момент происходит запрос на языке SQL о добавлении данных в базу.

Исходный код:

MySqlConnection connection = new MySqlConnection(bd);

connection.Open();//соединение с бд

MySqlCommand command = new MySqlCommand($"INSERT INTO `client` (`ID\_Client`, `Firstname`,`Middlename`, `Lastaname`, `date`,`serpass`,`nampass`,`number`,`datepass`,`kempass`,) VALUES (NULL, '{id\_client.Text}', '{fam.Text}','{name.Text}', '{otchectvo.Text}', '{date.Text}', '{number.Text}', '{serpass.Text}', '{nampass.Text}', '{datepass.Text}', '{kempass.Text}');")

{

Connection = connection//команда для заполнения

};

command.ExecuteNonQuery();//выполнение команды

MessageBox.Show("Данные успешно добавлены");

this.Hide();

Для предотвращения аварийного завершения работы приложения в модуль необходимо добавить обработку исключений:

try

{

MySqlConnection connection = new MySqlConnection(connect);

connection.Open();//соединение с бд

MySqlCommand command = new MySqlCommand($"INSERT INTO `client` (`ID\_Client`, `Firstname`,`Middlename`, `Lastaname`, `date`,`serpass`,`nampass`,`number`,`datepass`,`kempass`,) VALUES (NULL, '{id\_client.Text}', '{fam.Text}','{name.Text}', '{otchectvo.Text}', '{date.Text}', '{number.Text}', '{serpass.Text}', '{nampass.Text}', '{datepass.Text}', '{kempass.Text}');")

{

Connection = connection//команда для заполнения

};

command.ExecuteNonQuery();//выполнение команды

MessageBox.Show("Данные успешно добавлены");

this.Hide();

}

catch (Exception)

Изм.

Лист

№ докум.

Подпись

Дата

Лист

34

КП.09.02.07.ПКИПТ.335 - С.39.21 ПЗ

{

MessageBox.Show("Ошибка. Проверьте введенные данные");

}

Таким образом, если одна из строк будет пустая, то приложение не завершится аварийно, а сообщит об ошибке и продолжит свою работу.

Введено недостаточно данных:

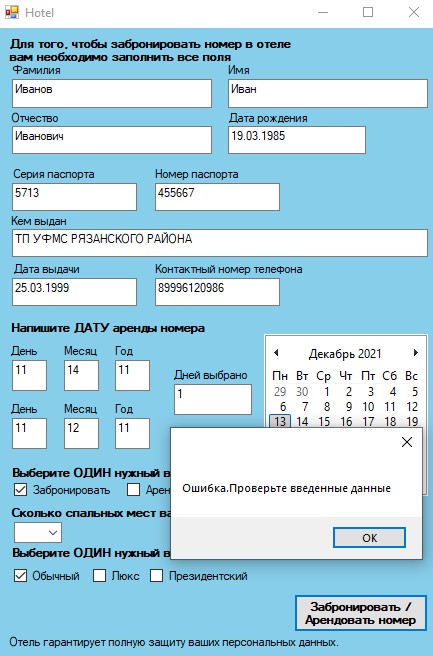


Рисунок 29 – Обработка исключений

Далее введены корректные и полные данные для проверки работоспособности окна и взаимодействия с базой данных:

Изм.

Лист

№ докум.

Подпись

Дата

Лист

35

КП.09.02.07.ПКИПТ.335 - С.39.21 ПЗ

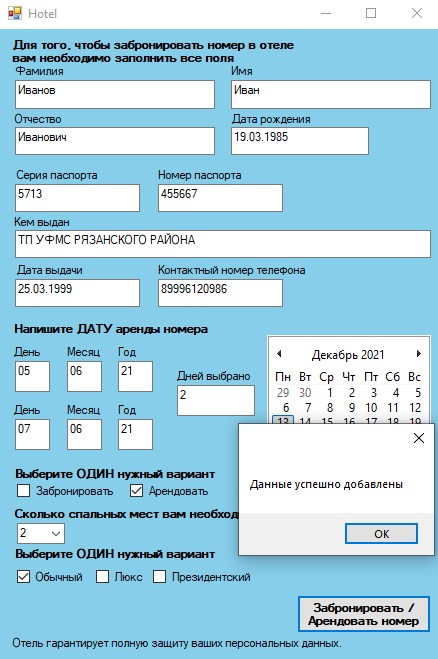


Рисунок 30 – Сообщение о успешном добавлении данных

Данные отображены в базе:

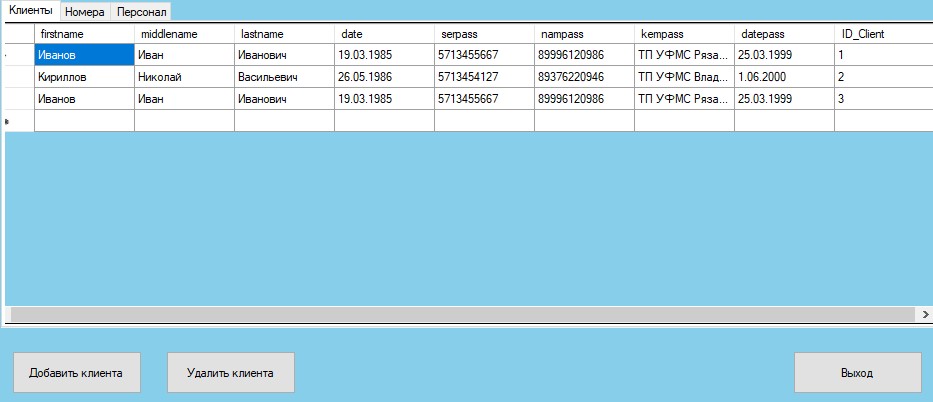


Рисунок 31 – Сообщение о успешном добавлении данных

Таким образом, приложение успешно прошло модульное тестирование. Багов и недочетов в работе проверяемых элементов обнаружено не было.

Далее будет проведено интеграционное тестирование. Данный вид тестирования подразумевает проверку взаимосвязи модулей между собой. Например, правильно ли кнопка с одной формы совершает переход на другую форму и т.д.

Для последовательного и безошибочного тестирования необходимо составить тестовый граф:

Изм.

Лист

№ докум.

Подпись

Дата

Лист

36

КП.09.02.07.ПКИПТ.335 - С.39.21 ПЗ

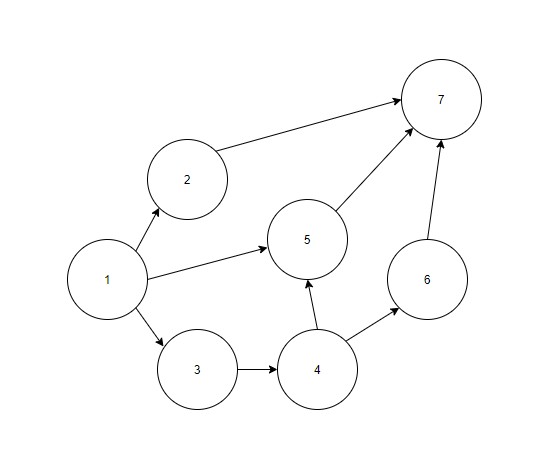


Рисунок 32 – Тестовый граф

Пояснения к тестовому графу:

Запуск главного меню приложения - 1

Попытка авторизации с заведомо неверными данными либо без них - 2

Авторизация с верными данными - 3

Открытие главной формы с базой данных - 4

Добавление данных - 5

Удаление данных - 6

Закрытие приложения - 7

Далее к имеющемуся тестовому графу нужно составить тестовые пути, по которым будет проводиться интеграционное тестирование.

Изм.

Лист

№ докум.

Подпись

Дата

Лист

37

КП.09.02.07.ПКИПТ.335 - С.39.21 ПЗ

Тестовые пути к графу:

Т1: 1 – 2 – 7;

Т2: 1 – 5 – 7;

Т3: 1 – 3 – 4 – 5 – 7;

Т4: 1 – 3 – 4 – 6 – 7.

Данные для интеграционного тестирования можно оформить в виде таблицы:

Таблица 1 - Тест-кейсы для формы авторизации:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | Исходные данные | Цель | Описание | Ожидаемый результат | Результат |
| 1 | Логин – 1234  Пароль - 1234 | Тестирование авторизации и открытия главного окна | Пользователь вводит корректные данные для авторизации, нажимает кнопку «Войти». | Успешная авторизация и открытие главного окна программы. | Успешная авторизация и открытие главного окна программы. |
| 2 | Случайный логин  Случайный пароль | Тестирование безопасности и обработки аварийных ситуаций. | Пользователь вводит неверные данные, пытаясь авторизоваться. | Сообщение об ошибке | Сообщение об ошибке. |

Таблица 1 – Тест кейсы для окна авторизации

Таблица 2 - Тест кейсы для главной формы:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | Исходные данные | Цель | Описание | Ожидаемый результат | Результат |
| 1 | Данные, вытянутые из базы | Тестирование удаления данных | Пользователь выделяет строку в таблице, нажимает кнопку «Удалить» | Выделенная строка удаляется из базы | Выделенная строка удаляется из базы |
| 2 | 8 – размер таблицы;  для заполнения таблицы | Тестирование заполнения таблицы | В сформированную таблицу вводятся числовые данные | Таблица заполняется числовыми данными | Таблица заполнена числовыми данными |

Таблица 2 - Тест кейсы для формы добавления:

Изм.

Лист

№ докум.

Подпись

Дата

Лист

38

КП.09.02.07.ПКИПТ.335 - С.39.21 ПЗ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | Исходные данные | Цель | Описание | Ожидаемый результат | Результат |
| 1 | Иванов Иван Иванович 89994441332 15.04.1968  4778553344  ТП УФМС Рязанского района  20.04.1980 | Тестирование формы добавления нового клиента | Пользователь вводит корректные данные, нажимает кнопку «Добавить». | Данные добавлены успешно. | Данные добавлены успешно. |
| 2 | Иванов Иван Иванович 15.04.1968  4778553344  ТП УФМС Рязанского района  20.04.1980 | Тест на ввод некорректных данных. | Пользователь вводит данные, но не вводит номер телефона. | Сообщение об ошибке | Сообщение об ошибке. |

**2.8 Руководство пользователя**

Изм.

Лист

№ докум.

Подпись

Дата

Лист

39

КП.09.02.07.ПКИПТ.335 - С.39.21 ПЗ

Для работы с оконным приложением необходимо иметь компьютер или ноутбук с операционной системой Windows. Первым делом нужно разархивировать папку с приложением. Затем нужно установить его. В процессе установки разрешить приложению установку на ноутбук, отклоняя его предупреждения о вредоносных файлах.

После установки приложение появится на рабочем столе вашего компьютера. Для работы с ним нужно нажать на иконку приложения. Работа приложения начинается с главной страницы (Рисунок 31). На которой находятся 2 кнопки: забронировать – для добавления новых клиентов и авторизоваться для персонала отеля.

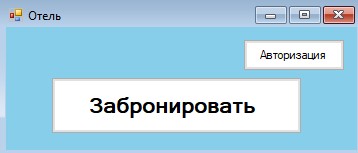
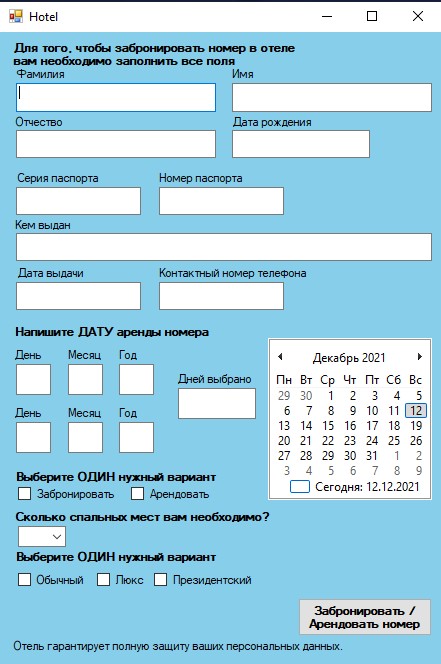


Рисунок 33 – Главное окно

При нажатии на кнопку «Забронировать» откроется окно для добавления новых клиентов в базу данных (Рисунок 32)



Изм.

Лист

№ докум.

Подпись

Дата

Лист

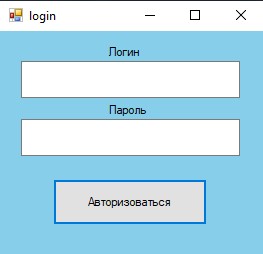
40

КП.09.02.07.ПКИПТ.335 - С.39.21 ПЗ

Рисунок 34 – Окно добавления данных

После того как нажмем кнопку Забронировать/Арендовать номер, данные добавятся в базу данных и в дальнейшем мы можем просмотреть список клиентов, арендовавших или забронировавших номер в отеле.

Для редактирования и просмотра клиентов, можем с главного окна через окно авторизации перейти на форму с базой данных. Для этого с главной формы нужно нажать на кнопку «Авторизоваться», после этого перейдем на вкладку авторизации(Рисунок).



Изм.

Лист

№ докум.

Подпись

Дата

Лист

41

КП.09.02.07.ПКИПТ.335 - С.39.21 ПЗ

Рисунок 35 – Окно авторизации

После того как корректно ввели данные и нажали на кнопку «Авторизоваться» попадаем на форму с базами данных(Рисунок).

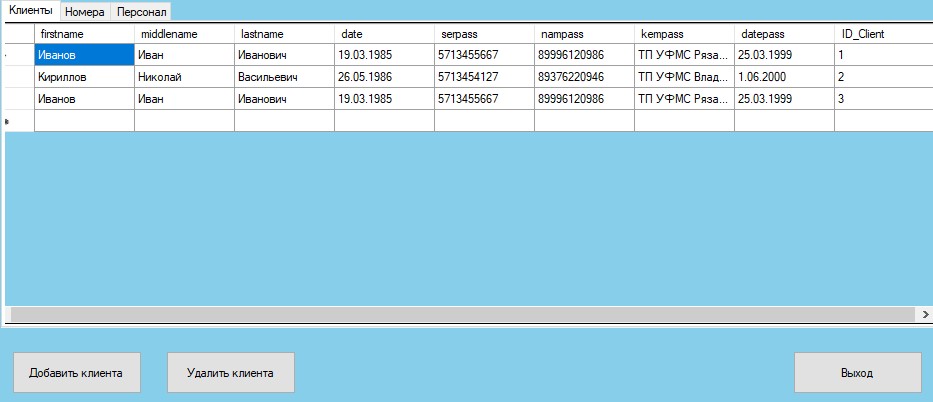


Рисунок 36 – Окно работы с БД

В форме с базами данных мы можем добавить и удалить клиентов при помощи соответствующих кнопок, а также закрыть приложение кнопкой выход.

**Заключение**

Изм.

Лист

№ докум.

Подпись

Дата

Лист

42

КП.09.02.07.ПКИПТ.335 - С.44.21 ПЗ

Результатом выполнения курсовой работы стало разработанное приложение баз данных, позволяющее автоматизировать систему бронирования и аренды номеров отелей. Разработанное приложение отвечает всем требованиям предметной области, таблицы созданной базы данных отвечают требованиям нормализации, что позволяет обеспечить целостность и непротиворечивость информации.

Средствами СУБД Microsoft SQL Server Management Studio 19 создан удобный пользовательский интерфейс для работы с базой данных. Приложение позволяет решать все задачи, сформулированные в задании на курсовую работу. Это позволяет сделать вывод, что задание выполнено полностью.

Программный продукт создан, отлажен и протестирован в среде разработки Microsoft Visual Studio 2019. Проектирование интерфейса так же происходило в данной среде разработки, благо функционал и имеющиеся элементы позволяют это сделать.

Во время работы над курсовым проектом была изучена научно-техническая и справочная литература по данной предметной области, проанализированы функциональные требования и разработана спецификация программы. Выполнена отладка, тестирование программного продукта. Составлена необходимая сопроводительная документация.

Тестирование показало, что разработанная автоматизированная система функционирует правильно и эффектно решает поставленную задачу.

Таким образом, задачи, поставленные в курсовом проекте, выполнены, цель  
 достигнута.

**Список используемой литературы**

Изм.

Лист

№ докум.

Подпись

Дата

Лист

43

КП.09.02.07.ПКИПТ.335 - С.44.21 ПЗ

1. Абрамян, Михаил Visual C# на примерах (+ CD-ROM) / Михаил Абрамян. - М.: БХВ-Петербург, 2017;
2. Агуров, Павел C#. Сборник рецептов (+CD-ROM) / Павел Агуров. - М.: БХВ-Петербург, 2018;
3. Бишоп, Дж. С# в кратком изложении / Дж. Бишоп, Н. Хорспул. - М.: Бином. Лаборатория знаний, 2016;
4. <http://www.cyberforum.ru/> - форум программистов и сисадминов.(25.03.21);
5. <https://ru.stackoverflow.com> – форум программистов и web – разработчиков. (25.03.21);
6. <https://www.sololearn.com> – курсы и уроки по C# и SQL.(26.03.21);
7. <http://metanit.com> - Огромный самоучитель по С# и SQL.(26.03.21);
8. [https://docs.microsoft.com/ru-ru/dotnet/csharp/](https://docs.microsoft.com/ru-ru/dotnet/csharp/%20)  – Документация по С# от создателей языка программирования. (28.03.21);
9. mycsharp.ru — С# для начинающих (28.03.21);
10. <https://dotnet.microsoft.com/learn> — Официальный туториал по языку от Microsoft.(5.04.21);
11. <http://www.tutorialspoint.com/csharp/index.htm> — «На TutorialsPoint в разделе о C# имеется целых два курса по этому языку. (8.04.21)
12. <https://www.learncs.org>— Курс learncs.org больше похож на C# Basic Tutorial от TutorialsPoint, но более лаконичен. (8.04.21);
13. Васильев, Алексей C#. Объектно-ориентированное программирование / Алексей Васильев. - М.: Питер, 2018;
14. Зиборов, В. В. Visual C# 2012 на примерах / В.В. Зиборов. - М.: БХВ-Петербург, 2018;
15. Зиборов, Виктор Visual C# 2010 на примерах / Виктор Зиборов. - М.: "БХВ-Петербург", 2017;
16. Касаткин, А. И. Профессиональное программирование на языке си. Управление ресурсами / А.И. Касаткин. - М.: Высшая школа, 2017;
17. Культин, Н. Microsoft Visual C# в задачах и примерах (+ CD-ROM) / Н. Культин. - М.: БХВ-Петербург, 2016;

Изм.

Лист

№ докум.

Подпись

Дата

Лист

44

КП.09.02.07.ПКИПТ.335 - С.39.21 ПЗ

1. Подбельский, В. В. Язык С#. Базовый курс / В.В. Подбельский. - М.: Финансы и статистика, 2019;
2. Рендольф, Ник Visual Studio 2019 для профессионалов / Ник Рендольф и др. - М.: Диалектика, 2020;
3. Скит, Джон C# для профессионалов. Тонкости программирования / Джон Скит. - М.: Вильямс, 2017;
4. Tроелсен, Эндрю Язык программирования C# 8.0 и платформа .NET 4.5 / Эндрю Троелсен. - М.: Вильямс, 2019;
5. Фленов, Михаил Библия C# / Михаил Фленов. - М.: БХВ-Петербург, 2019;
6. Фримен, Адам LINQ. Язык интегрированных запросов в C# 2019 для профессионалов / Адам Фримен , Джозеф Раттц-мл.. - М.: Вильямс, 2019;
7. Эндрю Троелсен Язык программирования C# 8.0 и платформа .NET 4.5; Диалектика / Вильямс - М., 2019;
8. Фленов Михаил Библия С#; БХВ-Петербург - М., 2019.

**ПРИЛОЖЕНИЕ А**

Изм.

Лист

№ докум.

Подпись

Дата

Лист

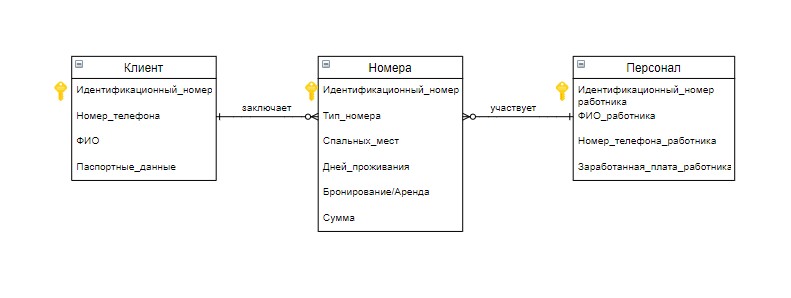
45

КП.09.02.07.ПКИПТ.335 - С.39.21 ПЗ

(обязательное)

Система автоматизации учета реализации и затрат на доставку мебели

ER – диаграмма



Лист

Изм

Система бронирования и аренды номеров отелей

ER - диаграмма

**19ИТ17**

КП.09.02.07.ПКИПТ.335 - С.39.21 ПЗ

**3**

Литера

Утв.

Стенькина Н.А,

Н. контр

Т. контр

Проверила

Лукьянова Г.С

Разработ.

Абрамов О.Е. г

№ докум.

Подп.

Дата

Масса

Лист 1

Листов 1

Масштаб

**ПРИЛОЖЕНИЕ Б**

Изм.

Лист

№ докум.

Подпись

Дата

Лист

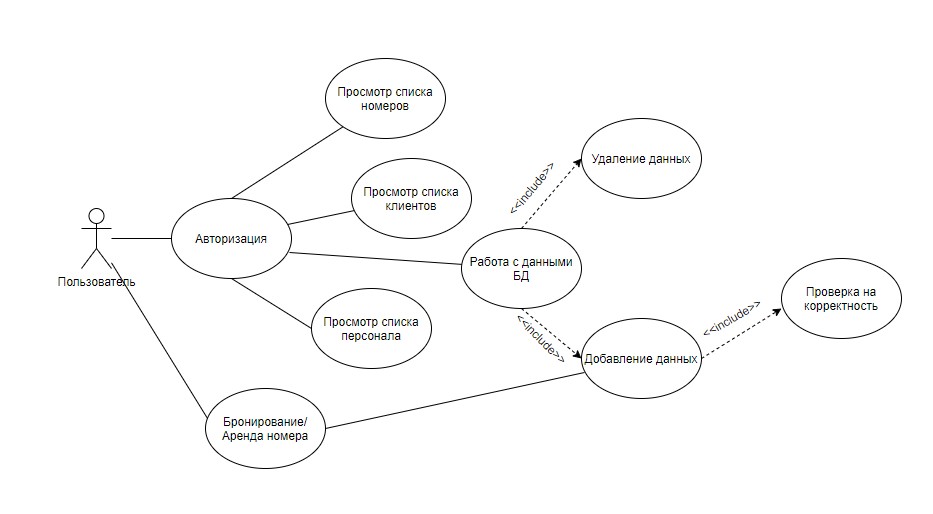
47

КП.09.02.07.ПКИПТ.335 - С.39.21 ПЗ

(обязательное)

Система бронирования и аренды номеров отелей

Диаграмма вариантов использования

****

Лист

Изм

Система бронирования и аренды номеров отелей

Диаграмма вариантов использования

**19ИТ17**

КП.09.02.07.ПКИПТ.335 - С.39.21 ПЗ

**3**

Литера

Утв.

Стенькина Н.А.

Н. контр

Т. контр

Проверила

Лукьянова Г.С

Разработ.

Абрамов О.Е. г

№ докум.

Подп.

Дата

Масса

Лист 1

Листов 1

Масштаб

**ПРИЛОЖЕНИЕ В**

Изм.

Лист

№ докум.

Подпись

Дата

Лист

49

КП.09.02.07.ПКИПТ.335 - С.39.21 ПЗ

(обязательное)

Система бронирования и аренды номеров отелей

Листинг с исходным кодом и результаты работы программы

Результаты работы программы

Изм.

Лист

№ докум.

Подпись

Дата

Лист

50

КП.09.02.07.ПКИПТ.335 - С.39.21 ПЗ

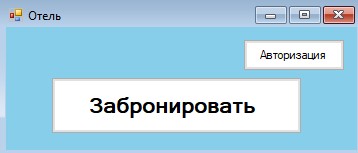


Рисунок 37 – Главное окно

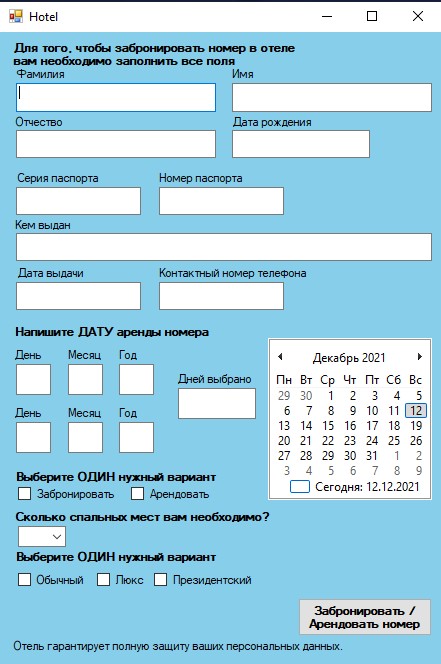
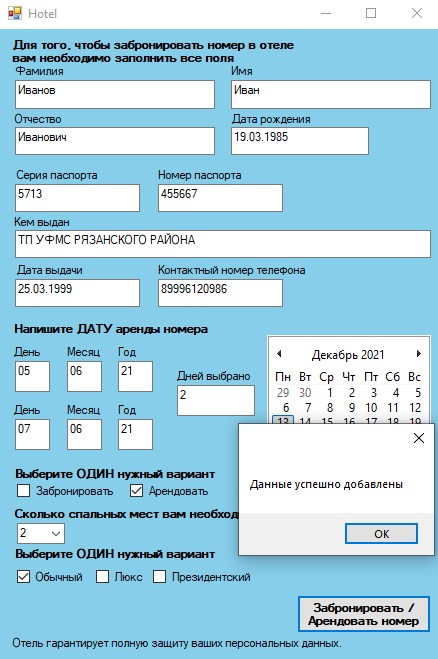


Рисунок 37 – Окно Забронировать



Изм.

Лист

№ докум.

Подпись

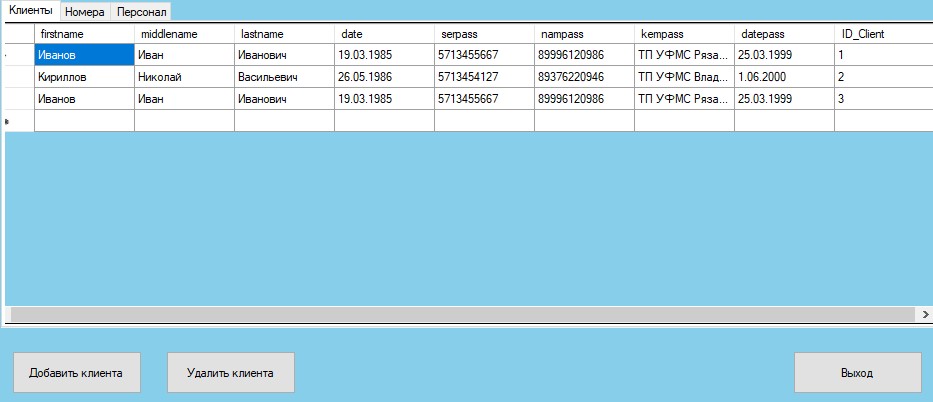
Дата

Лист

51

КП.09.02.07.ПКИПТ.335 - С.39.21 ПЗ

Рисунуок 39 – Пример ввода данных



Рисунуок 39 – Результат ввода данных

**Листинг программы:**

Изм.

Лист

№ докум.

Подпись

Дата

Лист

52

КП.09.02.07.ПКИПТ.335 - С.39.21 ПЗ

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;ч

using System.Threading.Tasks;

using System.Windows.Forms;

using MySql.Data.MySqlClient;

namespace Hotel

{

public partial class guest : Form

{

public guest()

{

InitializeComponent();

{

}

private void textBox16\_TextChanged(object sender, EventArgs e)

{

}

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

{

{

int date1;

int date2;

int sum;

int sumarent;

int sumlux;

int sumprez;

date1 = Convert.ToInt16(textBox10.Text);

date2 = Convert.ToInt16(textBox15.Text);

sum = date2 - date1;

if (sum == 0)

{

textBox16.Text = "1";

casharend.Text = ("Стоимость проживания в номере составляет:" + 1750);

}

if (!checkBox2.Checked == true && !checkBox4.Checked == true)

{

casharend.Text = ("Стоимость проживания в ЛЮКС номере составляет:" + sumlux);

}

if (!checkBox2.Checked == true && !checkBox3.Checked == true)

{

casharend.Text = ("Стоимость проживания в ОБЫЧНОМ номере составляет:" + sumarent);

}

if (!checkBox2.Checked == true && !checkBox1.Checked == true)

{

casharend.Text = ("Стоимость проживания в ОБЫЧНОМ номере составляет:" + sumprez);

}

try

{

MySqlConnection connection = new MySqlConnection(bd);

connection.Open();//соединение с бд

MySqlCommand command = new MySqlCommand($"INSERT INTO `client` (`ID\_Client`, `Firstname`,`Middlename`, `Lastaname`, `date`,`serpass`,`nampass`,`number`,`datepass`,`kempass`,) VALUES (NULL, '{id\_client.Text}', '{fam.Text}','{name.Text}', '{otchectvo.Text}', '{date.Text}', '{number.Text}', '{serpass.Text}', '{nampass.Text}', '{datepass.Text}', '{kempass.Text}');")

{

Connection = connection//команда для заполнения

};

command.ExecuteNonQuery();//выполнение команды

MessageBox.Show("Данные успешно добавлены");

this.Hide();

}

catch (Exception)

{

MessageBox.Show("Ошибка. Проверьте введенные данные");

}

try

{

MySqlConnection connection = new MySqlConnection(bd);

connection.Open();//соединение с бд

MySqlCommand command = new MySqlCommand($"INSERT INTO `number` (`id\_client`, `type\_numbers`, `Sleepbad`, `Daylive`,`Sleepbad`, `Daylive`, `Revelation`, `Cash`) VALUES (NULL, '{id\_client.Text}', '{type\_numbers.Text}', '{sleepbad.Text}', '{daylive.Text}', '{Revelation.Text}', '{cash.Text}');")

{

Connection = connection//команда для заполнения

};

command.ExecuteNonQuery();//выполнение команды

MessageBox.Show("Данные успешно добавлены");

this.Hide();

}

catch (Exception)

{

MessageBox.Show("Ошибка. Проверьте введенные данные");

}

try

{

MySqlConnection connection = new MySqlConnection(bd);

connection.Open();//соединение с бд

MySqlCommand command = new MySqlCommand($"INSERT INTO `personal` (`id\_personal`, `firstpersonal`, `middlepersonal`, `numberpersonal`, `cashpersonal`) VALUES (NULL, '{idpersonal.Text}', '{firstpersonal.Text}', '{middlepersonal.Text}', '{numberpersonal.Text}', '{cashpersonal.Text}');")

{

Connection = connection//команда для заполнения

};

command.ExecuteNonQuery();//выполнение команды

MessageBox.Show("Данные успешно добавлены");

this.Hide();

}

catch (Exception)

{

MessageBox.Show("Ошибка. Проверьте введенные данные");

}

}

}

}

}

}

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Hotel

{

public partial class login : Form

{

public login()

{

InitializeComponent();

}

private void textBox1\_TextChanged(object sender, EventArgs e)

{

}

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

{

string log = textBox1.Text;

string pass = textBox2.Text;

var log1 = "1234";

var pass2 = "1234";

if (log == log1 && pass == pass2)

{

MessageBox.Show("Авторизация успешна");

Form2 f = new Form2();

f.Show();

login c = new login();

c.Close();

}

else

{

MessageBox.Show("Неверный логин или пароль");

}

}

}

}

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Hotel

{

public partial class Form2 : Form

{

public Form2()

{

InitializeComponent();

}

}

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

{

try

{

String connectionString = ConfigurationManager.ConnectionStrings["ConnectionString"].ConnectionString;

SqlConnection con = new SqlConnection(connectionString);

SqlCommand cmd = new SqlCommand("DELETE from [client] where cid ('" + cids.Text + "')", con);

con.Open();

cmd.ExecuteNonQuery();

con.Close();

}

catch (SqlException)

{

Label1.Text = "error";

}

}

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

{

guest f = new guest();

f.Show();

}

}

}

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Hotel

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

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

{

guest f = new guest();

f.Show();

}