



МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
ВЫСШАЯ ШКОЛА ПЕЧАТИ И МЕДИАИНДУСТРИИ

*Институт Принтмедиа и информационных технологий
Кафедра Информатики и информационных технологий*

направление подготовки

09.03.02 «Информационные системы и технологии»,

КУРСОВОЙ ПРОЕКТ

Дисциплина: Базы Данных _____

Тема: Разработка информационной системы проката _____

Выполнил(а): студент(ка) группы 191-724

Зеленков Глеб Александрович _____
(Фамилия И.О.)

Дата, подпись 28.6.21 _____
(Дата) (Подпись)

Проверил: _____
(Фамилия И.О., степень, звание)

Дата, подпись _____
(Дата) (Подпись)

Замечания: _____

Москва

2021

Оглавление

Введение	3
Часть 1. Проектирование	4
1.1 Описание предметной области	4
1.2 Выбор инструментов.....	4
Глава 2. Разработка информационной системы.....	6
2.1 Проектирование и физическая реализация базы данных на СУБД	6
2.2 Разработка программного продукта	10
Заключение.....	18

Введение

Тема проекта – информационная система предназначенная для проката автомобилей.

Часто перед клиентами стаёт потребность в аренде авто для разных целей, кому то для переезда, кто то хочет уехать куда то на праздник, у кого то машина в ремонте и нужна временная замена. Для более точного подбора под нужды клиента возникла потребность в создании удобного приложения для проката авто, с выбором функций интересующих арендаторов. Так же приложения избавит от необходимости куда либо ехать и выбирать авто на месте, а позволит например , подобрать и арендовать авто из дома или в дороге.

Цель – создание ИС для подбора и бронирования прокатных автомобилей.

Сформированы задачи:

1. Изучение предметной области
2. Выбор инструментов разработки
3. Проектирование БД
4. Реализация физической модели

Часть 1. Проектирование

1.1 Описание предметной области

Автомобиль человек подбирает исходя из своих потребностей и обстоятельств при которых он её арендует. Сейчас машины людям нужны достаточно часто из-за чего на прокатные автомобили есть высокий спрос. Для потенциальных арендаторов было создано приложение позволяющее им подобрать машину исходя из их запросов и сроков аренды.

Приложение предназначено для работы с 2мя типами пользователей : клиент и администратор.

Для клиентов приложение помогает подбирать необходимые товар в программе и бронировать его без непосредственной поездки в салон и общения с людьми. Это сократит время просмотра поиска и ожидания , которое будет затрачено на бронирование.

Технология бронирования со стороны клиента: Клиент выбирает автомобиль и срок аренды, вводит свои данные и подтверждает заказ (заказ отправляется в БД)

После этих действий клиент ожидает звонок от администратора с информацией о заказе.

Для администратора: изменение ассортимента в приложении, что позволит быстро и без ошибок обеспечивать работу сервиса.

В связи с этим заказ автоматизирован и оптимизирован для клиентов и администраторов.

1.2 Выбор инструментов

Для реализации выбрана десктопная форма приложения для достижения более стабильной работы и ориентированности на крупные компании которые могут заказы на большие мероприятия и иметь постоянный и быстрый доступ к каталогу.

Упомянутые ранее причины подбора автомобиля требуют внимательного изучения товара. Так же клиента интересует всё же получить машину а не расстаться с деньгами. Из за этого в приложение будут входить легальные и проверенные прокатные сервисы.

Подразумевается создание приложения на Windows из за низкой распространённости MacOS и Linux. Для этого была выбрана среда разработки Visual Studio и язык программирования C#. В связи с тем что у перечисленной выше ОС ,среды разработки и языка одна студия разработчик

И высокая совместимость между друг другом ожидается очень стабильная и надёжная работа.

Microsoft Visual Studio — продукт компании Microsoft, в которую входят интегрированная среда разработки программного обеспечения и

множество других инструментов. Данный продукт даёт возможность разрабатывать приложения разных типов - консольные приложения, веб-сайты, вебприложения, веб-службы, для всех платформ, поддерживаемых Windows, Windows Mobile, Windows CE, .NET Framework, Xbox, Windows Phone .NET Compact Framework и Silverlight. А также приложения с графическим интерфейсом, в том числе, с поддержкой технологии Window

C# — современный объектно-ориентированный язык программирования. C# относится к известному семейству языков C, и покажется хорошо знакомым любому, кто работал с C, C++, Java или JavaScript. C# является объектно-ориентированным языком, но поддерживает также и компонентно-ориентированное программирование. Разработка современных приложений все больше тяготеет к созданию программных компонентов в форме автономных и самоописательных пакетов, реализующих отдельные функциональные возможности. Важная особенность таких компонентов — это модель программирования на основе свойств, методов и событий. Каждый компонент имеет атрибуты, предоставляющие декларативные сведения о компоненте, а также встроенные элементы документации. C# предоставляет языковые конструкции, непосредственно поддерживающие такую концепцию работы. Благодаря этому C# отлично подходит для создания и применения программных компонентов. Для создания баз данных в роли СУБД в данном проекте выступает MySQL Workbench. MySQL Workbench — инструмент для визуального проектирования баз данных, интегрирующий проектирование, моделирование, создание и эксплуатацию БД в единое бесшовное окружение для системы баз данных MySQL.

Для создания баз данных в роли СУБД в данном проекте выступает MySQL Workbench. MySQL Workbench — инструмент для визуального проектирования баз данных, интегрирующий проектирование, моделирование, создание и эксплуатацию БД в единое бесшовное окружение для системы баз данных MySQL. В качестве среды разработки баз данных, созданных на MySQL Workbench, выбрана программа MySQL Workbench 8.0 CE.

Данное приложение бесплатно распространяется ORACLE и представляет собой удобный интерфейс позволяющий взаимодействовать с БД как при помощи кода, так и без его участия. В возможности это приложения входят такие функции как: Создание, администрирование, редактирование, удаление таблиц и самих БД. Данное приложение по популярности конкурирует с MsSQL и PostgreSQL.

Глава 2. Разработка информационной системы.

2.1 Проектирование и физическая реализация базы данных на СУБД

Для реализации данного проекта была создана база данных, состоящая из трёх таблиц. Таблица реляционной базы данных — это совокупность связанных данных, хранящихся в структурированном виде в базе данных.

Далее представлены структуры всех таблиц с пояснением к каждой.

Таблица `ud` — пользователь — таблица, в которой хранятся все данные о пользователях.

Данная таблица содержит:

- `IdUD` — присваивается автоматически,
- `Fam` — Фамилия ;
- `Im` — имя;
- `Otch` — отчество;
- `Pasport` — серия и номер паспорта;
- `drlic` — номер водительского удостоверения;
- `phnum` — номер телефона;
- `mail` — электорнный адрес;

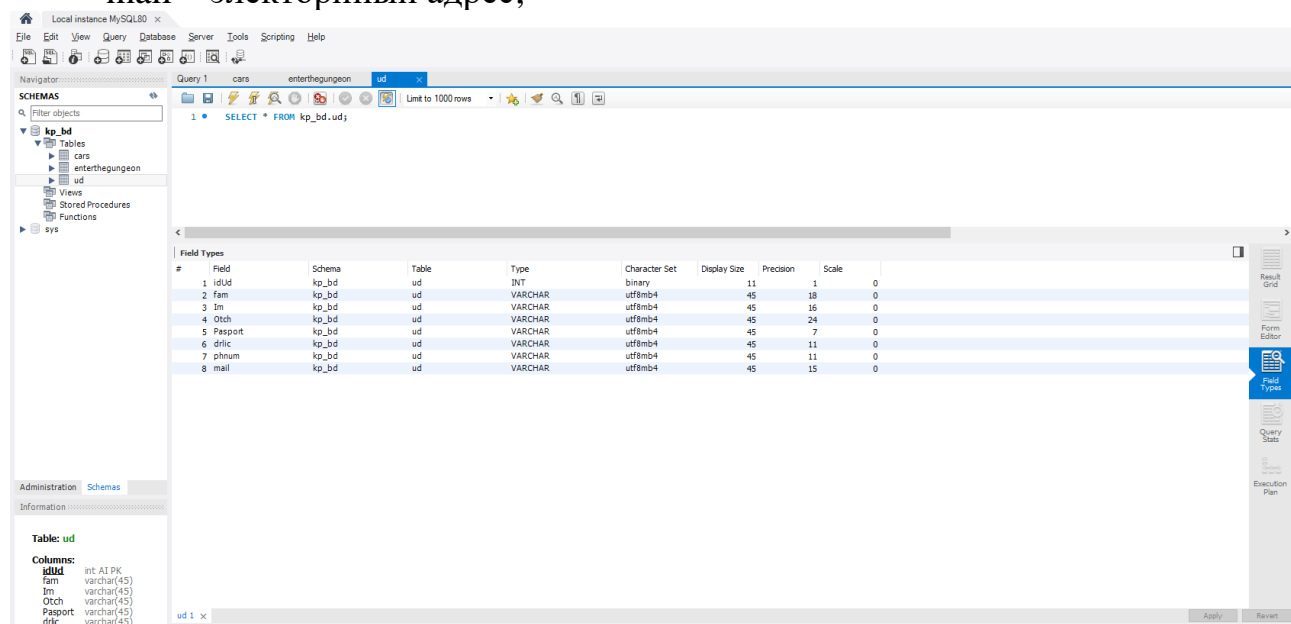


Рисунок 2.1 Таблица пользователей

Таблица `cars` — автомобили — таблица, содержащая всю необходимую информацию об ассортименте и характеристиках автомобилей.

Данная таблица включает в себя поля:

- `id` — также, как и в таблице пользователей определяется автоматически;

- Name – название товара;
- year – год выпуска;
- mileage – пробег ;
- bodytype – тип кузова автомобиля;
- price – цена аренды;
- licence – категория прав необходимая для управления автомобилем;

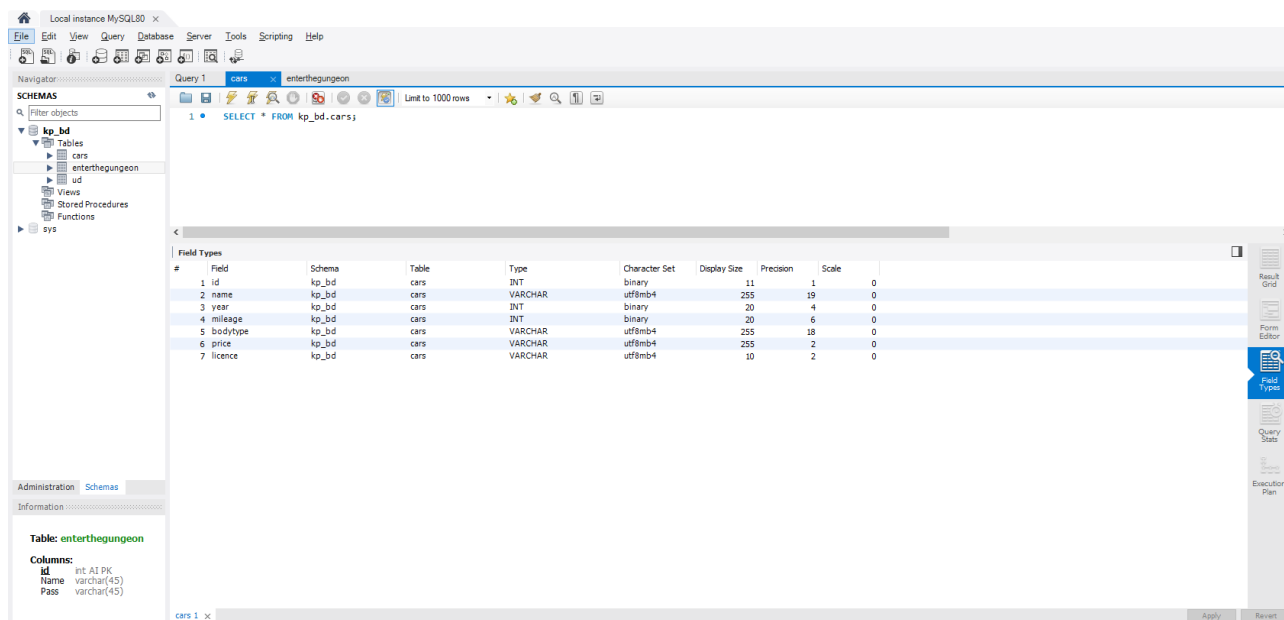


Рисунок 2.2 Таблица автомобилей

Таблица enterthegungeon – таблица, в которой содержится информация о администраторах приложения которые могут как расширять так и сужать ассортимент.

В эту таблицу входят:

id – также, как и в предыдущих таблицах определяется автоматически;

Name – имя учётной записи администратора;

Pass – пароль учётной записи.

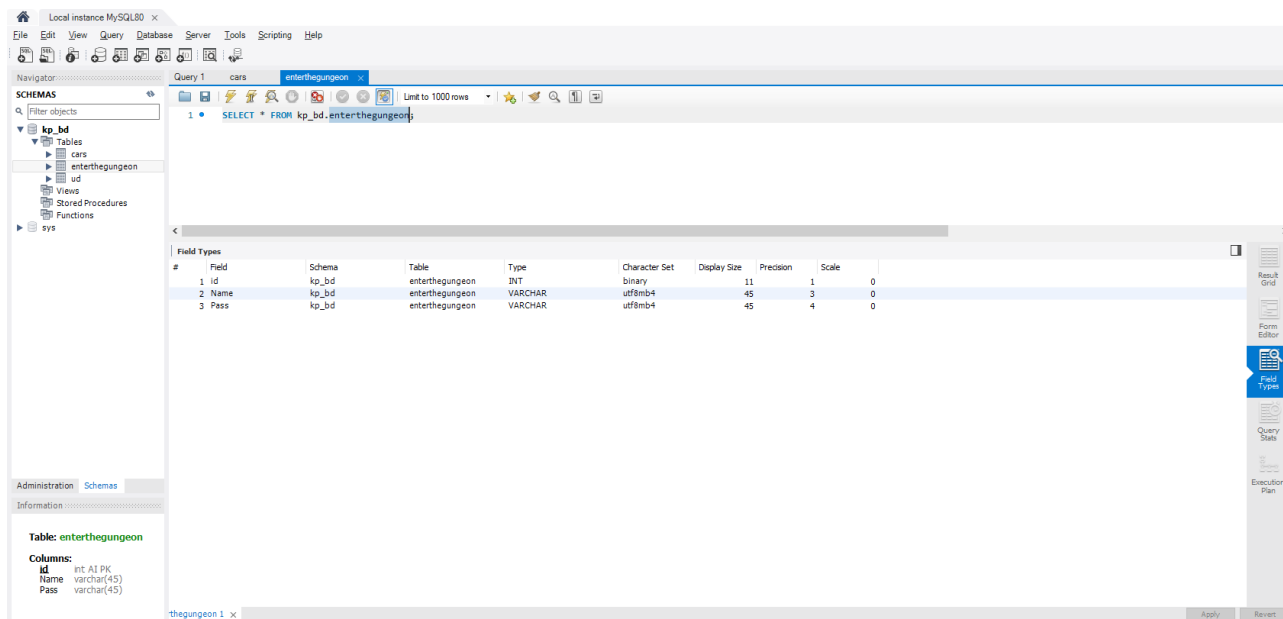


Рисунок 2.3 Таблица Администраторов

Примеры заполнения таблиц данными

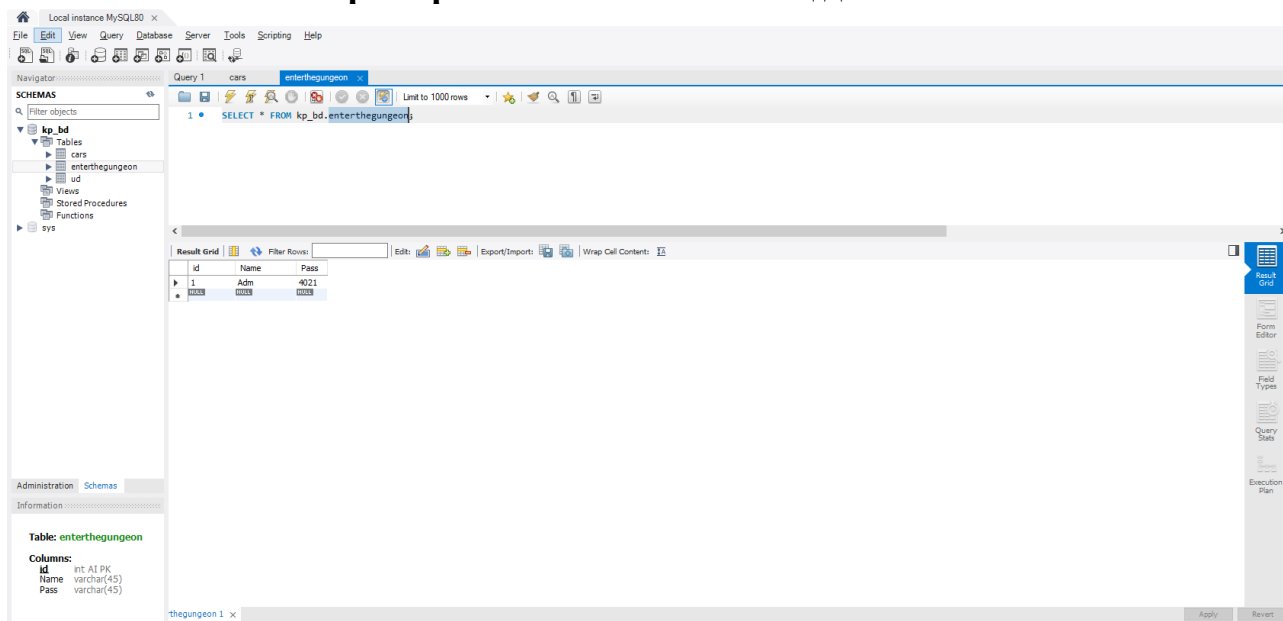


Рисунок 2.4 Таблица Администраторов пример заполнения

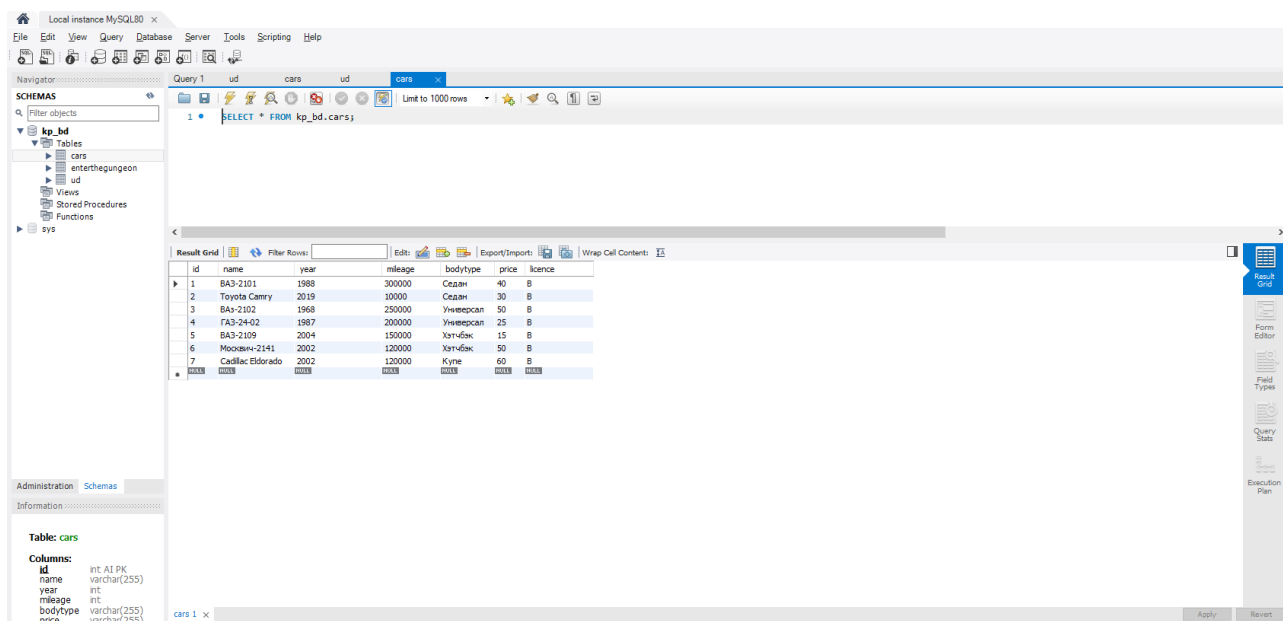


Рисунок 2.5 Таблица автомобилей пример заполнения

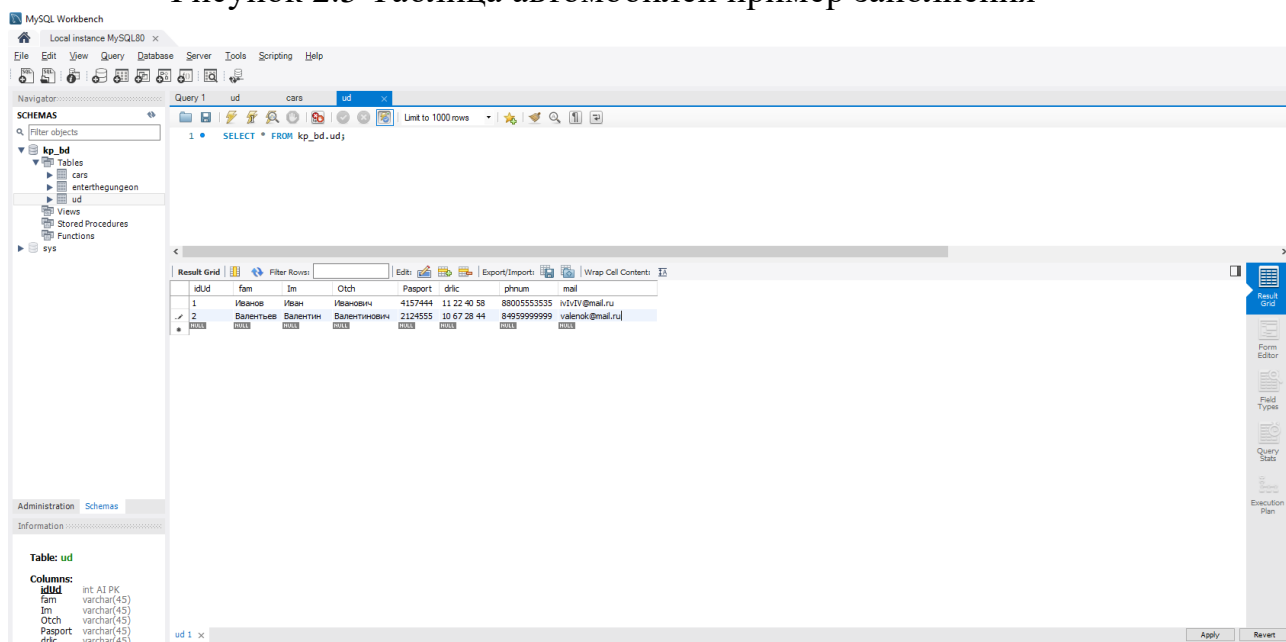
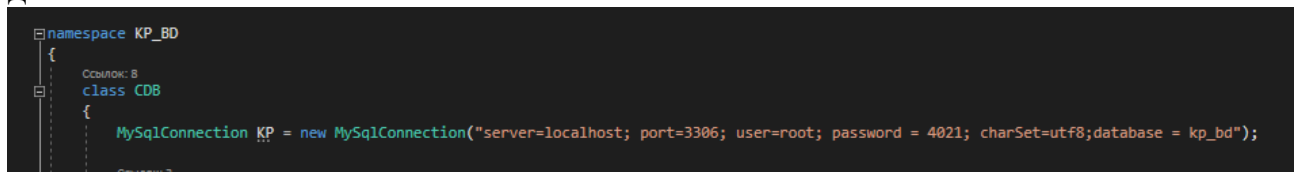


Рисунок 2.6 Таблица пользователей пример заполнения

2.2 Разработка программного продукта

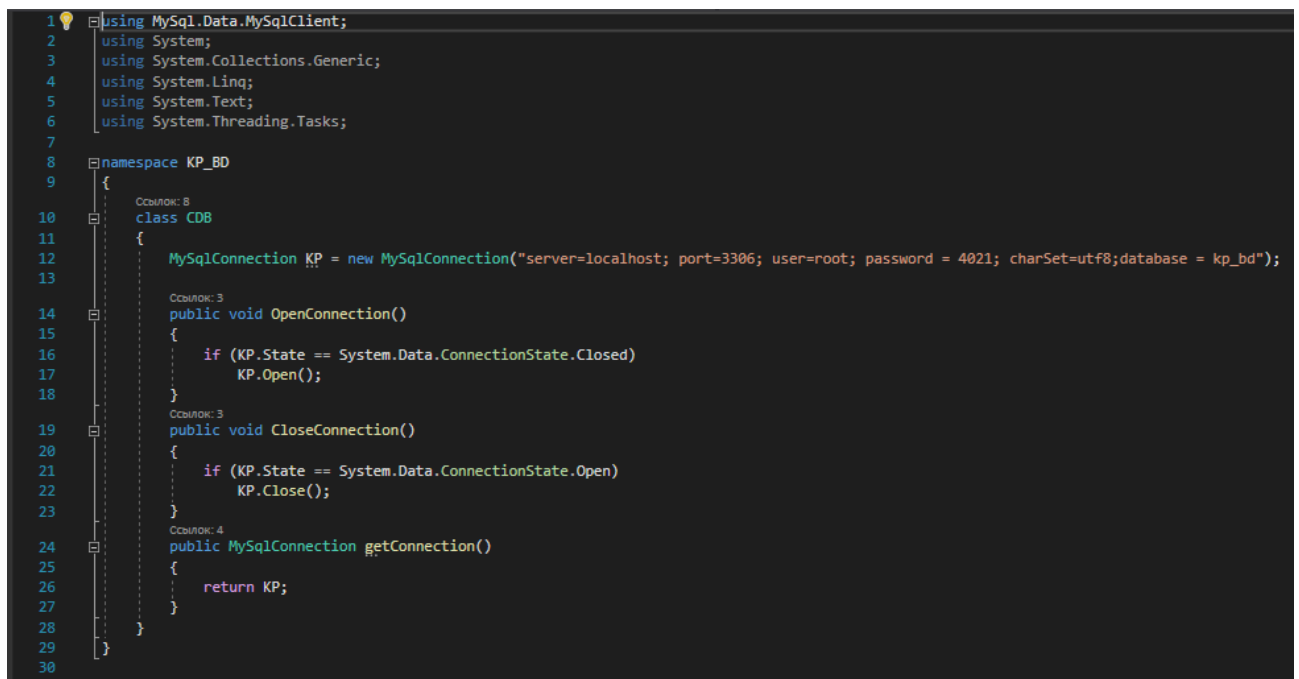
Чтобы понимать, как данный проект был реализован, необходимо рассмотреть основные части программного кода (язык программирования – C#). Далее представлены главные функции приложения в программном виде. Поскольку данный проект создавался в среде Visual Studio с использованием технологии Windows Forms для реализации графического интерфейса приложения, то для наполнения приложения необходимым контентом, необходимо было подключить созданную ранее базу данных, содержащую весь необходимый материал. Ниже представлена программная реализация подключения базы данных.



```
namespace KP_BD
{
    class CDB
    {
        MySqlConnection KP = new MySqlConnection("server=localhost; port=3306; user=root; password = 4021; charSet=utf8;database = kp_bd");
    }
}
```

Рисунок 2.7 Подключение к базе данных

Так же был разработан отдельный класс для быстрого подключения к БД – класс CDB.



```
1 using MySql.Data.MySqlClient;
2 using System;
3 using System.Collections.Generic;
4 using System.Linq;
5 using System.Text;
6 using System.Threading.Tasks;
7
8 namespace KP_BD
9 {
10     class CDB
11     {
12         MySqlConnection KP = new MySqlConnection("server=localhost; port=3306; user=root; password = 4021; charSet=utf8;database = kp_bd");
13
14         public void OpenConnection()
15         {
16             if (KP.State == System.Data.ConnectionState.Closed)
17                 KP.Open();
18         }
19
20         public void CloseConnection()
21         {
22             if (KP.State == System.Data.ConnectionState.Open)
23                 KP.Close();
24         }
25
26         public MySqlConnection getConnection()
27         {
28             return KP;
29         }
30     }
31 }
```

Рисунок 2.8 Листинг класса

```

41
CDB bd = new CDB();
MySQLCommand command = new MySQLCommand ("INSERT INTO kp_bd.cars (name, year, mileage, bodytype, price,licence) VALUES (@name,@ye,@mil,@body,@price,@licence)" , bd.getConnection());
command.Parameters.Add("@name", MySqlDbType.VarChar).Value=markcar.Text;
command.Parameters.Add("@ye", MySqlDbType.Int32).Value = year.Text;
command.Parameters.Add("@mil", MySqlDbType.Int32).Value = mileage.Text;
command.Parameters.Add("@body", MySqlDbType.VarChar).Value = body.Text;
command.Parameters.Add("@price", MySqlDbType.VarChar).Value = price.Text;
command.Parameters.Add("@licence", MySqlDbType.VarChar).Value = drivecard.Text;
bd.OpenConnection();
if (command.ExecuteNonQuery() == 1)
    MessageBox.Show("Добавлена машина");
else
    MessageBox.Show("Добавление не удалось");
bd.CloseConnection();

```

Рисунок 2.9 Пример использования класса

Был написан скрипт для Администратора приложения позволяющий добавить новый автомобиль. Так как в базу данных планируется не только добавлять но и удалять из неё машины отдельно был написан скрипт удаления данных из таблицы.

```

CDB bd = new CDB();
MySQLCommand command = new MySQLCommand ("INSERT INTO kp_bd.cars (name, year, mileage, bodytype, price,licence) VALUES (@name,@ye,@mil,@body,@price,@licence)" , bd.getConnection());
command.Parameters.Add("@name", MySqlDbType.VarChar).Value=markcar.Text;
command.Parameters.Add("@ye", MySqlDbType.Int32).Value = year.Text;
command.Parameters.Add("@mil", MySqlDbType.Int32).Value = mileage.Text;
command.Parameters.Add("@body", MySqlDbType.VarChar).Value = body.Text;
command.Parameters.Add("@price", MySqlDbType.VarChar).Value = price.Text;
command.Parameters.Add("@licence", MySqlDbType.VarChar).Value = drivecard.Text;
bd.OpenConnection();
if (command.ExecuteNonQuery() == 1)
    MessageBox.Show("Добавлена машина");
else
    MessageBox.Show("Добавление не удалось");
bd.CloseConnection();

```

Рисунок 2.10 Листинг скрипта добавления данных

```

private void button1_Click(object sender, EventArgs e)
{
    CDB del = new CDB();
    MySQLCommand com = new MySQLCommand ("DELETE FROM kp_bd.cars WHERE id = @delb", del.getConnection());
    com.Parameters.Add("@delb", MySqlDbType.VarChar).Value = DELB.Text;
    del.OpenConnection();
    if (com.ExecuteNonQuery() == 1)
        MessageBox.Show("Машина Удалена");
    else
        MessageBox.Show("Удаление не удалось");
    del.CloseConnection();
}

```

Рисунок 2.11 Листинг скрипта удаления данных

Исходя из вопросов безопасности данных и работоспособности приложения написан скрипт для входа администратора в приложение. Без входа в приложение под учётной записью администратора выполнение вышеописанных взаимодействий добавления и/или удаления невозможно. Принцип работы: когда администратор вводит свои данные и нажимает кнопку входа, страница подключается к базе данных. Далее считываются id, пароль и логин. Затем эти данные сравниваются с данными в базе данных. Если они совпали, то открывается окно редактирования таблицы, в противном случае на экран будет выведено уведомление о неверном пароле, вход в приложение не будет произведён и пользователь может произвести попытку входа ещё раз

```

String LoG = LogIn.Text;
String PaS = Password.Text;

CDB dB = new CDB();
DataTable enterthegungeon = new DataTable();
MySqlDataAdapter adapter = new MySqlDataAdapter();
MySqlCommand command = new MySqlCommand("SELECT * FROM kp_bd.enterthegungeon WHERE Name= @username AND Pass= @password", dB.getConnection());
command.Parameters.Add("@username", MySqlDbType.VarChar).Value = LoG;
command.Parameters.Add("@password", MySqlDbType.VarChar).Value = PaS;
adapter.SelectCommand = command;
adapter.Fill(enterthegungeon);
if (enterthegungeon.Rows.Count > 0)
{
    this.Hide();
    Form3 voiti = new Form3();
    voiti.Show();
}
else
    MessageBox.Show("Неверный пароль");

```

Рисунок 2.12 Листинг скрипта входа администратора

Так как при выборе пользователем автомобиля просматривать в ручную огромный список в поисках необходимой машины тяжело добавлен скрипт поиска по базе данных.

```

private void button1_Click(object sender, EventArgs e)
{
    if (rtb.Text != "")
    {
        string script = "Select * from kp_bd.cars WHERE CONCAT(" + rtb.Text + ") like '%" + FoundBox.Text + "%'";
        con = new MySqlConnection(connect);
        con.Open();
        MySqlDataAdapter ms_data = new MySqlDataAdapter(script, connect);
        SD.DataTable table = new SD.DataTable();
        ms_data.Fill(table);
        dataGridView1.DataSource = table;
        con.Close();
    }
    else
    {
        MessageBox.Show("ERROR");
    }
}

```

Рисунок 2.13 Листинг скрипта строки поиска

Для пользователей всех уровней необходим простой и понятный интерфейс с которым приятно и легко взаимодействовать. С точки зрения маркетинга, правильно оформленное приложение лучше воспринимается клиентом и

располагает его к приобретению товара. Первое, что видит пользователь, запуская приложение – это страница входа. На данной странице пользователь может зарегистрироваться, или сразу перейти к подбору машины. С этой же страницы осуществляется вход в аккаунт администратора.

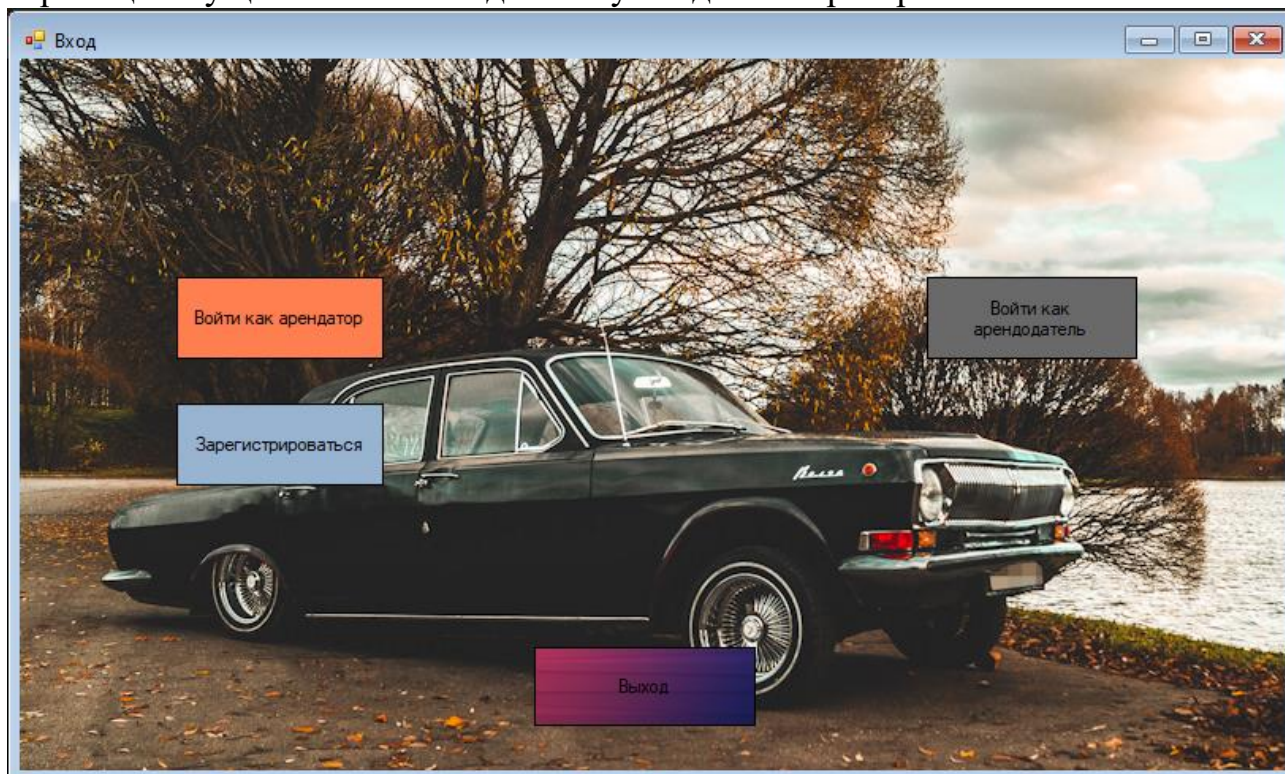


Рисунок 2.14 Окно входа

Если пользователь решает зарегистрироваться то для этого в приветственном окне он нажимает кнопку «зарегистрироваться» и приложение переносит его на страницу регистрации. Потенциальный покупатель должен заполнить необходимые строки соответствующими личными данными. Затем эти данные по нажатию кнопки зарегистрироваться на соответствующей странице будут занесены системой в базу данных проекта.

Form4

Фамилия	<input type="text"/>	Паспортные данные	<input type="text"/>
Имя	<input type="text"/>	Номер прав	<input type="text"/>
Отчество	<input type="text"/>	Номер телефона	<input type="text"/>
		Эл.Почта	<input type="text"/>

Зарегистрироваться

Рисунок 2.15 Окно регистрации

Если пользователь хочет сразу посмотреть ассортимент он нажимает на кнопку «Войти как арендатор» и оказывается на экране поиска автомобиля. Данный экран имеет строку поиска что позволяет быстро сортировать автомобили по различным критериям. Пользователь нажимает на кнопку «Нажмите что бы начать подбор» потом в строку ввода вводит необходимые параметр для поиска и нажимает кнопку «Поиск» после чего в нижней области окна отобразиться необходимый результат. С данной страницы существует переход на предыдущее окно если пользователь выбрал себе машину и хочет её арендовать.



Рисунок 2.16 Окно подбора авто

Вторая ветка приложения предназначена для администратора и начинается с необходимости входа в приложение, по нажатию соответствующей кнопки администратор перейдёт к данному окну после чего ему необходимо ввести данные учётной записи и подтвердить вход

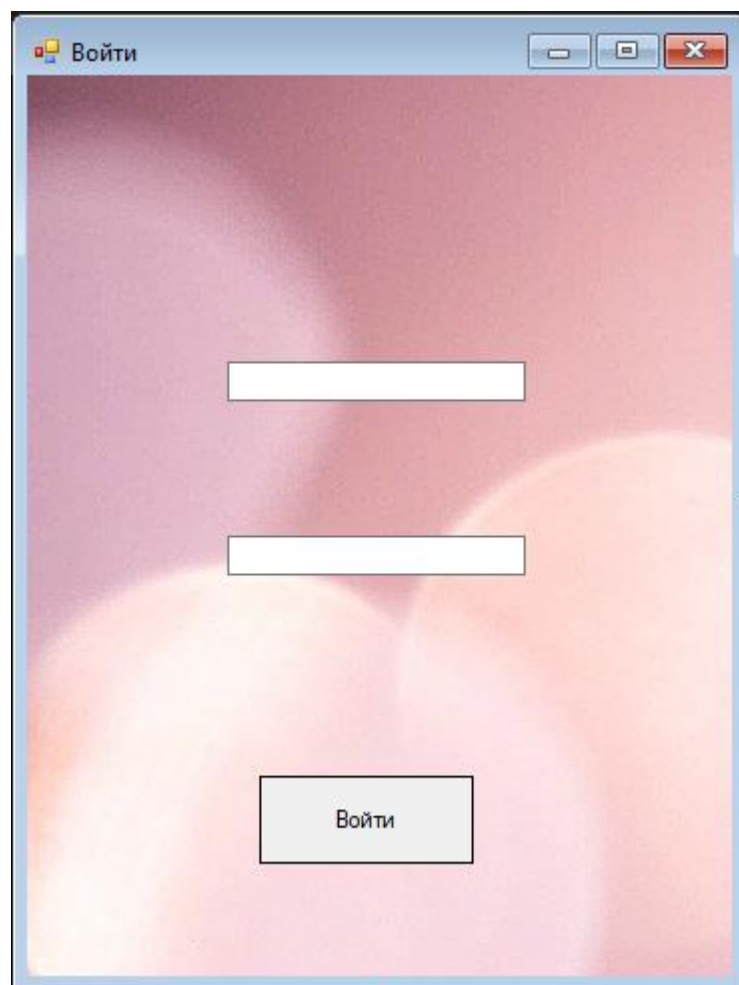


Рисунок 2.16 Окно авторизации администратора

После прохождения авторизации администратору открывается возможность редактировать ассортимент базы данных. Если в автопарке появляется новая машина её можно достаточно быстро внести в список. А если её разобьёт или она например безвозвратно сломается то её можно будет так же быстро удалить. При добавлении машины происходит обращение к БД и данные из ячеек формы переносятся в таблицу. Далее просмотр изменений в таблице можно увидеть на другом листе окна нажав кнопку «Показать список». Если машины в автопарке по каким то причинам больше нет введя её номер в таблице её можно удалить

Редактор списка

tabPage1 tabPage2

Марка

Кузов

Год

Цена

Пробег

Категория прав

Добавить машину

Рисунок 2.17 Страница добавления автомобиля

Редактор списка

tabPage1 tabPage2

Показать список

Введите номер строки и нажмите удалить

Удалить

Рисунок 2.18 Список автомобилей и кнопка удаления

Заключение

По итогу, цель данного проекта была достигнута.

Информационная система была реализована с использованием языка запросов SQL и при помощи системы управления базами данных MySQL Workbench. За счет чего были получены навыки работы с данными инструментами.

В ходе проекта были получены навыки:

- проектирования баз данных,
- создания десктопного приложения,
- подключения созданной базы данных к приложению.

В результате осуществления данного проекта была создана информационная система для сервиса проката автомобилей, которая позволила упростить процесс заказа автомобилей в аренду.

Библиографический список

1. Попов Д.И., Попова Е.Д. Информационные технологии. Базы данных [Текст]: учеб. пособие для студ. вузов /Д.И. Попов. – М: гос. ун-т, печати, 2009. – 117 с.
2. Карпова И.П. Базы данных [Текст]: учеб. пособие для студ. вузов /И.П. Беляев. – М: Московский государственный институт электроники и математики, 2009. – 130 с.
3. Руководство по PHP [Электронный ресурс] . - <http://php.net/manual/ru/> - статья в интернете.
4. Oracle MySQL Cloud Service [Электронный ресурс]. - <https://www.oracle.com/ru/mysql/> - статья в интернете.
5. Энциклопедия [Электронный ресурс]. - <https://ru.wikipedia.org>
6. Описание продукта Microsoft Visual Studio [Электронный ресурс]. - <https://itpro.ua/product/visual-studio-enterprise-MSDN/?tab=description> - статья в интернете.
7. Подключение к базе данных [Электронный ресурс]. - <https://o7planning.org/ru/10511/connecting-to-sql-server-database-using-csharp> - статья в интернете
8. Дополнительные материалы курса Базы Данных (4 семестр) <https://lms.mospolytech.ru/course/view.php?id=2900§ion=10>