Минобрнауки России

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «ОРЕНБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Факультет математики и информационных технологий

Кафедра геометрии и компьютерных наук

**ОТЧЕТ**

Проект базы данных в соответствии с индивидуальным заданием

по дисциплине «Технологии баз данных»

ОГУ 02.03.02. 7021. 337. П

Руководитель

Доцент кафедры ГКН

–––––––––Э.Ф.Морковина

«\_\_»\_\_\_\_\_\_\_\_\_\_\_\_2021 г.

Студент группы

20ФИИТ(б)ОП

\_\_\_\_\_\_\_\_\_\_\_ А. Е.Поляков

«\_\_»\_\_\_\_\_\_\_\_\_\_\_\_2021 г.

Оренбург 2021

Оглавление

[Введение: 3](#_Toc93407237)

[1 Анализ предметной области 4](#_Toc93407238)

[1.1 Изучении истории Hyperloop 4](#_Toc93407239)

[2 MySQL 5](#_Toc93407240)

[2.1 Изучение возможностей MySQL Workbench; 5](#_Toc93407241)

[3 Подключаем БД к приложению 9](#_Toc93407242)

[3.1 Скачиваем приложения для успешного подключения к БД 9](#_Toc93407243)

[3.2 Подключение БД 9](#_Toc93407244)

[4 Написание рабочего приложения. 10](#_Toc93407245)

[4.1 Создание авторизации с помощью БД 10](#_Toc93407246)

[4.2 Создадим регистрацию 12](#_Toc93407247)

[4.3 Само приложение 14](#_Toc93407248)

[5 Тестировка приложения 15](#_Toc93407249)

[5.1 Вход пользователя 15](#_Toc93407250)

[5.2 Регистрация пользователя 15](#_Toc93407251)

[6 Вывод 18](#_Toc93407252)

[7 Список литературы 19](#_Toc93407253)

[8 Приложение 20](#_Toc93407254)

# Введение

Существует множество различных способов добраться из точки А в точку В, например, такие как самолет, машина, корабли или даже ракеты. Но среди них можно выделить пятый вид транспорта. Это гордое название носит Hyperloop. Hyperloop — это предлагаемая высокоскоростная транспортная система для пассажирских и грузовых перевозок. Этот термин был придуман Илоном Маском для описания современного проекта с открытым исходным кодом, изначально задуманного в 1900-х годах. Hyperloop описывается как большая вакуумная герметичная труба или система соединенных вакуумных герметичных трубок с очень низким давлением воздуха, через которые капсула может двигаться практически без сопротивления воздуха или трения. На данный момент идея далека от своего воплощения, но уже существуют прототипы. Так, например подземное шоссе в Лос-Анджелесе используют похожую технологию. На данный момент нет реализации «ПО» для Hyperloop, но я решил исправить эту проблему. В моем проекте реализовано некое подобие такси: мы имеем возможность добраться из одной точки в другую за очень короткий срок. Также приложение подстроено под современные реалии нашего мира. Пассажиры, как и водители, не выбирают транспортное средство. Это позволяет значительно ускорить и разнообразить работу такси.

Цель проекта - изучениеоснов разработки информационных систем, а также реализации приложения Hyperloop. Для достижения поставленной цели выделены следующие задачи:

- проанализировать предметную область;

- разработать логическую модель данных;

- изучить возможности MySQL Workbench;

- реализовать базу данных средствами MySQL;

- написать рабочее приложение, используя полученный ранее опыт;

- протестировать приложение.

# 1 Анализ предметной области

## 1.1 Изучении истории Hyperloop

Впервые концепция вакуумного поезда была предложена американским ученым Робертом Годдардом в 1904 году.

Первый проект движения поездов в вакууме в России был предложен в 1911 году российским ученым Борисом Вейнбергом. По его замыслу, внутри трубы, из которой откачан воздух, должна была перемещаться капсула. Она приводилась в движение с помощью «электромагнитной пушки» и теоретически могла развивать скорость 800—1000 км/ч. Учёный провёл опыты в Томском технологическом институте по перемещению капсулы в трубе, но воплощению идеи помешала Первая мировая война.

В целом история движения поездов в вакууме описана во множестве источников задолго до 2013 года, что отражено в публикации «Магнитолет профессора Вейнберга».

Идея Hyperloop возникла в ответ на правительственный проект высокоскоростной железной дороги California High-Speed Rail, которая к 2029 году должна соединить Лос-Анджелес и Сан-Франциско. «Высокоскоростная» линия предполагает движение поездов со скоростью свыше 322 км (200 миль) в час, стоимость проекта в опубликованном в апреле 2012 года бизнес-плане была оценена в $68,4 млрд.По этому поводу Илон Маск заявил, что калифорнийская дорога станет самой медленной среди скоростных железнодорожных магистралей мира и при этом самой дорогой в пересчёте на милю.

Впервые Маск упомянул о Hyperloop в интервью с Сарой Лэйси, которое дал в июле 2012 года на мероприятии PandoDaily в Санта-Монике. Предприниматель пообещал, что новое транспортное средство будет в 2 раза быстрее самолёта и в 3—4 раза быстрее скоростного поезда, время в пути от Лос-Анджелеса до Сан-Франциско (расстояние 561 км по прямой) займёт всего 30 минут. Позже Маск сделал ещё ряд громких заявлений: проект окажется в 10 раз дешевле по сравнению с California High-Speed Rail; транспорт не будет подвержен авариям; дорога должна работать от солнечной энергии; пассажирам не придётся подстраиваться к расписанию, ибо транспортные капсулы будут двигаться с короткими интервалами, как в метро; все спецификации окажутся в свободном доступе. Наконец, в августе 2013 года была опубликована 58-страничная «альфа-версия» проекта.

# 2 MySQL

# 2.1 Изучение возможностей MySQL Workbench

Создадим базу под названием Taxi: (Рис. 1)

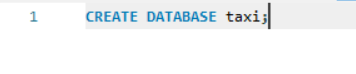


Рисунок 1- Создание базы данных

Для того, чтобы переключиться на нашу базу данных, можно выполнить команду: (Рис. 2)

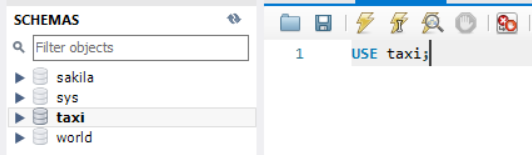


Рисунок 2 – Подключение к БД

Для того, чтобы создать таблицу в SQL, используется выражение CREATE TABLE. (Рис. 3)

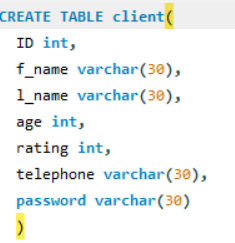


Рисунок 3 – Создание таблицы

Заполним таблицу с помощью следующей команды: INSERT INTO `taxi`.`client` (`f\_name`, `l\_name`, `telephone`, `password`) VALUES ('Дамир', 'Григорьевич', '89139250249', '1234'); (Рис. 4)

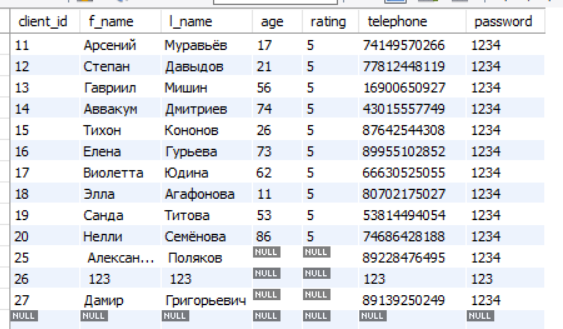


Рисунок 4 – Таблица client

2.2 Создание и заполнение таблиц.

Таким же образом создадим и заполним следующие таблицы:

-Car (Рис. 5)

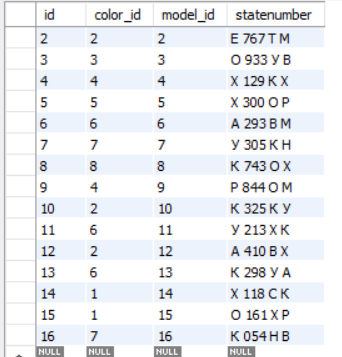


Рисунок 6 – Таблица car

-model(Рис. 6)



Рисунок 6 – Таблица model

-Color(Рис. 7)



Рисунок 7 – Таблица color

-driver(Рис. 8)

 Рисунок 6 – Таблица driver

# 3 Подключаем БД к приложению

## 3.1 Скачиваем приложения для успешного подключения к БД

<https://dev.mysql.com/downloads/connector/net/>

## 3.2 Подключение БД

Добавим класс DB. Подключимся к БД(рис. 9)

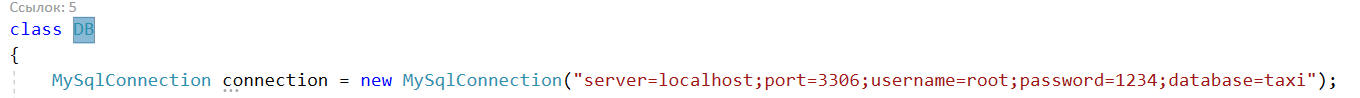


Рисунок 9 – Подключимся к БД

Создадим функции для удобной работы в Visual Studio:

Открытие и закрытие соединений с БД (рис. 10)

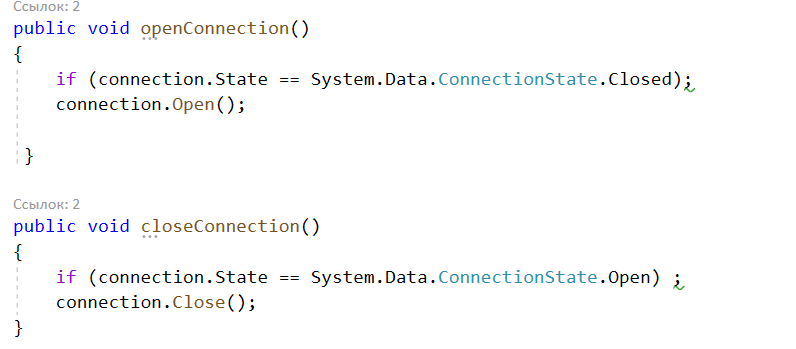


Рисунок 10 – Открытие и закрытие соединений с БД

Также создадим функцию для возвращения соединения с БД: (рис. 11)

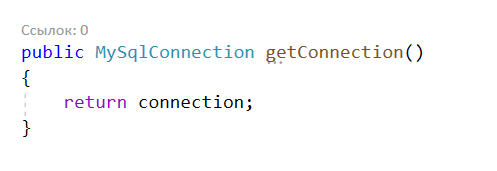


Рисунок 11 – возвращения соединения с БД

# 4 Написание рабочего приложения

## 4.1 Создание авторизации с помощью БД

В конструкторе Windows Form создадим дизайн нашего приложения(рис. 12)

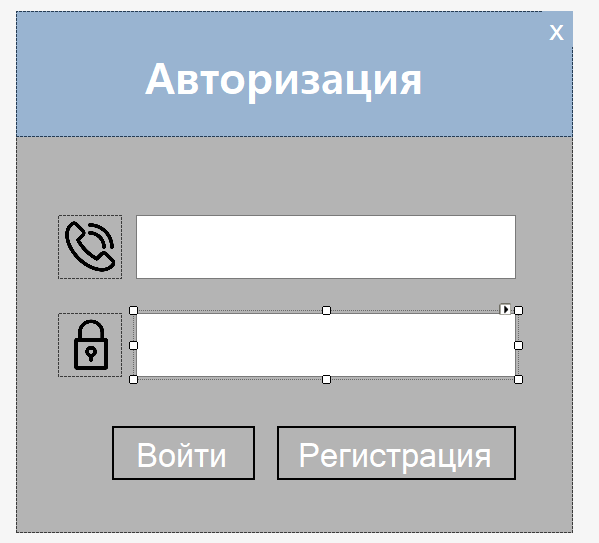


Рисунок 12 – дизайн приложения

Подробно разберем все элементы:

Дизайн разделен на 2 части: фоновая надпись авторизация и рабочая поверхность.

Подробнее рассмотрим 2-ю часть:

В середине мы видим два TextBox-a для ввода логина(в роли которого выступает номер телефона) и пароля.

Ниже мы видим 2 кнопки (войти и регистрация)

Пропишем функционал кнопки войти:

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

{

string loginUser = loginField.Text;

string passUser = passField.Text;

DB db = new DB();

DataTable table = new DataTable();

MySqlDataAdapter adapter = new MySqlDataAdapter();

MySqlCommand command = new MySqlCommand("SELECT \* FROM taxi.client WHERE telephone = @uL AND password = @uP", db.getConnection());

command.Parameters.Add("@uL", MySqlDbType.VarChar).Value = loginUser;

command.Parameters.Add("@uP", MySqlDbType.VarChar).Value = passUser;

adapter.SelectCommand = command;

adapter.Fill(table);

if(table.Rows.Count > 0)

{

this.Hide();

MainForm mainForm = new MainForm();

mainForm.Show();

}

else

{

MessageBox.Show("No");

}

}

Подробнее разберем элементы кода:

Создадим 2 стринговые переменные для ввода логина и пароля пользователем:

string loginUser = loginField.Text;

string passUser = passField.Text;

После получения данных мы можем сверить их с данными из нашей БД.

Создадим некий объект который будет идти от класса DB и выделяем под него память:

DB db = new DB();

Создадим MySQL команду с помощью языка с#:

MySqlDataAdapter adapter = new MySqlDataAdapter();

MySqlCommand command = new MySqlCommand("SELECT \* FROM taxi.client WHERE telephone = @uL AND password = @uP", db.getConnection());

Создадим заглушки для более безопасного пользования БД:

command.Parameters.Add("@uL", MySqlDbType.VarChar).Value = loginUser;

command.Parameters.Add("@uP", MySqlDbType.VarChar).Value = passUser;

С помощью SelectCommand мы укажем какую комманду мы будем выполнять:

adapter.SelectCommand = command;

Параметр adapter позволяет выбрать данные из БД.

Соответственно из тех данных, которые он выдал, мы создадим что-то в духе массива данных:

adapter.Fill(table);

Мы указываем, что заполняем такой объект как table с помощью тех данных, которые мы получили.

Обращаясь у объекта table к количеству рядов, мы считаем количество рядов. Если их больше чем 0, то мы с 100% вероятностью можем утверждать, что пользователь у нас есть. Если пользователя нет, то выведем NO:

if(table.Rows.Count > 0)

{

this.Hide();

MainForm mainForm = new MainForm();

mainForm.Show();

}

else

{

MessageBox.Show("No");

}

С помощью данного кода при успешном выполнении if мы перейдем в другую WindowsForm

this.Hide();

MainForm mainForm = new MainForm();

mainForm.Show();

## 4.2 Создадим регистрацию

Р-рим вторую кнопку на рис. 12. Она нам позволяет совершить регистрацию пользователя.

При нажатии на эту кнопку мы перейдем в другую Windows Form где возможна регистрация (т.е. добавление его данных в БД) пользователя.

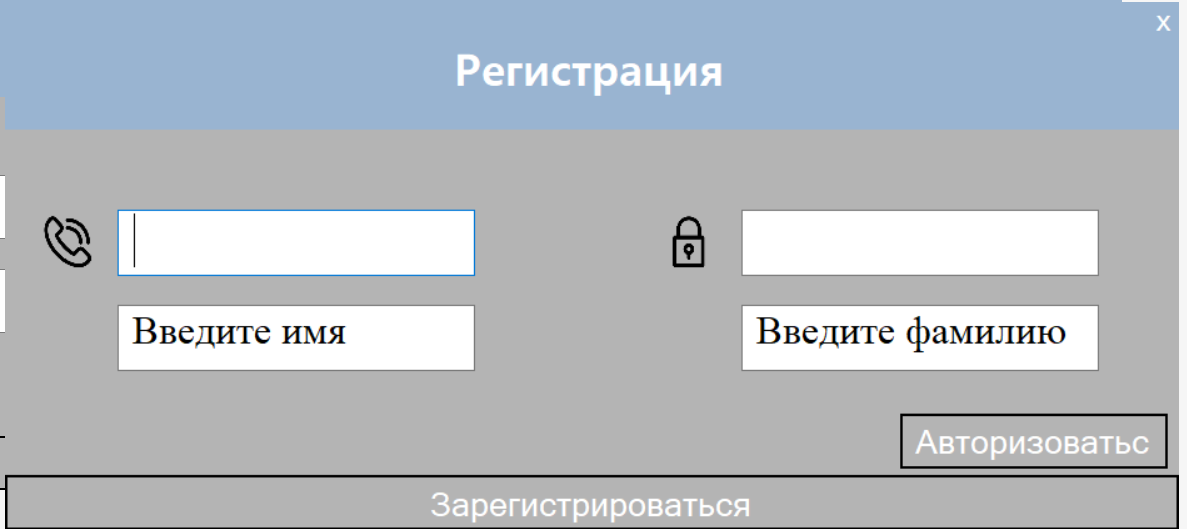


Рисунок 13 – окно регистрации

Действия в этом классе аналогичные предыдущему классу за исключение добавления в БД. Рассмотрим этот кусок кода:

MySqlCommand command = new MySqlCommand("INSERT INTO taxi.client (f\_name, l\_name, telephone, password) VALUES (@name, @surname, @login, @pass)", db.getConnection());

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

command.Parameters.Add("@name", MySqlDbType.VarChar).Value = userNameField.Text;

command.Parameters.Add("@surname", MySqlDbType.VarChar).Value = userSurnameField.Text;

command.Parameters.Add("@pass", MySqlDbType.VarChar).Value = passField.Text;

Команда MySQL INSERT INTO добавляет в нашу бд данные о пользователе. Тут также реализована система заглушек.

В коде также реализованы некоторые защитные меры, такие как проверка на введение всех TextBox-ов:

if (userNameField.Text == "Введите имя")

{

MessageBox.Show("Введите имя");

return;

}

if (userSurnameField.Text == "Введите фамилию")

{

MessageBox.Show("Введите фамилию");

return;

}

if (loginField.Text == " ")

{

MessageBox.Show("Введите телефон");

return;

}

if (passField.Text == " ")

{

MessageBox.Show("Введите пароль");

return;

}

Или на существование этого пользователя:

if (checkUser())

return;

public Boolean checkUser()

{

DB db = new DB();

DataTable table = new DataTable();

MySqlDataAdapter adapter = new MySqlDataAdapter();

MySqlCommand command = new MySqlCommand("SELECT \* FROM taxi.client WHERE telephone = @uL", db.getConnection());

command.Parameters.Add("@uL", MySqlDbType.VarChar).Value = loginField.Text;

adapter.SelectCommand = command;

adapter.Fill(table);

if (table.Rows.Count > 0)

{

MessageBox.Show("Данный телефон зарегестрирован");

return true;

}

else

{

return false;

}

}

## 4.3 Само приложение

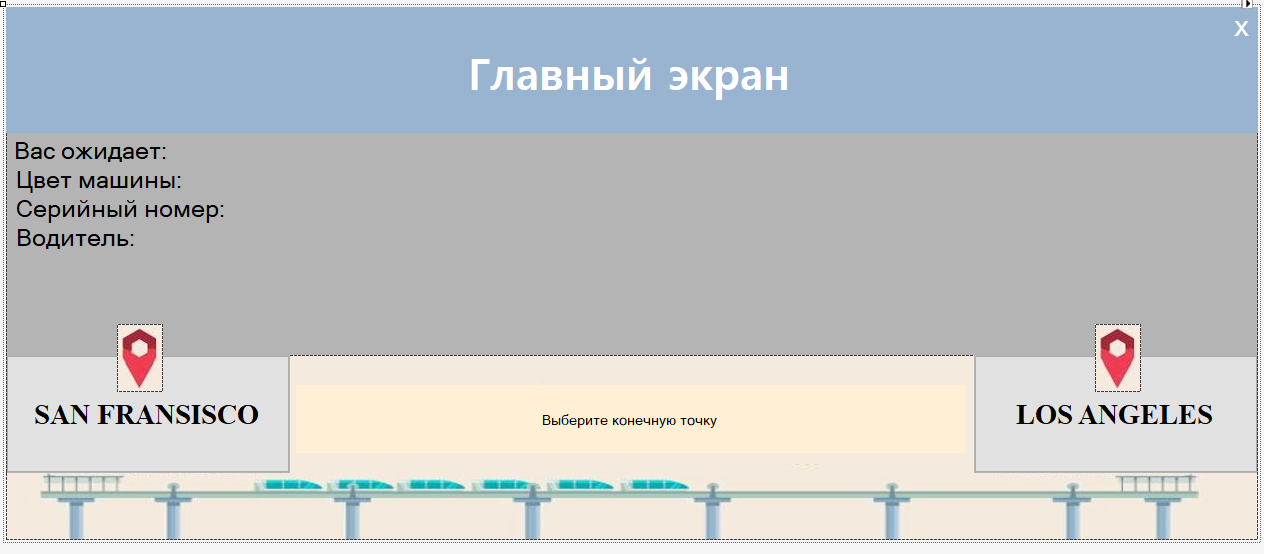


Рисунок 14 – Главный экран

При выборе отправной точки вам автоматически определится водитель и машина, информацию о которых вы получите на главном экране приложения.

# 5 Тестировка приложения

## 5.1 Вход пользователя

При правильном вводе логина пароля вас перекинет в главное меню (рис. 15)

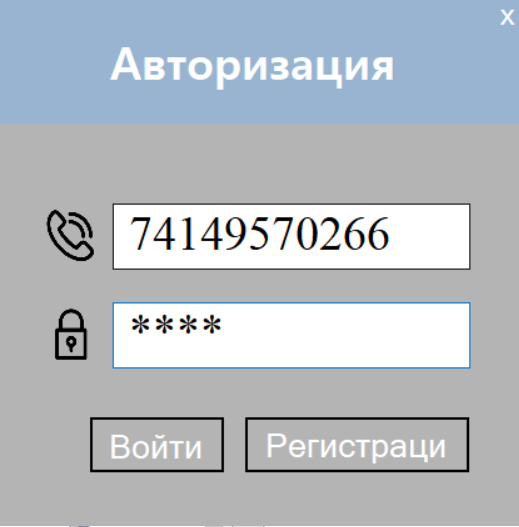


Рисунок 15 – Переход в окно аторизации

## 5.2 Регистрация пользователя

При успешной регистрации пользователя (рис. 16) мы можем увидеть наши данные в приложении MySQL (рис. 17) (ID 25)

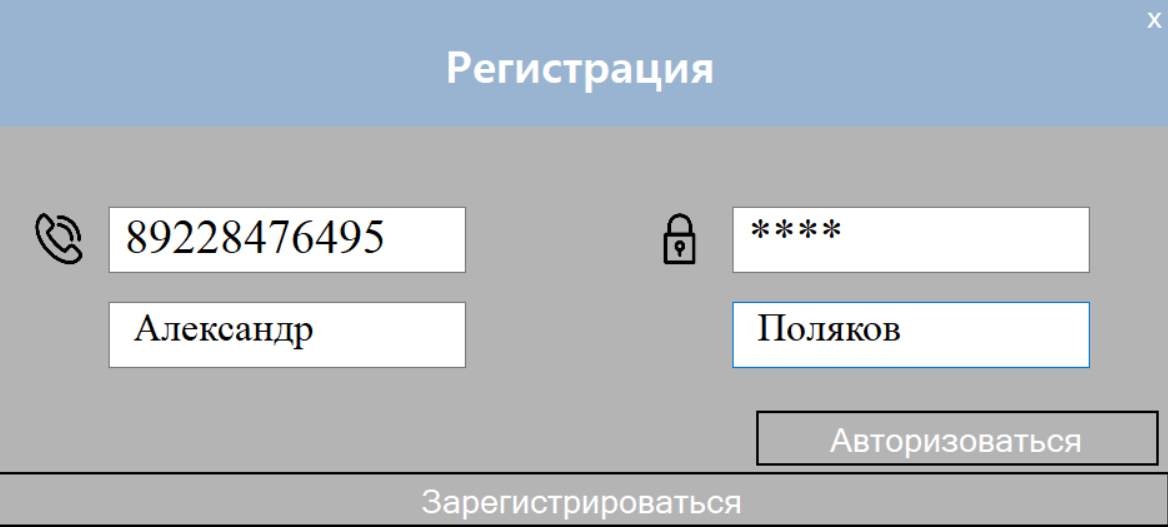


Рисунок 16 – окно регистрации с заполненными данными



Рисунок 17 – Отображение новой информации в БД

Также при выборе отправного пункта в главном меню (рис. 18) успешно будет выведена вся информация о машине и водителе, серийном номере и цвете машины.

Есть возможность отправить из двух пунктов: LS & SF:

DB db = new DB();

DataTable table = new DataTable();

MySqlDataAdapter adapter = new MySqlDataAdapter();

MySqlCommand command = new MySqlCommand("SELECT \* FROM taxi.crew JOIN taxi.car ON taxi.car.id = taxi.crew.car\_id join taxi.driver on taxi.crew.driver\_id = taxi.driver.driver\_id join taxi.color on taxi.car.color\_id = taxi.color.id join taxi.model on taxi.model.id = taxi.car.model\_id WHERE 1 ORDER BY RAND() LIMIT 1", db.getConnection());

adapter.SelectCommand = command;

adapter.Fill(table);

db.openConnection();

textBox5.Text = table.Rows[0].ItemArray[10].ToString();

textBox6.Text = table.Rows[0].ItemArray[9].ToString();

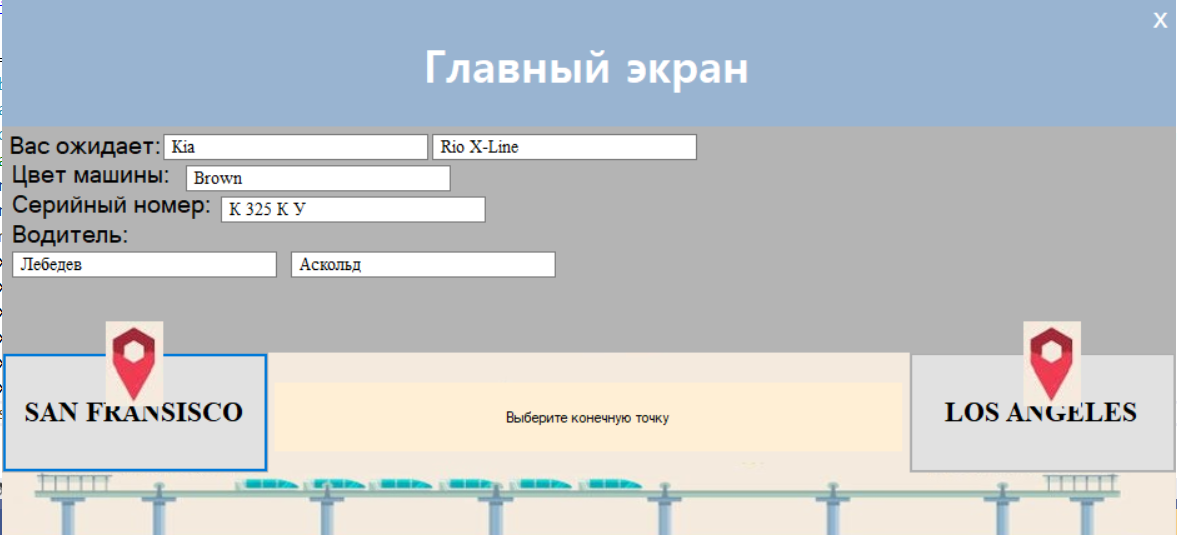
textBox3.Text = table.Rows[0].ItemArray[17].ToString();

textBox4.Text = table.Rows[0].ItemArray[7].ToString();

TextBox1.Text = table.Rows[0].ItemArray[20].ToString();

textBox2.Text = table.Rows[0].ItemArray[19].ToString();

db.closeConnection();



# Рисунок 18 – Главный экран приложения

# 6 Вывод

Пройдя обучение, я успешно научился пользоваться БД и реализовал свои знания в проекте.

# 7 Список литературы

1. Youtube [Электронный ресурс] <https://www.youtube.com/watch?v=eLQAgHexThM&t=187s>

# 8 Приложение

namespace taxi

{

class DB

{

MySqlConnection connection = new MySqlConnection("server=localhost;port=3306;username=root;password=1234;database=taxi");

public void openConnection()

{

if (connection.State == System.Data.ConnectionState.Closed);

connection.Open();

}

public void closeConnection()

{

if (connection.State == System.Data.ConnectionState.Open) ;

connection.Close();

}

public MySqlConnection getConnection()

{

return connection;

}

}

}

namespace taxi

{

public partial class MainForm : Form

{

public MainForm()

{

InitializeComponent();

}

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

{

Application.Exit();

}

Point LastPoint;

private void panel1\_MouseMove(object sender, MouseEventArgs e)

{

if (e.Button == MouseButtons.Left)

{

this.Left += e.X - LastPoint.X;

this.Top += e.Y - LastPoint.Y;

}

}

private void panel1\_MouseDown(object sender, MouseEventArgs e)

{

LastPoint = new Point(e.X, e.Y);

}

private void Exit\_MouseEnter(object sender, EventArgs e)

{

Exit.ForeColor = Color.Black;

}

private void Exit\_MouseLeave(object sender, EventArgs e)

{

Exit.ForeColor = Color.Transparent;

}

private void panel1\_Paint(object sender, PaintEventArgs e)

{

}

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

{

DB db = new DB();

DataTable table = new DataTable();

MySqlDataAdapter adapter = new MySqlDataAdapter();

MySqlCommand command = new MySqlCommand("SELECT \* FROM taxi.crew JOIN taxi.car ON taxi.car.id = taxi.crew.car\_id join taxi.driver on taxi.crew.driver\_id = taxi.driver.driver\_id join taxi.color on taxi.car.color\_id = taxi.color.id join taxi.model on taxi.model.id = taxi.car.model\_id WHERE 1 ORDER BY RAND() LIMIT 1", db.getConnection());

//command.Parameters.Add("taxi.driver.f\_name", MySqlDbType.VarChar).Value = TextBox1.Text;

adapter.SelectCommand = command;

adapter.Fill(table);

db.openConnection();

textBox5.Text = table.Rows[0].ItemArray[10].ToString();

textBox6.Text = table.Rows[0].ItemArray[9].ToString();

textBox3.Text = table.Rows[0].ItemArray[17].ToString();

textBox4.Text = table.Rows[0].ItemArray[7].ToString();

TextBox1.Text = table.Rows[0].ItemArray[20].ToString();

textBox2.Text = table.Rows[0].ItemArray[19].ToString();

db.closeConnection();

}

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

{

DB db = new DB();

DataTable table = new DataTable();

MySqlDataAdapter adapter = new MySqlDataAdapter();

MySqlCommand command = new MySqlCommand("SELECT \* FROM taxi.crew JOIN taxi.car ON taxi.car.id = taxi.crew.car\_id join taxi.driver on taxi.crew.driver\_id = taxi.driver.driver\_id join taxi.color on taxi.car.color\_id = taxi.color.id join taxi.model on taxi.model.id = taxi.car.model\_id WHERE 1 ORDER BY RAND() LIMIT 1", db.getConnection());

adapter.SelectCommand = command;

adapter.Fill(table);

db.openConnection();

textBox5.Text = table.Rows[0].ItemArray[10].ToString();

textBox6.Text = table.Rows[0].ItemArray[9].ToString();

textBox3.Text = table.Rows[0].ItemArray[17].ToString();

textBox4.Text = table.Rows[0].ItemArray[7].ToString();

TextBox1.Text = table.Rows[0].ItemArray[20].ToString();

textBox2.Text = table.Rows[0].ItemArray[19].ToString();

db.closeConnection();

}

}

}

namespace taxi

{

public partial class RegisterForm : Form

{

public RegisterForm()

{

InitializeComponent();

userNameField.Text = "Введите имя";

userSurnameField.Text = "Введите фамилию";

}

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

{

Application.Exit();

}

Point LastPoint;

private void panel1\_MouseMove(object sender, MouseEventArgs e)

{

if (e.Button == MouseButtons.Left)

{

this.Left += e.X - LastPoint.X;

this.Top += e.Y - LastPoint.Y;

}

}

private void panel1\_MouseDown(object sender, MouseEventArgs e)

{

LastPoint = new Point(e.X, e.Y);

}

private void Exit\_MouseEnter(object sender, EventArgs e)

{

Exit.ForeColor = Color.Black;

}

private void Exit\_MouseLeave(object sender, EventArgs e)

{

Exit.ForeColor = Color.Transparent;

}

private void userNameField\_Enter(object sender, EventArgs e)

{

if (userNameField.Text == "Введите имя")

{

userNameField.Text = " ";

}

}

private void userNameField\_Leave(object sender, EventArgs e)

{

if (userNameField.Text == " ")

{

userNameField.Text = "Введите имя";

}

}

private void userSurnameField\_Enter(object sender, EventArgs e)

{//userSurnameField.Text = "Введите фамилию"

if (userSurnameField.Text == "Введите фамилию")

{

userSurnameField.Text = " ";

}

}

private void userSurnameField\_Leave(object sender, EventArgs e)

{

if (userSurnameField.Text == " ")

{

userSurnameField.Text = "Введите фамилию";

}

}

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

{

if (userNameField.Text == "Введите имя")

{

MessageBox.Show("Введите имя");

return;

}

if (userSurnameField.Text == "Введите фамилию")

{

MessageBox.Show("Введите фамилию");

return;

}

if (loginField.Text == " ")

{

MessageBox.Show("Введите телефон");

return;

}

if (passField.Text == " ")

{

MessageBox.Show("Введите пароль");

return;

}

if (checkUser())

return;

DB db = new DB();

MySqlCommand command = new MySqlCommand("INSERT INTO taxi.client (f\_name, l\_name, telephone, password) VALUES (@name, @surname, @login, @pass)", db.getConnection());

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

command.Parameters.Add("@name", MySqlDbType.VarChar).Value = userNameField.Text;

command.Parameters.Add("@surname", MySqlDbType.VarChar).Value = userSurnameField.Text;

command.Parameters.Add("@pass", MySqlDbType.VarChar).Value = passField.Text;

db.openConnection();

if(command.ExecuteNonQuery() == 1)

{

MessageBox.Show("Аккаунт был создан");

}

else

{

MessageBox.Show("Аккаунт не был создан");

}

db.closeConnection();

}

public Boolean checkUser()

{

DB db = new DB();

DataTable table = new DataTable();

MySqlDataAdapter adapter = new MySqlDataAdapter();

MySqlCommand command = new MySqlCommand("SELECT \* FROM taxi.client WHERE telephone = @uL", db.getConnection());

command.Parameters.Add("@uL", MySqlDbType.VarChar).Value = loginField.Text;

adapter.SelectCommand = command;

adapter.Fill(table);

if (table.Rows.Count > 0)

{

MessageBox.Show("Данный телефон зарегестрирован");

return true;

}

else

{

return false;

}

}

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

{

this.Hide();

loginFotn loginFotn = new loginFotn();

loginFotn.Show();

}

}

}

namespace taxi

{

public partial class loginFotn : Form

{

public loginFotn()

{

InitializeComponent();

}

private void label2\_Click(object sender, EventArgs e)

{

Application.Exit();

}

private void label2\_MouseEnter(object sender, EventArgs e)

{

label2.ForeColor = Color.Black;

}

private void label2\_MouseLeave(object sender, EventArgs e)

{

label2.ForeColor = Color.Transparent;

}

Point LastPoint;

private void panel1\_MouseMove(object sender, MouseEventArgs e)

{

if(e.Button == MouseButtons.Left)

{

this.Left += e.X - LastPoint.X;

this.Top += e.Y - LastPoint.Y;

}

}

private void panel1\_MouseDown(object sender, MouseEventArgs e)

{

LastPoint = new Point(e.X, e.Y);

}

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

{

string loginUser = loginField.Text;

string passUser = passField.Text;

DB db = new DB();

DataTable table = new DataTable();

MySqlDataAdapter adapter = new MySqlDataAdapter(); // 74149570266 `taxi`.`client`

MySqlCommand command = new MySqlCommand("SELECT \* FROM taxi.client WHERE telephone = @uL AND password = @uP", db.getConnection());

command.Parameters.Add("@uL", MySqlDbType.VarChar).Value = loginUser;

command.Parameters.Add("@uP", MySqlDbType.VarChar).Value = passUser;

adapter.SelectCommand = command;

adapter.Fill(table);

if(table.Rows.Count > 0)

{

this.Hide();

MainForm mainForm = new MainForm();

mainForm.Show();

}

else

{

MessageBox.Show("No");

}

}

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

{

this.Hide();

RegisterForm registerForm = new RegisterForm();

registerForm.Show();

}

}

}