|  |
| --- |
| МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  федеральное государственное АВТОНОМНОЕ образовательное учреждение высшего образования  «Национальный исследовательский ядерный университет «МИФИ» |
| **Димитровградский инженерно-технологический институт –**  филиал федерального государственного автономного образовательного учреждения высшего образования «Национальный исследовательский ядерный университет «МИФИ»  **(ДИТИ НИЯУ МИФИ)** |

**Специальность 09.02.07 Информационные системы и программирование**

**МДК.01.01 Разработка программных модулей**

VS + Windows Forms + SQL

Выполнил студенты \_3\_ курса \_331\_ группы

Казаков Матвей Олегович

(Ф.И.О. студента)

Преподаватель ДИТИ НИЯУ МИФИ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_А.В. Надеждина

Работа сдана 17.04.2023г

Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Димитровград, 2023

# СОДЕРЖАНИЕ

[Введение 3](#_Toc132572542)

[Теоретическая часть 4](#_Toc132572543)

[Практическая часть 5](#_Toc132572544)

[Заключение 6](#_Toc132572545)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 8](#_Toc132572546)

# Введение

Данная работа заключается в чуть более углубленном взаимодействии с языком программирования C# и базами данных MySQL. C# является одним из актуальных на данный момент языков программирования и занимает среди всех лидирующие позиции. Использование баз данных MySQL, так же не являются редкостью.

Тема данной работы – Ремонтная мастерская электронного оборудования. Актуальность работы обусловлена потребностью использования данной работы в мастерских. Для облегчения введения учета работ.

Цель данной работы заключается в создании проекта на языке программирования C#, являющимся журналом ведения работ. Приложение должно включать в себя базу данных MySQL, состоящую из шести таблиц: список поступившего оборудования, список работ, список мастеров, список запасных частей, журнал учета использования запасных частей, список клиентов.

Для чёткого выполнения цели работы был сформулирован следующий список задач:

* создание окна авторизации;
* подключение MySQL и создание базы данных;
* обеспечение авторизации пользователя через базу данных;
* создание окна регистрации;
* обеспечение регистрации пользователя в базе данных;
* переход между окнами приложения;
* вывод таблицы на форму приложения, после авторизации.

# Теоретическая часть

Для выполнения работы была выбрана среда разработки VisualStudio, и интерфейс WindowsForms языка C#. Это отличный вариант для новичков. WindowsForms сильно упрощает процесс создания графических приложений, за счёт готовых компонентов, лёгкого изменения свойств, и добавления событий. В общем Windows Forms выполняет всю работу по созданию интерфейса автоматически, разработчику же остаётся лишь его настройка.

Плюсы использования языка программирования C#:

1. Независимость от аппаратного функционала. Программу не нужно адаптировать под многочисленные платформы и операционные системы. Виртуальная машина .NET Framework сама выполняет эту задачу. Это позволяет использовать один и тот же код на самых разных устройствах: смартфонах, компьютерах, серверах, банкоматах, умных часах и т.д.

2. Поддержка. Данный язык был создан компанией Microsoft. Именно поэтому C# лучше всего подходит для работы с экосистемой Windows.

3. Управление памятью. Язык программирования C# выполняет очистку памяти программы в автоматическом режиме. Таким образом, не нужно контролировать расход памяти, устранять её утечки или удалять «мёртвые» куски кода.

4. Строгая типизация. Если вы объявляете переменную в C#, необходимо сначала указать, что в ней лежит — строка, число или массив. Разработка таким методом выполняется немного дольше. Однако написанный код становится более предсказуемым, так как числа взаимодействуют с числами, строки со строками и т.д. Да, языки со слабой типизацией предоставляют большую свободу действий, но вероятность допустить ошибку при создании или проверке кода также будет выше.

5. Крупное сообщество. С# используется миллионами программистов. В социальных сетях можно найти многочисленные чаты и сообщества «шарпистов». Там есть ответы на все важные вопросы по поводу работы с языком. Более того, можно найти ментора, который будет обучать вас.

Работа с базой данных MySQL была реализована с помощью программы MAMP, создающей локальный сервер и позволяющий пользоваться phpMyAdmin для создания, редактирования, а также настройки связей базы данных с помощью графического интерфейса. MySQL имеет может работать с различными языками, работает с разными типами таблиц и имеет высокую скорость выполнения операций среди одноклассников. Созданные таблицы имеют тип MyISAM. Выбор использованного типа был обусловлен тем, что работает данный тип независимо от платформы, а также имеет высокую скорость выполнения запросов на выборку данных.

Преимущества использования базы данных MySQL:

1. Производительность. Для веб-приложений важна скорость работы. Пользователь не будет ждать загрузки страницы — он просто уйдет с неё, если приложение не будет работать достаточно быстро. В плане производительности приложения «болезненным» местом является база данных, и MySQL не подведет вас.
2. Кроссплатформенность. MySQL предоставляет кроссплатформенную базу данных: она работает на Linux, FreeBSD и конечно на Windows. Этот критерий нужно учитывать при выборе СУБД для проектов, нацеленных на несколько платформ, в частности веб-приложений. К слову, MySQL входит в состав популярных стеков для разработки веб-приложений — LAMP (Linux, Apache, MySQL, PHP) и WAMP (Windows, Apache, MySQL, PHP).
3. Доступность СУБД. Oracle развивает MySQL как проект с открытым исходным кодом, что позволяет сообществу вносить в него свой вклад и, при необходимости, адаптировать его под свои задачи. СУБД предоставляется для конечных пользователей бесплатно.
4. Выбор движков. В MySQL есть два типа движков: транзакционные и нетранзакционные. Транзакционные движки хранения данных подразумевают, что операции в этих БД могут быть отменены, если они не завершены.
5. Безопасность. MySQL предлагает пользователям высокий уровень безопасности. В MySQL есть встроенные инструменты безопасности, которые поддерживают управление пользователями и их привилегиями. При недостатке стандартных инструментов пользователь всегда может установить дополнительные плагины.

Система контроля версий Git и GitHub. Git – распределённая система контроля версий, позволяющая разработчикам совместно работать над одним проектом и отслеживать изменения в файлах. GitHub является сервисом онлайн хостинга репозиториев. Он поддерживает весь функционал Git, и даже больше. GitHub представляет много возможностей в плане обучения, работы и взаимодействия с другими разработчиками. Взаимодействие с GitHub осуществляется с помощью GitBash.

Еще один плюс выбора данной среды разработки, что позволило снять дополнительную нагрузку на пк в виде еще одного приложения или вкладки браузера, а также довольно быстро загружать изменения и не отвлекаться между открытыми приложениями.

# Практическая часть

Создал форму авторизации и спроектировал дизайн для приложения «Workshop». Был добавлен цвет фона, текст, два поля для ввода, кнопка, а также некоторые элементы дизайна (рис. 1).

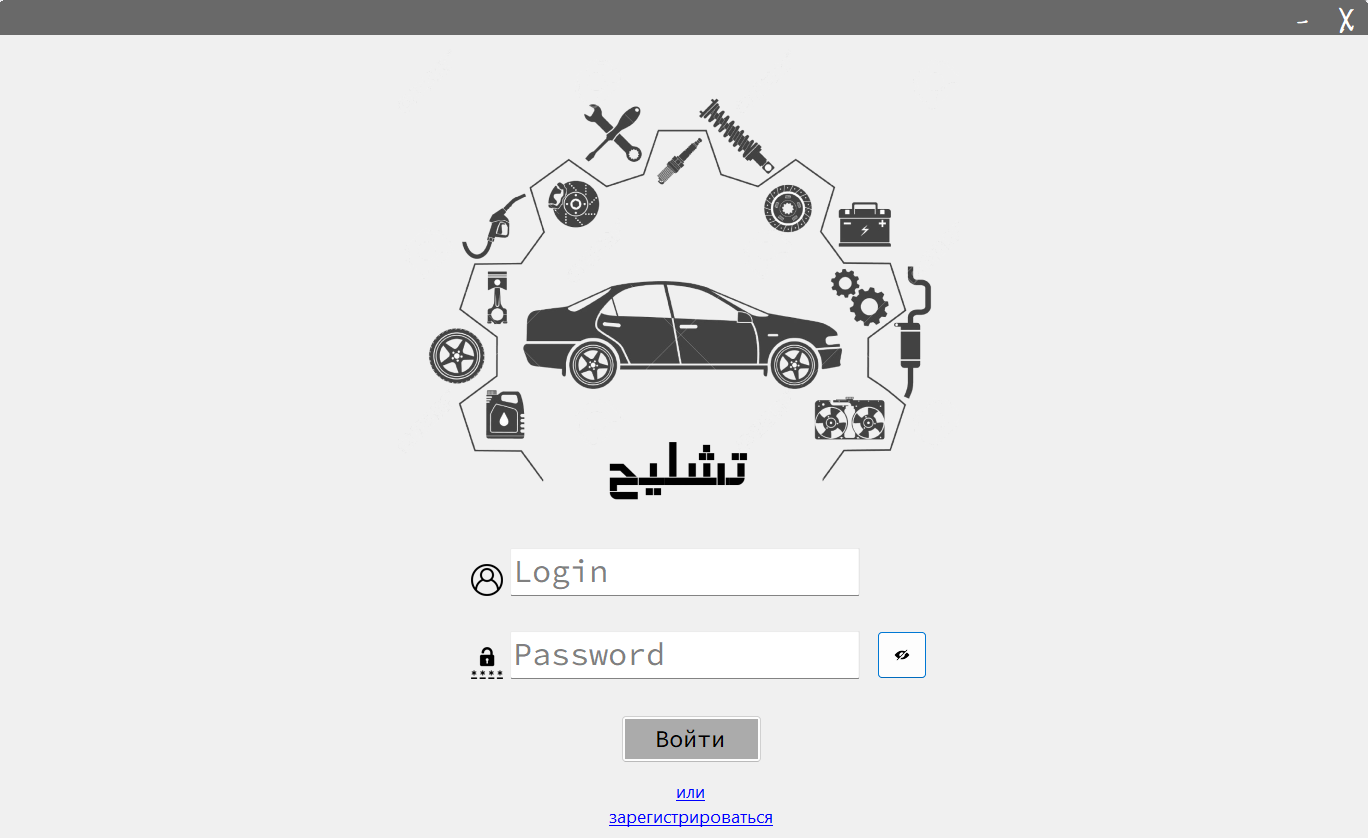


Рисунок 1 – Авторизация приложения «Workshop»

Написал код формы авторизации приложения «Workshop» (рис. 2)



Рисунок 2 – Код формы авторизации «Workshop»

Подключил MySql с помощью MySqlConnector скачанный через VS NuGet, а для работы с базой данных было выбрано приложение MAMP (рис. 3-4).

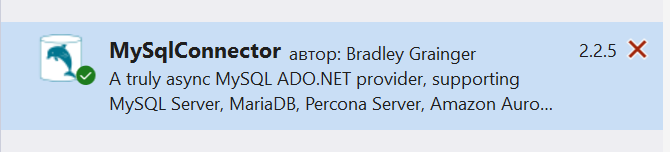


Рисунок 3 – Connector MySql

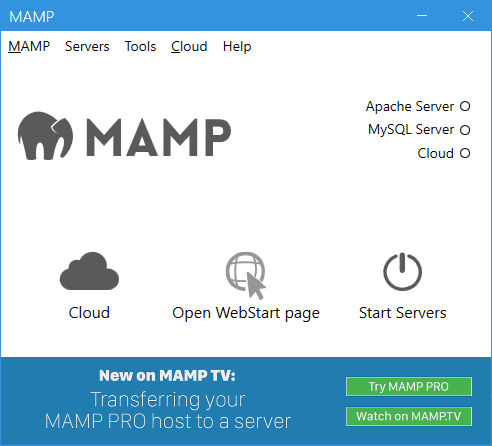


Рисунок 4 – Программа MAMP

Для подключения к локальной базе данных был написан код (рис. 5).

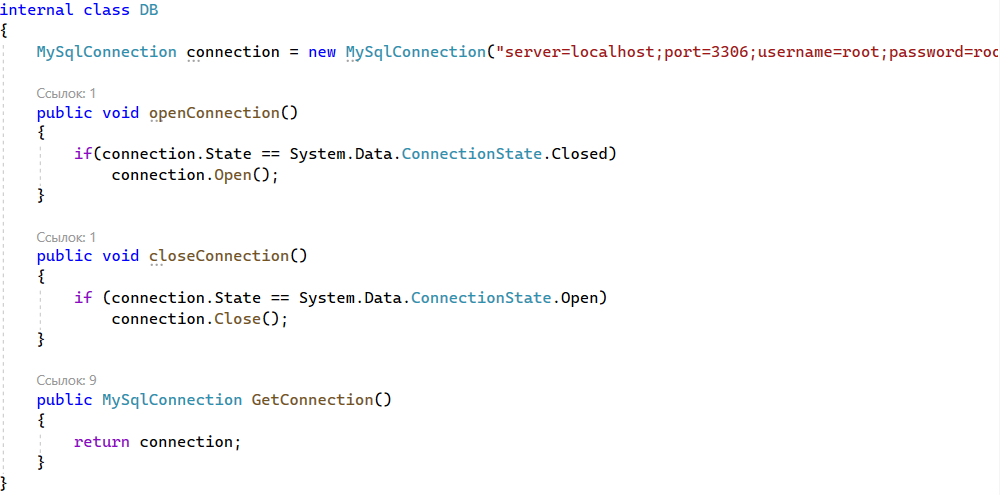


Рисунок 5 – Код подключения базы данных

В phpMyAdmin создал базу данных «workshop» и создал таблицу «clients», которая будет использоваться для хранения регистрационных данных о пользователе (рис. 6).



Рисунок 6 – Таблица «clients»

Написал код для авторизации пользователя на форме «Авторизация» (рис. 7-8).



Рисунок 7 – Код для авторизации пользователя

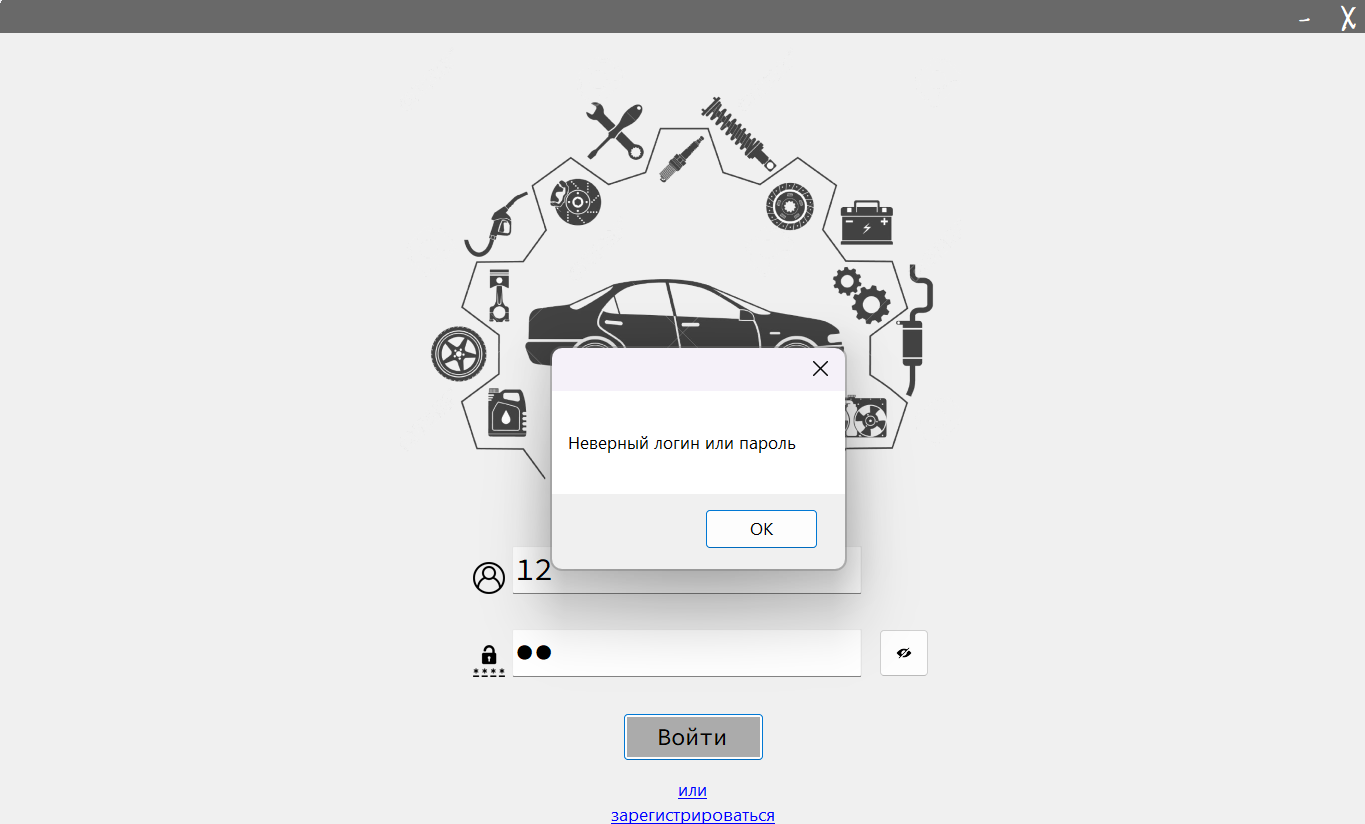


Рисунок 8 – Проверка авторизации

Создал дизайн формы «Регистрация». Было добавлено 6 полей ввода данных, две кнопки, и текст. (рис. 9).

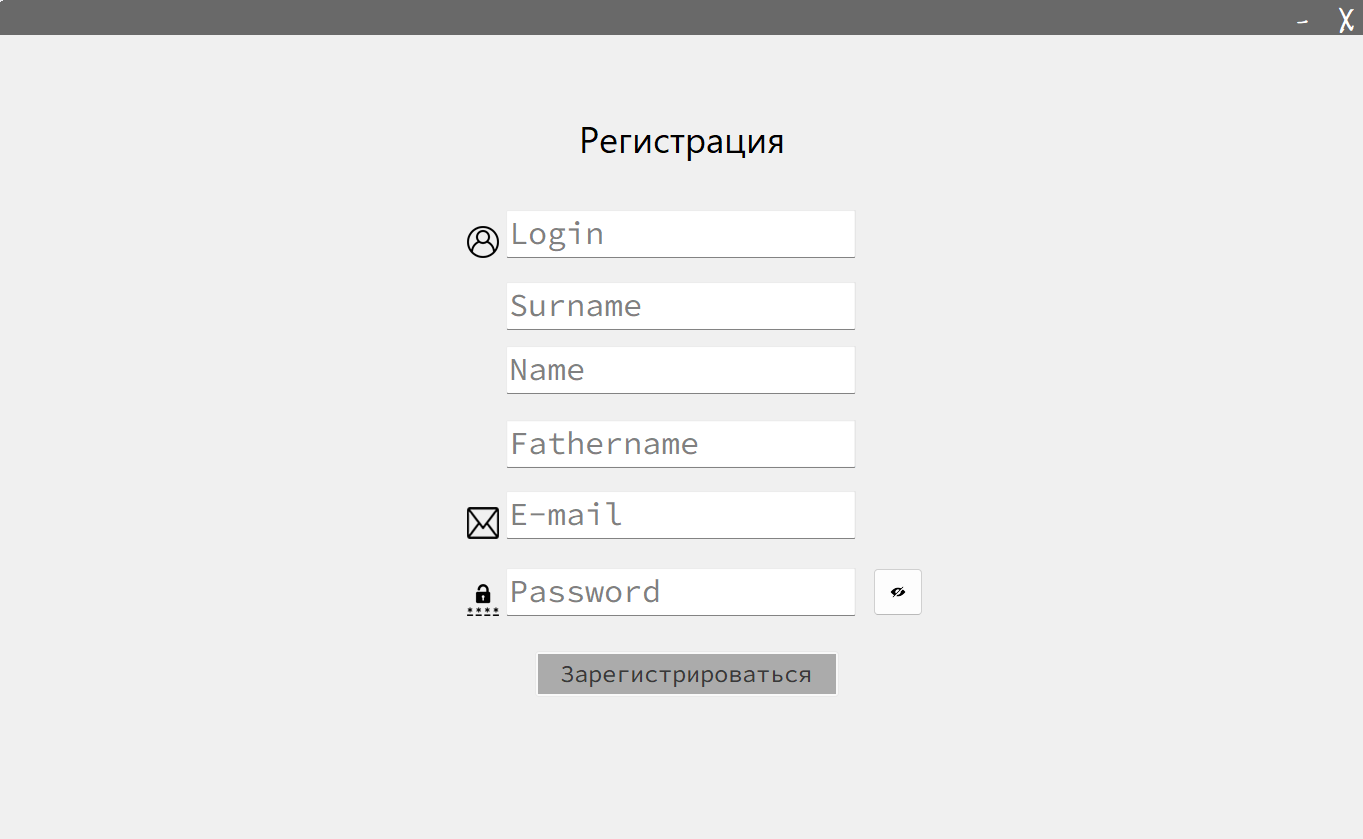


Рисунок 9 – Форма «Регистрация»

Написал код формы «Регистрация» (рис. 10-11).

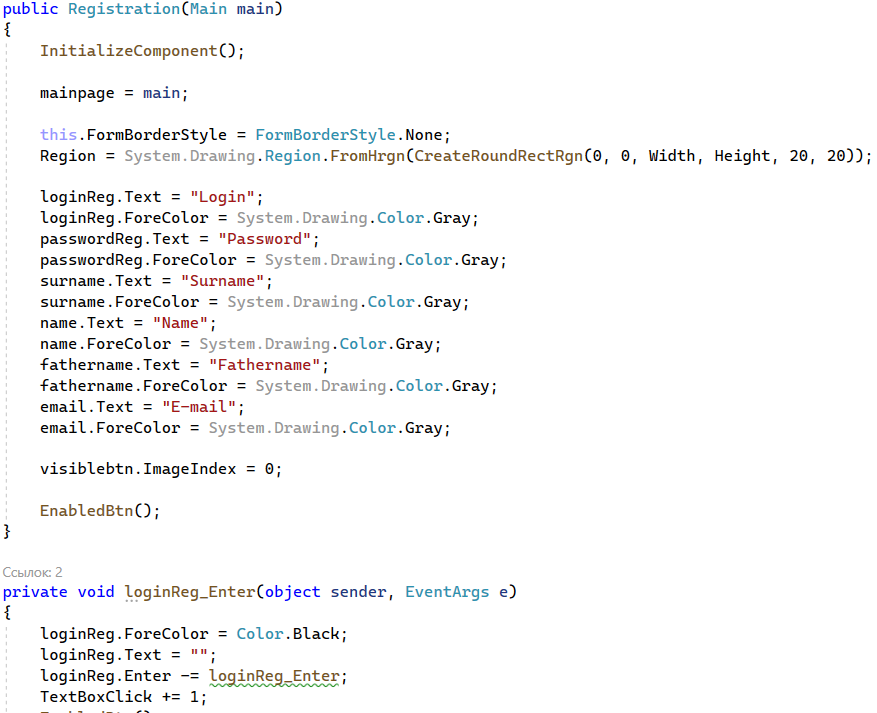


Рисунок 10 – Код формы регистрация

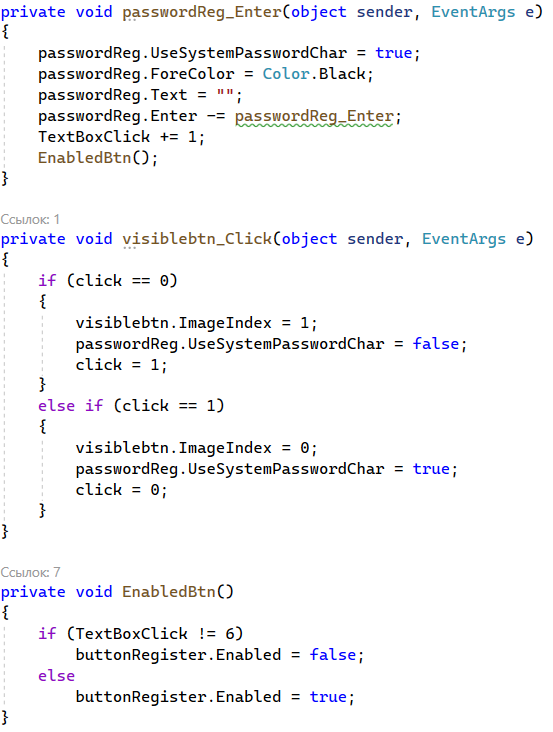


Рисунок 11 – Код формы регистрация

Написал код для регистрации пользователя на форме «Регистрация» (рис. 12).

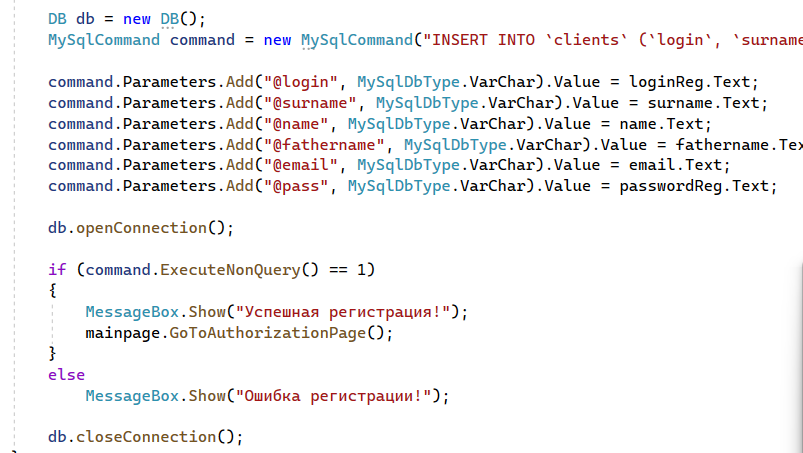


Рисунок 12 – Код для регистрации пользователя

Написал код для проверки на уже существующего пользователя для формы «Регистрация» (рис. 13).



Рисунок 13 – Код проверка пользователя

Провёл проверку формы «Регистрация» на подключение к базе данным, к чтению и правильностью записи данных в неё (рис. 14-18).

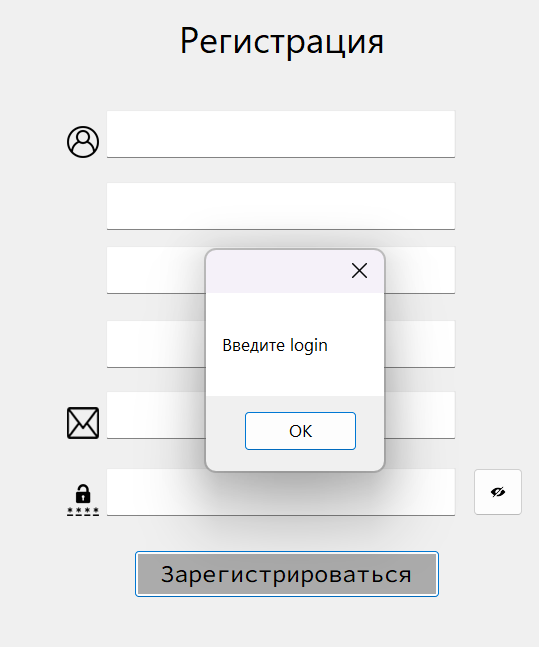


Рисунок 14 – Проверка на ввод данных

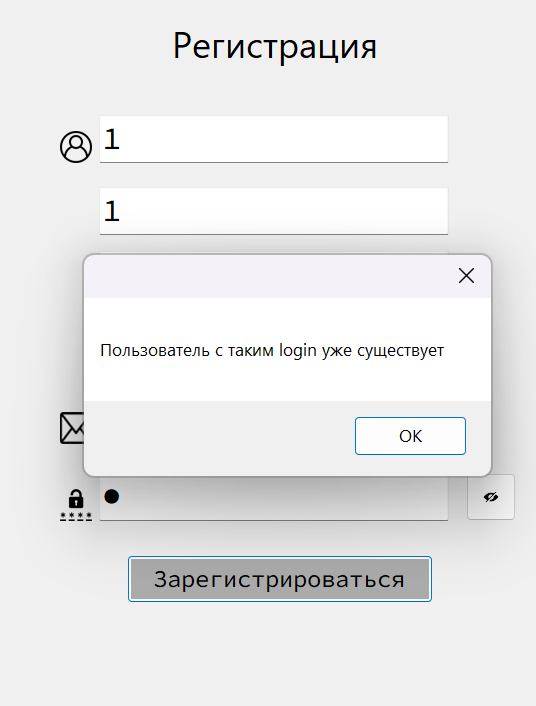


Рисунок 15 – Проверка на существующий login

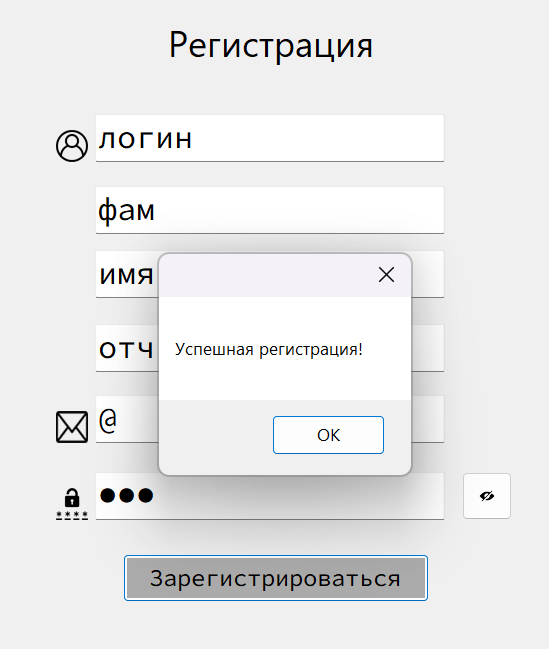


Рисунок 16 – Успешная регистрация пользователя в системе



Рисунок 17 – Проверка наличия созданной записи в базе данных

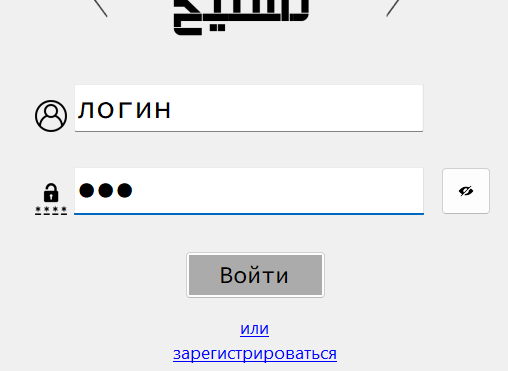


Рисунок 18 – Проверка авторизации после регистрации

Создал метод для перехода между формами (рис. 19).

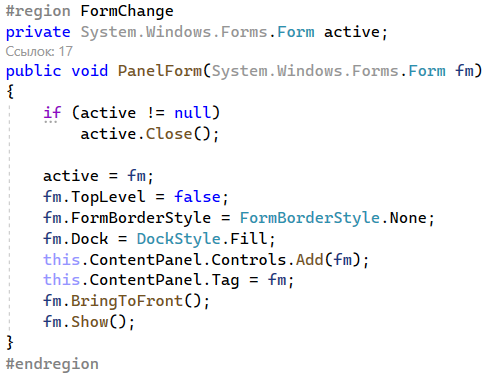


Рисунок 19 – Метод для перехода между формами

Переход между формами осуществляется с помощью кнопок на главной форме (рис. 20-21).

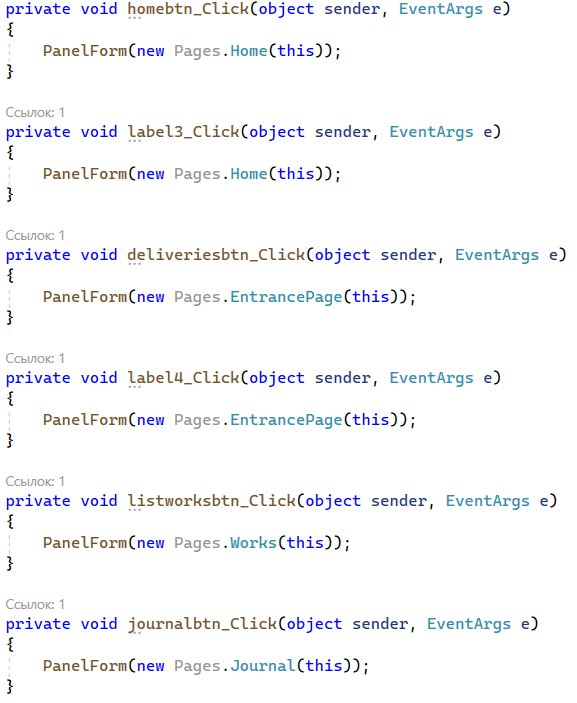


Рисунок 20 – Код кнопок для перехода между формами

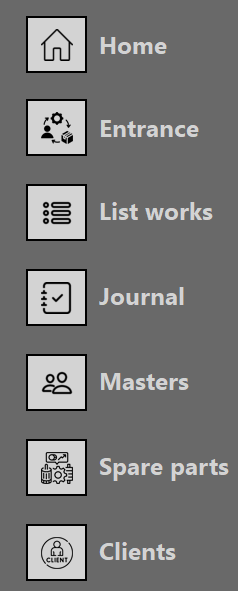


Рисунок 21 – Кнопки навигации по приложению

Создал главную форму «main», на которой осуществляется вывод всех форм приложения, а также кнопки сворачивания приложения и выхода из неё (рис. 22).

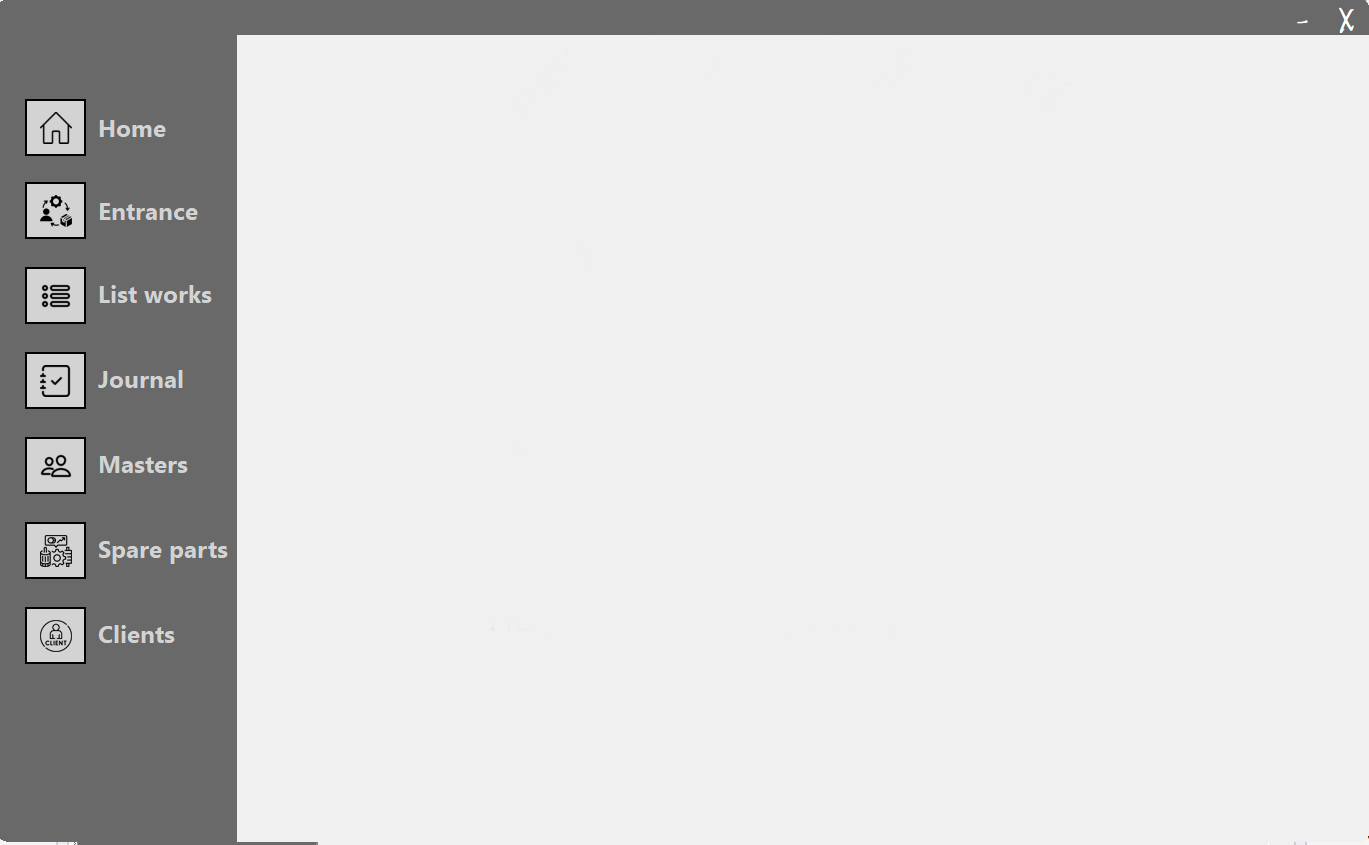


Рисунок 22 – Главная форма «main»

Создал форму «Home», на которой выведено приветствие авторизированного пользователя (рис. 23).

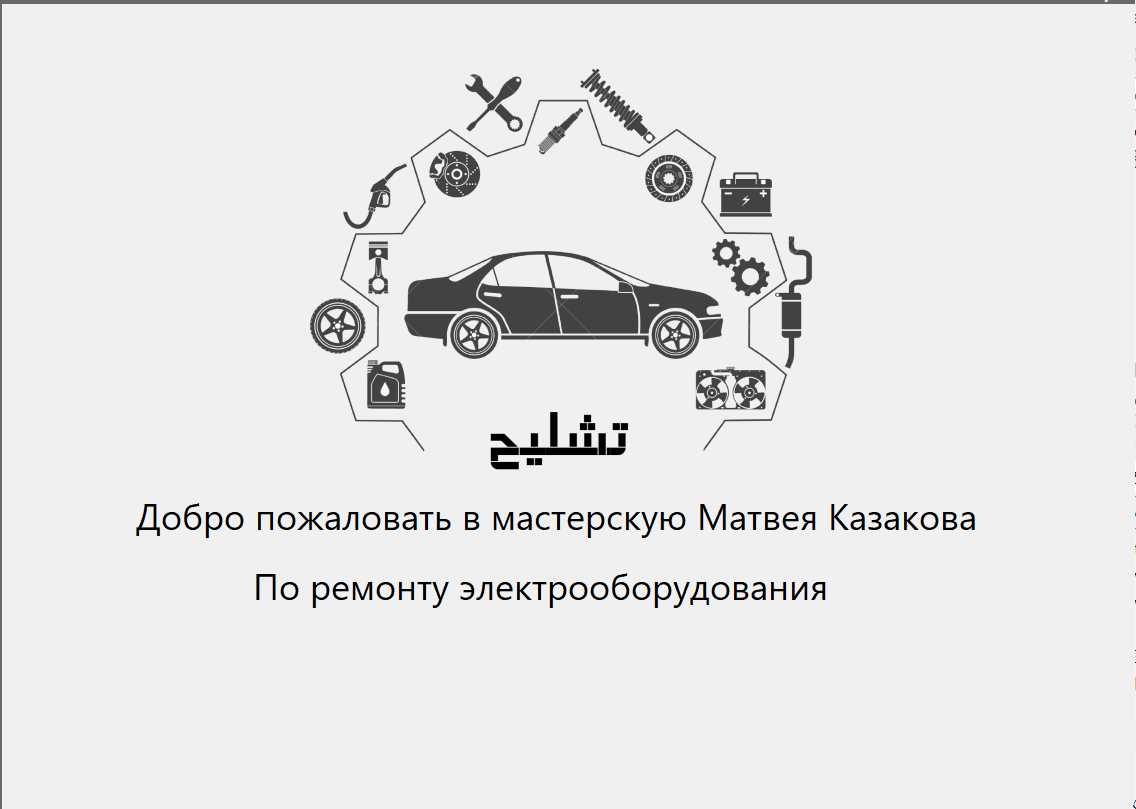


Рисунок 23 – Форма «Home»

Далее были созданы таблицы в базе данных «workshop». Были созданы такие таблицы как: «Клиенты», «Оборудование», «Мастера», «Запчасти», «Список работ», «Журнал использованных деталей в работах». Установлены связи между таблицами. И были заполнены данными (рис. 24-31).

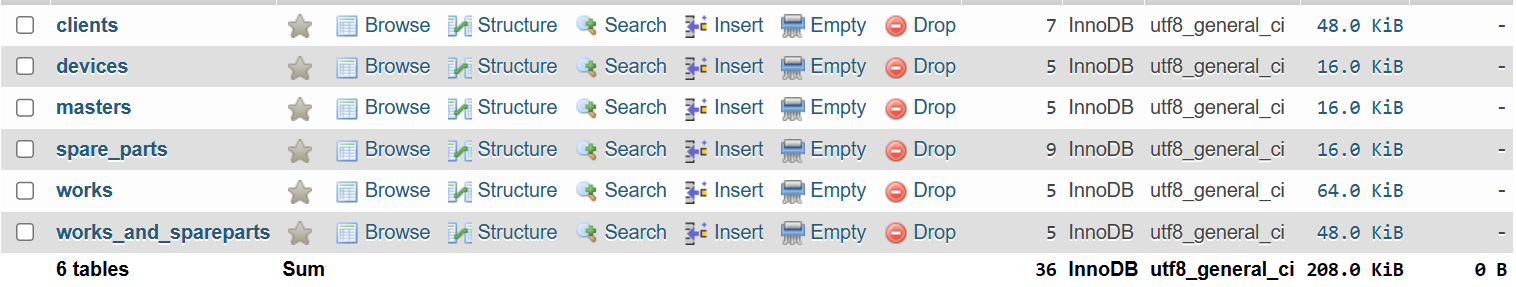


Рисунок 24 – Таблицы базы данных «workshop»

Рисунок 25 – Структура таблицы «Клиенты»

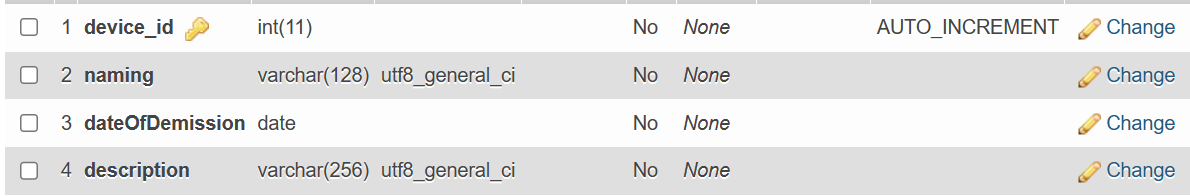


Рисунок 26 – Структура таблицы «Оборудование»



Рисунок 27 – Структура таблицы «Мастера»

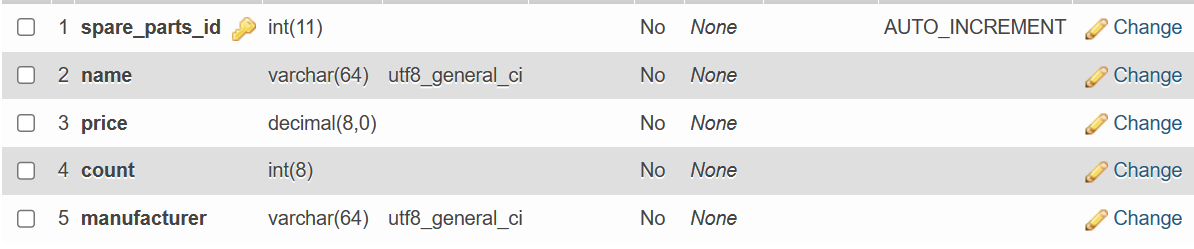


Рисунок 28 – Структура таблицы «Запчасти»



Рисунок 29 – Структура таблицы «Список работ»

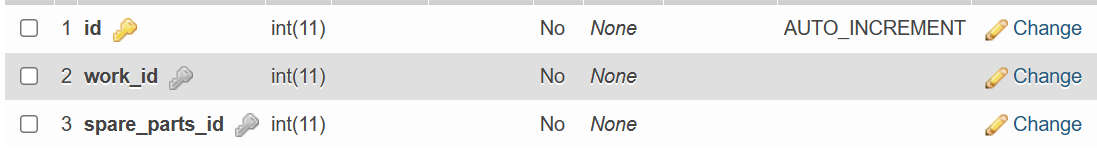


Рисунок 30 – Структура таблицы «Журнал использованных деталей в работах»

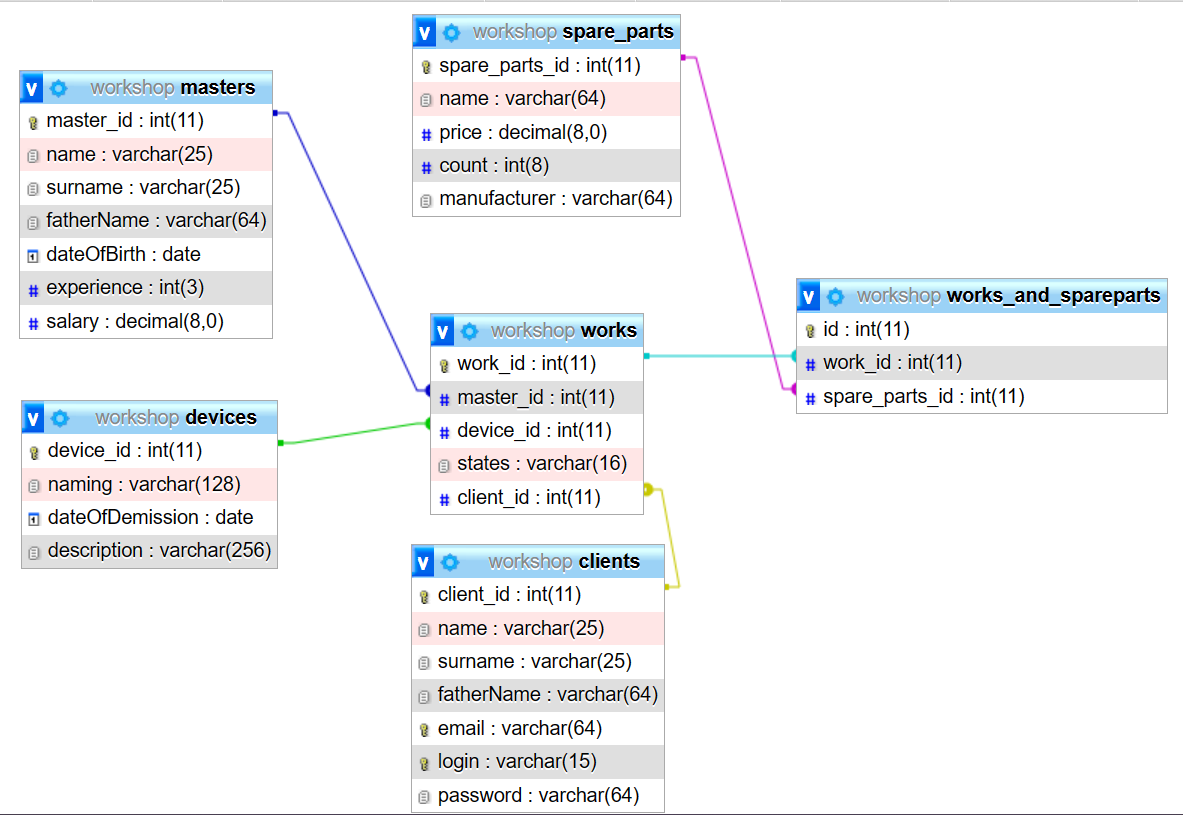


Рисунок 31 - Связи базы данных «workshop»

Заполнение таблиц данными (рис. 32-37).



Рисунок 32 – Записи в таблице «Клиенты»

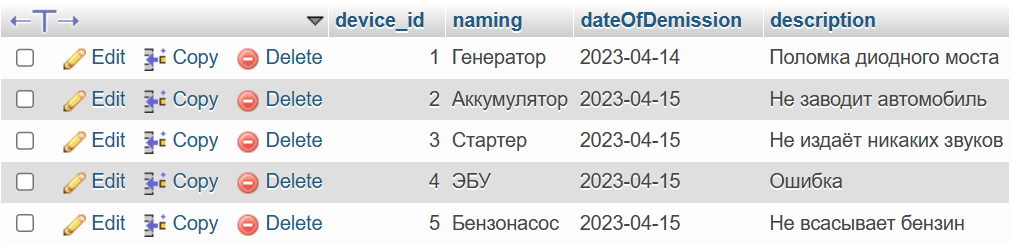


Рисунок 33 – Записи в таблице «Оборудование»



Рисунок 34 – Записи в таблице «Мастера»



Рисунок 35 – Записи в таблице «Запчасти»

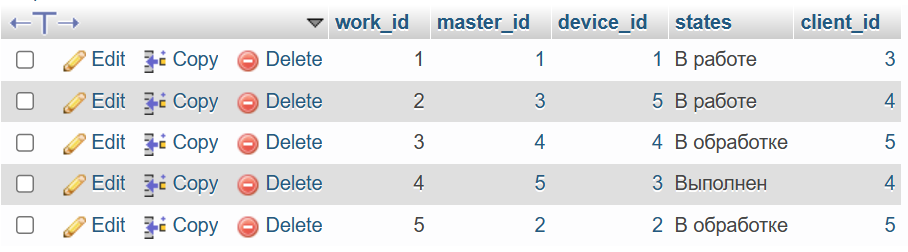


Рисунок 36 – Записи в таблице «Список работ»

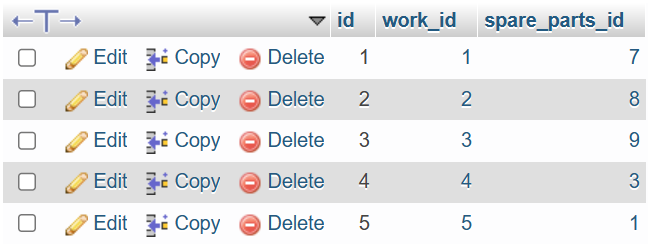


Рисунок 37 – Записи в таблице «Журнал использованных деталей в работах»

Создал формы для вывода таблиц (которые будут выводится через главную форму «main» созданная ранее) такие как: «EntrancePage», «Works», «Journal», «Masters», «Spareparts», «Clients». Также был написан код для вывода таблиц (рис. 38-51).

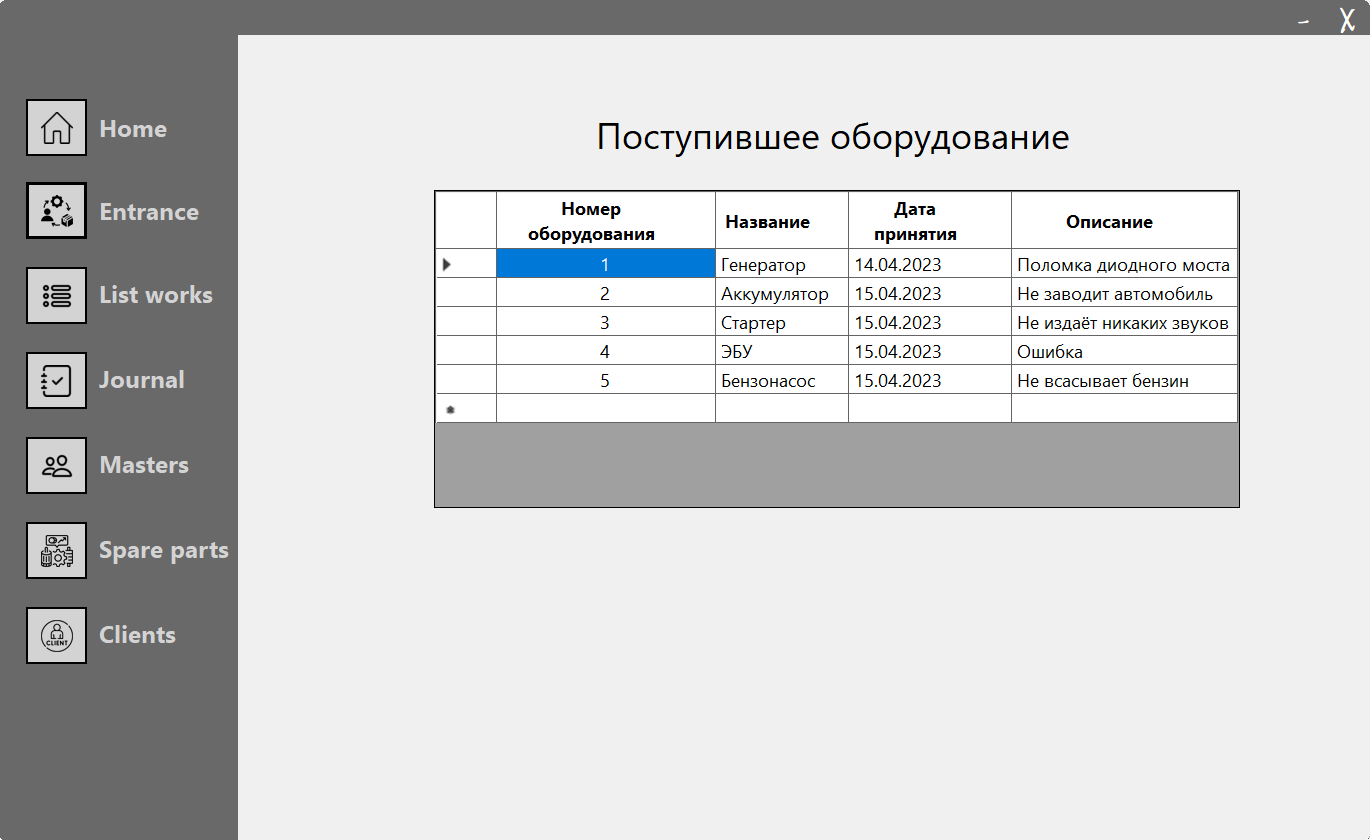


Рисунок 38 – Форма «EnrancePage», вывод таблицы «Оборудование»



Рисунок 39 – Код для вывода таблицы «Оборудование»

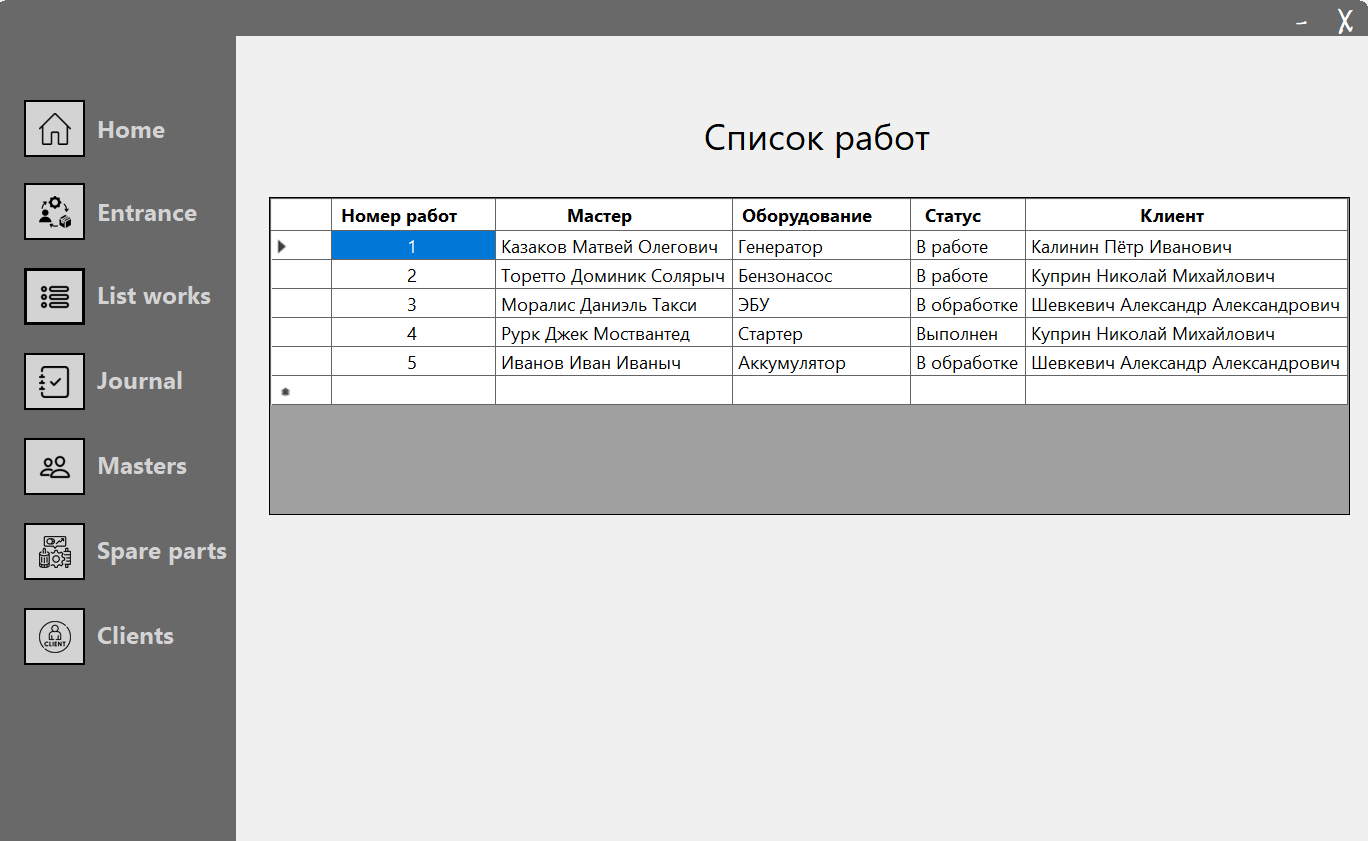


Рисунок 40 – Форма «Works», вывод таблицы «Список работ»



Рисунок 41 – Код для вывода таблицы «Список работ»

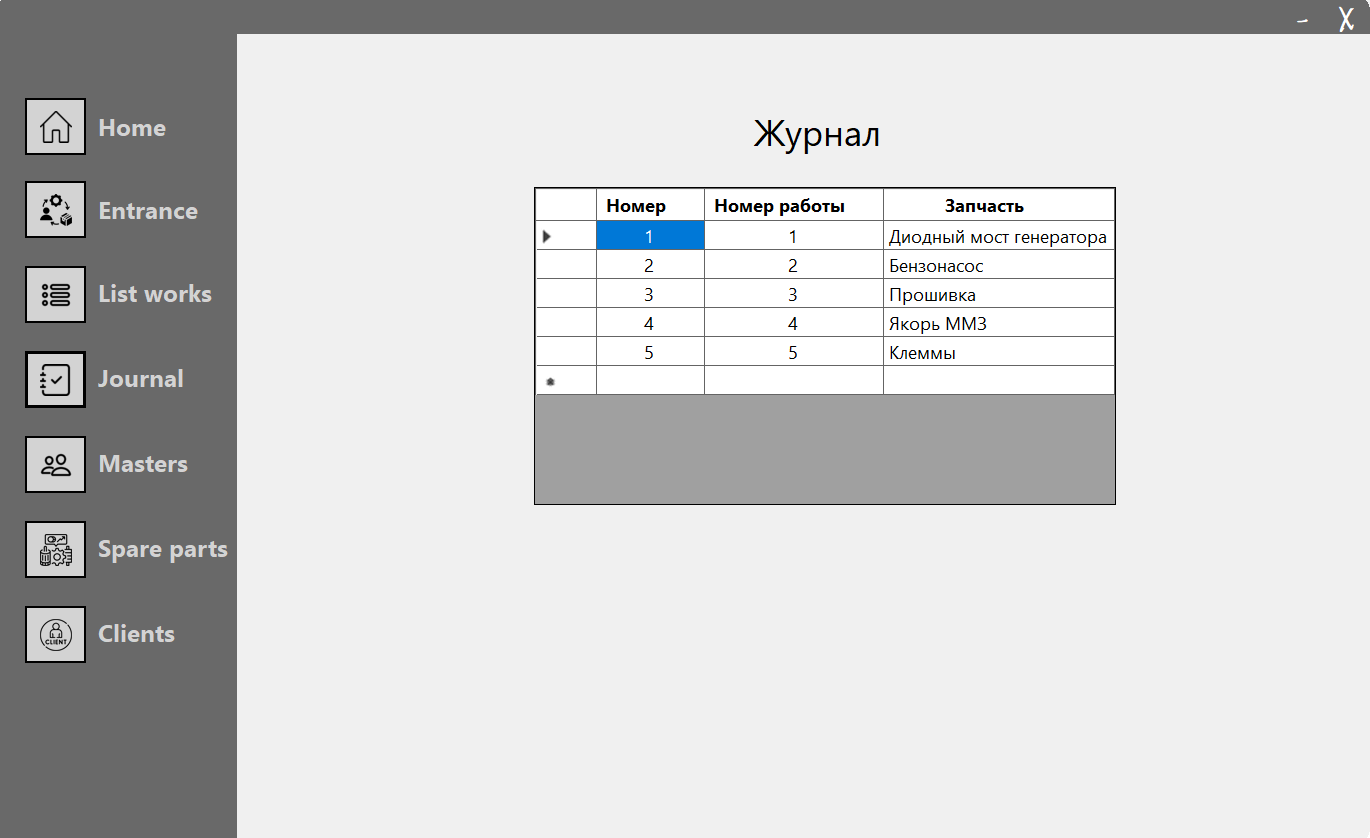


Рисунок 42 – Форма «Journal», вывод таблицы «Журнал использованных деталей в работах»



Рисунок 43 – Код для вывода таблицы «Журнал использованных деталей в работах»

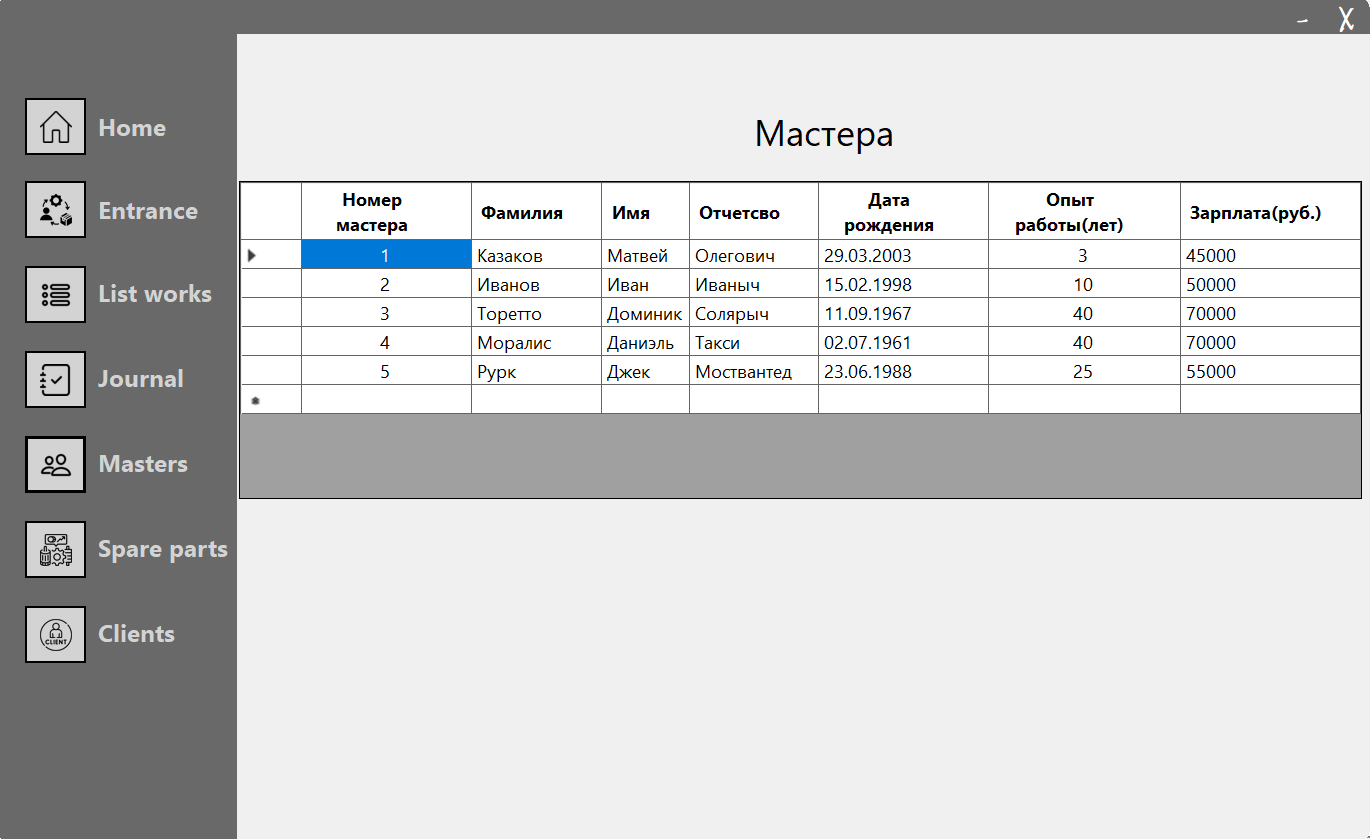


Рисунок 44 – Форма «Masters», вывод таблицы «Мастера»



Рисунок 45 – Код для вывода таблицы «Мастера»

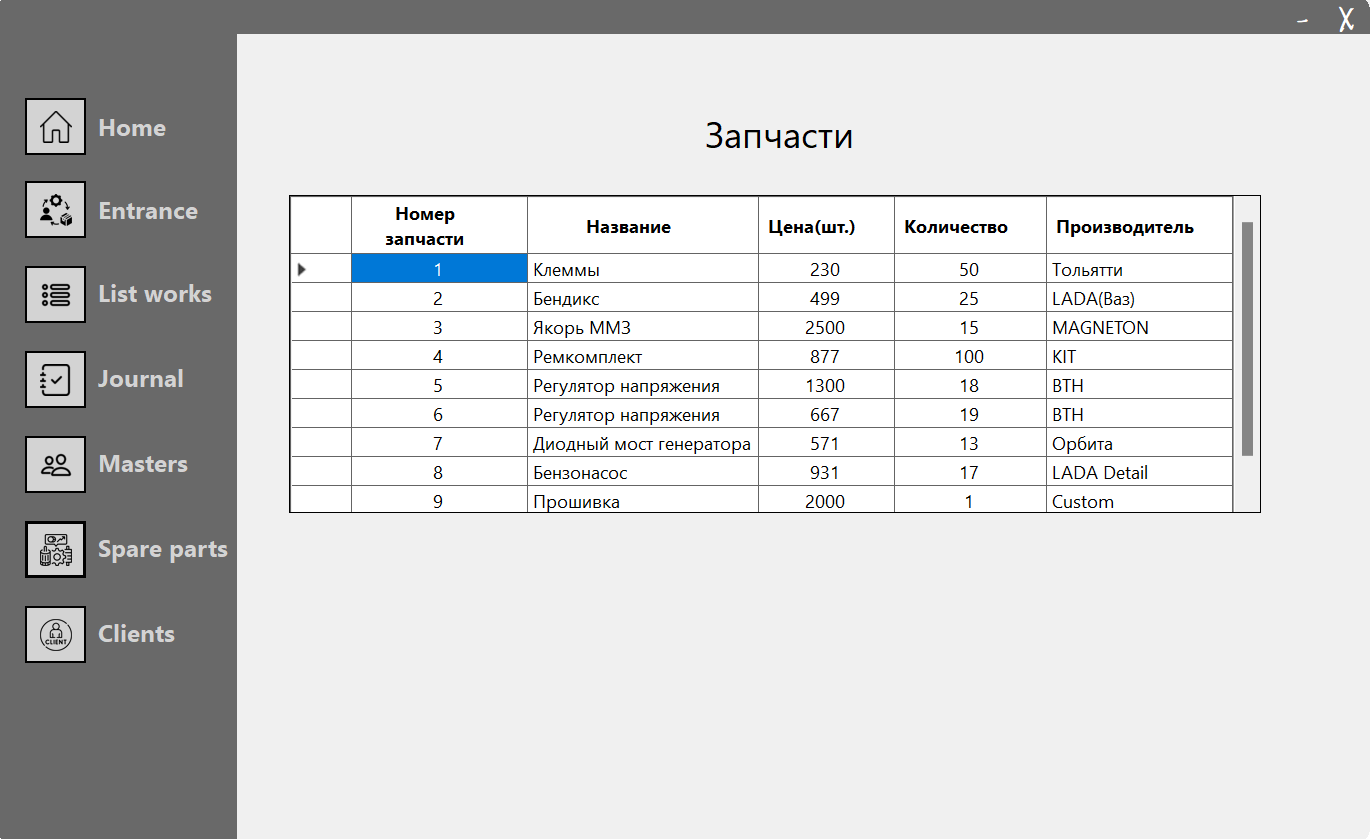


Рисунок 46 – Форма «Spareparts», вывод таблицы «Запчасти»



Рисунок 47 – Код для вывода таблицы «Запчасти»

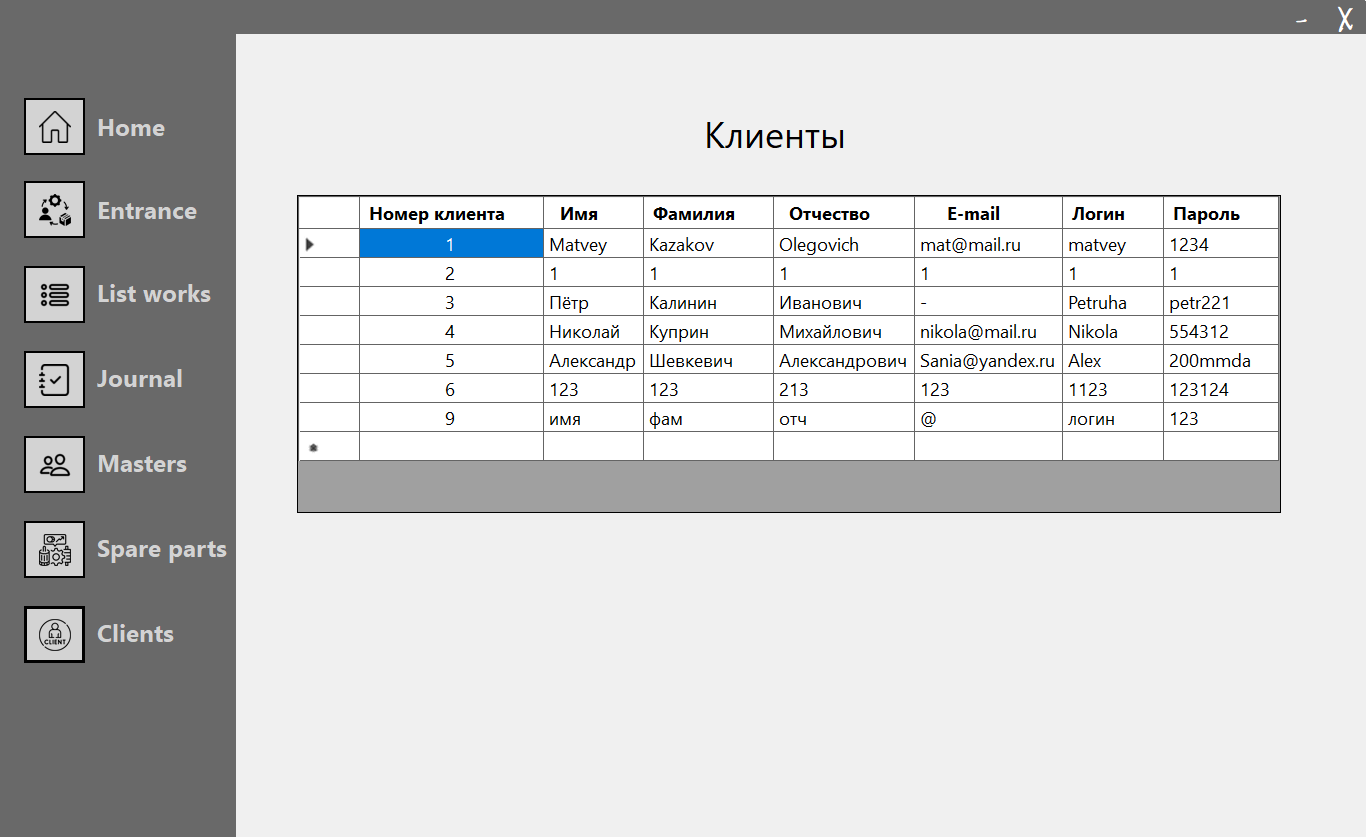


Рисунок 48 – Форма «Clients», вывод таблицы «Клиенты»



Рисунок 49 – Код для вывода таблицы «Клиенты»

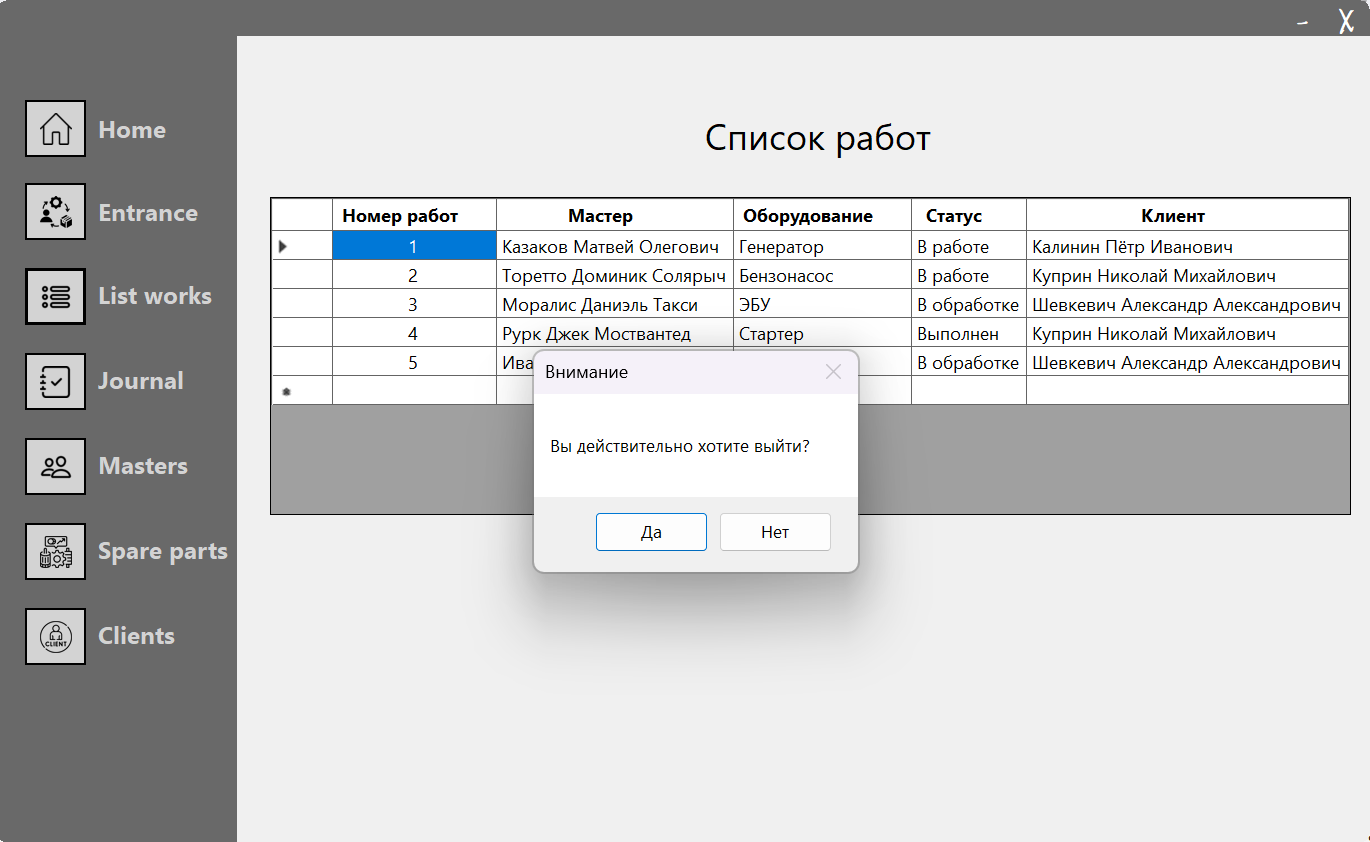
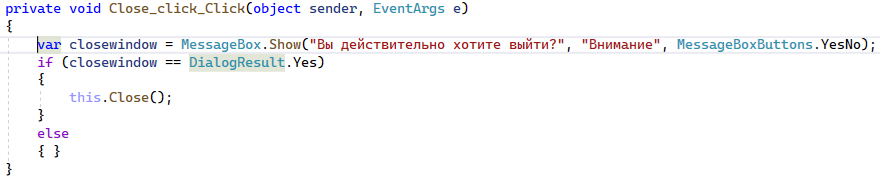


Рисунок 50 – Предупреждение о выходе из системы

Рисунок 51 – Код для кнопки выхода из системы

# Заключение

Целью данной работы является разработка приложения на языке C# с использованием приложения для создания графических интерфейсов WindowsForms, а также создание базы данных MySQL, состоящую из 5 таблиц. В качестве среды разработки была использована Microsoft VisualStudio. Во время выполнения работы были изучены соответствующие ресурсы и отобраны наиболее удачные материалы, сильно облегчившую работу.

Перед созданием приложения, было принято решение о создании базы данных, а также ER-диаграммы для неё. База данных состоит из таких таблиц как: «клиенты», «оборудование», «мастера», «запчасти», «список работ», «журнал учета использования запчастей». Работа с базой данных происходила с помощью графического интерфейса, который предоставляет нам phpMyAdmin. Все таблицы имеют тип InnoDB.

Проектирование началось с таблицы «clients», хранящая в себе данные обо всех клиентах системы. Она состоит из семи полей: **client\_id** – идентификатор клиента, **name** – имя клиента, **surname** – фамилия клиента, **fatherName** – отчество клиента, **email** – введённая при регистрации почта клиента, **login** – введенный при регистрации логин клиента, **password** – пароль клиента для входа в систему.

Таблица «device» хранит в себе информацию о поступившем оборудовании. Состоит из четырех полей: **device\_id** – идентификатор оборудования, **naming** – название оборудования, **dateOfDemission** – дата поступления оборудования, **description** – описание поломки.

Таблица «masters» хранит в себе информацию о мастерах. Состоит из семи полей: **master\_id –** идентификатор мастера, **name** – имя мастера, **surname** – фамилия мастера, **fatherName** – отчество мастера, **dateOfBirth** – дата рождения, **experience** – опыт работы, **salary** – зарплата мастера.

Таблица «spare­­\_parts» хранит в себе информацию о запчастях. Состоит из пяти полей: **spare\_parts\_id –** идентификатор запчасти, **name –** название запчасти, **price –** цена запчасти, **count –** количество запчастей, **manufacturer –** производитель запчасти.

Таблица «works» является списком работ. Состоит из пяти полей: **work\_id –** идентификатор работы, **master\_id –** идентификатор мастера, **device\_id** – идентификатор оборудования**, states –** статус работы, **client\_id** – идентификатор клиента.

Таблица «works\_and\_spareparts» хранит в себе информацию об использованной детали в работе. Состоит из трех полей: **id –** идентификатор, **work\_id –** идентификатор работы, **spare\_parts\_id –** идентификатор запчасти.

Следующей задачей стала разработка приложения. Оно состоит из десяти форм, каждая из которых имеет собственное предназначение. Первая и основная форма представляет из себя контейнер, который хранит методы и панель для отображения данных. Основная её цель — это отображение всех остальных форм. Следующая форма – это форма авторизации, она осуществляет вход клиента в систему, с помощью введения логина и пароля, хранящихся в базе данных. Из неё же можно перейти на форму регистрации, в которой пользователь создаёт свой аккаунт, введённые данные вносятся в базу данных. После успешной авторизации пользователь попадает на главную форму, на которой расположено: с левой стороны, меню навигации по приложению, сверху расположен панель, на которой расположены кнопки сворачивания формы и выхода из системы, по центру расположена панель, на которой отображаются остальные формы. Переключение по формам происходит с помощью нажатия на кнопки боковом меню. Рассмотрим оставшиеся вкладки по подробнее.

Вкладка «Home» это главная страница приложения, в ней отображается лишь приветствие.

Далее рассмотри вкладки с отображение данных в таблицах.

Вкладка «Entrance» предназначена для отображения данных о поступившем оборудовании.

Вкладка «List works» предназначена для отображения списка работ.

Вкладка «Journal» предназначена для отображения данных об использованных запчастях в работах.

Вкладка «Masters» предназначена для отображения данных о мастерах.

Вкладка «Spare parts» предназначена для отображения данных о запчастях.

Вкладка «Clients» предназначена для отображения данных обо всех клиентах.

В заключении, можно сказать, что мной была проделана неплохая работа, благодаря которой я усвоил полезные знания, закрепил все знания в работе. Так же я приобрёл практических опыт создания приложения и работы с базой данных, что пригодиться в будущих работах.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Васильев, А.Н. Программирование на C# для начинающих. Основные сведения / А.Н. Васильев. – Москва: Эксмо, 2023. – 592 с.
2. Рихтер, Дж. CLR via C#. Программирование на платформе Microsoft .NET Framework 4.5 на языке C# / Дж. Рихтер. – СПб.: Питер, 2019. – 896 c.
3. Руководство по программированию в Windows Forms. – Режим доступа: https://metanit.com/sharp/windowsforms/ (дата обращения 24.03.2023).
4. Создание базы данных в MySQL. – Режим доступа: https://selectel.ru/blog/tutorials/how-to-create-databases-in-mysql/ (дата обращения 01.04.2023).
5. Создание приложения Windows Forms на C# в Visual Studio. – Режим доступа: https://learn.microsoft.com/ru-ru/visualstudio/ide/create-csharp-winform-visual-studio?sa=X&ved=2ahUKEwjWg5vbyrroAhVXAJ0JHckQBzYQ9QF6BAgEEAI&view=vs-2022 (дата обращения 29.03.2023).