**Министерство образования Московской области**

**ГОУ ВО МО «Государственный гуманитарно-технологический университет» (ГГТУ)**

**Ликино-Дулевский политехнический колледж – филиал ГГТУ**

**О Т Ч Ё Т**

**ПО ПРОИЗВОДСТВЕННОЙ ПРАКТИКЕ**

по ПМ.01 Разработка модулей программного обеспечения для компьютерных систем

Обучающейся Мальчика Артёма Вадимовича

Курс \_\_\_\_3\_\_\_\_\_ группа \_\_\_ИСП.19А\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Специальность 09.02.07 Информационные системы и программирование\_\_\_\_

Место практики\_\_ АО «Стекломаш»

Период практики с 04.05.2023 г. по 31.05.2023 г.

Руководители практики

от колледжа \_\_\_\_\_\_\_\_\_\_Кузьмина Елена Евгеньевна\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Гжегожевский Сергей Владимирович\_\_\_\_\_\_\_\_\_\_

Пронина Алла Юрьевна \_\_\_\_\_\_\_\_\_\_

Селиверстова Ольга Михайловна \_\_\_\_\_\_\_\_\_\_\_

от организации\_\_\_\_\_\_ Дёмин Артём Дмитриевич\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

МП

г. Ликино-Дулево

2023 г.

Содержание

[1. Общие сведения о предприятии 3](#_Toc107235165)

[1.1. Структура организации 4](#_Toc107235166)

[2. Анализ материально-технической базы 5](#_Toc107235167)

[2.1. Состав программного обеспечения 5](#_Toc107235168)

[2.2. Состав технических средств 5](#_Toc107235169)

[2.3. Структура локальной сети предприятия 9](#_Toc107235170)

[3. Выполнение индивидуального задания 10](#_Toc107235171)

[3.1. Разработка технического задания 10](#_Toc107235172)

[3.2. Проектирование пользовательского интерфейса 13](#_Toc107235173)

[3.3. Блок-схема программы 15](#_Toc107235174)

[3.4. Руководство программиста 16](#_Toc107235175)

[3.5. Руководство пользователя 18](#_Toc107235176)

[3.6. Отладка ПО 21](#_Toc107235177)

[3.7. Методика тестирования и испытания задачи 21](#_Toc107235178)

[3.8. Текст программы 22](#_Toc107235179)

[3.9 Предложения по улучшению 22](#_Toc107235180)

[Заключение 23](#_Toc107235181)

[Список использованной литературы 23](#_Toc107235182)

[Приложение 24](#_Toc107235183)

# Общие сведения о предприятии

Группа компаний «Стекломаш» (г. Орехово-Зуево, Московская область) — это предприятие с богатой историей, которое ориентировано на производство широкого спектра машиностроительной продукции, изделий строительного комплекса и технологического оборудования.

Конструкторско-технологический департамент позволяет оперативно реагировать на потребности современного рынка. Производить в кратчайшие сроки качественную и уникальную продукцию.

Производственные возможности обеспечивают выпуск продукции, отвечающей современным стандартам качества. Предприятие сертифицировано по международному стандарту железнодорожной отрасли IRIS. Производство продукции организовано согласно системе менеджмента качества и соответствует ГОСТ ISO 9001-2011 (ISO 9001:2008).

## 1.1. Структура организации



Рис. 1 «Структура организации»

# 2. Анализ материально-технической базы

## 2.1. Состав программного обеспечения

**Microsoft Office - о**фисный пакет приложений, созданных корпорацией Microsoft для операционных систем Windows, Windows Phone, Android, macOS, iOS. В состав этого пакета входит программное обеспечение для работы с различными типами документов: текстами, электронными таблицами, базами данных и др. Microsoft Office является сервером OLE-объектов и его функции могут использоваться другими приложениями, а также самими приложениями Microsoft Office. Поддерживает скрипты и макрокоманды, написанные на VBA.

**Windows 7** - пользовательская операционная система семейства Windows NT компании Microsoft. Следует по времени выхода за Windows Vista и предшествует Windows 8. Расширенная поддержка прекращена 14 января 2020 года. Поддержка для ОС на сегодняшний день платная, действует до 10 января 2023 года. Это последняя версия ОС Windows, использующая скевоморфический дизайн интерфейса, все последующие, вплоть до Windows 11, используют плоский дизайн в стиле Metro.

**Windows 10** - операционная система для персональных компьютеров и рабочих станций, разработанная корпорацией Microsoft в рамках семейства Windows NT. После Windows 8.1 система получила номер 10, минуя 9. Серверные аналоги Windows 10 - Windows Server 2016 и Windows Server 2019.

**Windows 11** - проприетарная операционная система для персональных компьютеров, разработанная компанией Microsoft в рамках семейства Windows NT, чтобы стать преемницей Windows 10. Она была представлена на мероприятии Microsoft 24 июня 2021 года. Выпущена 5 октября 2021 года, вместе с Microsoft Office 2021.

**Антивирус Касперского** - антивирусное программное обеспечение, разрабатываемое Лабораторией Касперского. Предоставляет пользователю защиту от вирусов, троянских программ, шпионских программ, руткитов, adware, а также от неизвестных угроз с помощью проактивной защиты, включающей компонент HIPS. Первоначально, в начале 1990-х, именовался -V, затем - AntiViral Toolkit Pro.

**SQL Server Management Studio (SSMS)** - это интегрированная среда для управления любой инфраструктурой SQL. Используйте SSMS для доступа, настройки, администрирования, администрирования и разработки всех компонентов SQL Server, [базы данных Azure SQL](https://learn.microsoft.com/ru-ru/azure/azure-sql/database/sql-database-paas-overview), [Управляемый экземпляр SQL Azure](https://learn.microsoft.com/ru-ru/azure/azure-sql/managed-instance/sql-managed-instance-paas-overview), [SQL Server на виртуальной машине Azure](https://learn.microsoft.com/ru-ru/azure/azure-sql/virtual-machines/windows/sql-server-on-azure-vm-iaas-what-is-overview) и [Azure Synapse Analytics](https://learn.microsoft.com/ru-ru/azure/synapse-analytics/sql-data-warehouse/sql-data-warehouse-overview-what-is/). Среда SSMS предоставляет единую комплексную служебную программу, которая сочетает в себе обширную группу графических инструментов с рядом многофункциональных редакторов скриптов для доступа к SQL Server для разработчиков и администраторов баз данных всех профессиональных уровней.

**Google Chrome** - браузер, разрабатываемый компанией Google на основе свободного браузера Chromium и движка Blink.

## 2.2. Состав технических средств

**МФУ лазерный HP LaserJet Pro RU M428dw, A4, лазерный**



Рис. 2 «МФУ, установленное на предприятии»

Таблица №1 «Характеристики МФУ»

|  |  |
| --- | --- |
| Технология печати | Лазерный |
| Тип печати | Чёрно-белый |
| Формат печати | А4 |
| Размещение | Настольный |
| Встроенный ЖК-дисплей | Цветной |
| ЖК-дисплей | Сенсорный |
| Диагональ дисплея | 2.7’’ |
| Сканер | Есть |
| Копировальный аппарат | Есть |

**Плоттер HP Designjet Z6 PostScript, 24"**

****

Рис. 3 «Плоттер»

Таблица №2 «Характеристики плоттера»

|  |  |
| --- | --- |
| Формат | А1 |
| Технология печати | Струйная |
| Количество цветов картриджей | 6 |
| Макс. разрешение печати | 2400х1200 dpi |
| Макс. ширина материала | 610 мм |
| Макс. плотность печатных носителей | 500 г/м2 |
| Способ подачи | Полистовая/рулонная |
| Автоматический резак | Есть |
| Память | 3.9 ГБ |
| Ёмкость жёсткого диска | 500 ГБ |
| Порт Ethernet | Gigabit Ethernet |
| Разъём USB | Есть |
| Размеры | 1293х695х998 мм |

**Сервер Dell PowerEdge T440 2x4114 2x16Gb 2RRD x16**

****

Рис. 4 «Сервер предприятия»

Таблица №3 «Характеристики сервера»

|  |  |
| --- | --- |
| Процессор, кэш | 14 МВ |
| Процессор, кол-во ядер | 10 |
| Установлено процессоров | 2 |
| Частота установленного процессора | 2.2 ГГц |
| Тип памяти | DDR4 |
| Кол-во слотов памяти | 12 |
| Макс. объём памяти | 384 ГБ |
| Установленные модули памяти | 2 х 16 ГБ |
| ОЗУ, доп.параметр1 | 2R |
| ОЗУ, доп.параметр2 | RD |
| ОЗУ, частота | 2.6 ГГц |

**Коммутатор D-Link DES-1024D/G1A 24x100M**



Рис. 5 «Коммутатор предприятия»

Таблица №4 «Характеристики коммутатора»

|  |  |
| --- | --- |
| Порты 10-100Base-TX | 24 шт. |
| Таблица MAC-адресов | 8K |
| Особенности | Полный/полудуплексный режим для 10/100 Мбит/с  Полнодуплексный режим для Gigabit Ethernet  Автоопределение MDI/MDIX для каждого порта  Металлический корпус |
| Размеры | 282x150x44мм |
| Вес | 1.14 кг |

**Модем 3G4G Alcatel Link Zone MW45V USB Wi-Fi Firewall Router**

****

Рис. 6 «Модем предприятия»

Таблица №5 «Характеристики модема»

|  |  |
| --- | --- |
| Интерфейс | USB |
| Скорость передачи данных | до 150 Мбит |
| Класс | 2G/3G/4G |
| Подключение к компьютеру | RJ-45 |
| Межсетевой экран (Firewall) | есть |
| Маршрутизатор | есть |

**Компьютер**



Рис. 7 «Компьютер предприятия»

Таблица №6 «Характеристики компьютера»

|  |  |
| --- | --- |
| Процессор | AMD Ryzen 5 5600G |
| Частота | 3.9 ГГц (4.4 ГГц, в режиме Turbo) |
| Кол-во ядер | 6 |
| Чипсет материнской платы | AMD B550A |
| Оперативная память | 8 ГБ, DDR4, DIMM, 3200 МГц |
| Графика | NVIDIA GeForce RTX 3060 - 12288 Мб |
| Объём SSD | 512 ГБ |
| Тип корпуса | tower |

## 2.3. Структура локальной сети предприятия

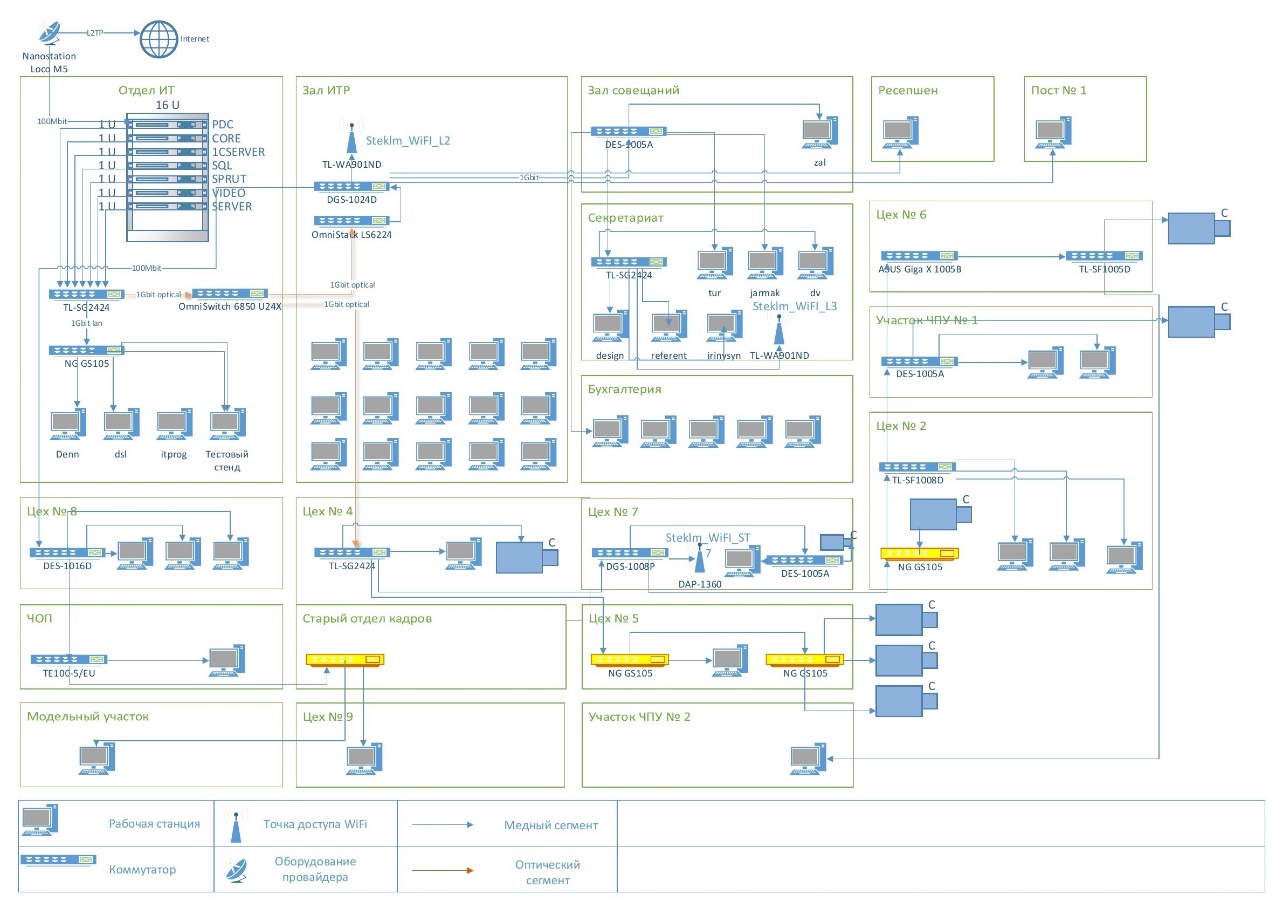


Рис. 8 «Структура локальной сети»

# 3. Выполнение индивидуального задания

## 3.1. Разработка технического задания

**Назначение разработки:**

Программа предназначена для упрощения и автоматизации работы с картами контроля, вывод их на печать, а также для внесения ответов сотрудниками.

Для занесения данных о проекте работают сотрудники различных должностей. При поступлении сотрудника на работу все данные регистрируются.

**Требования к программе или программному изделию**

**Функциональные требования:**

Реализовать программу для занесения данных в карту контроля в базу данных и дальнейшей работы с этими данными.

Функциональные требования:

— возможность создавать/просматривать/редактировать/удалять информацию о картах контроля;

— возможность создавать/просматривать/редактировать/удалять информацию о шаблонах;

— возможность создавать/просматривать/редактировать/удалять информацию о разделах;

— возможность создавать/просматривать/редактировать/удалять информацию о пунктах;

— для просмотра таблиц предусмотреть различные способы сортировки (по основным полям).

Необходимая для хранения информация:

— название шаблона;

— название детали;

— серийный номер;

— заводской номер;

— даты приёмки;

— логин пользователя;

— пароль пользователя.

Добавить сущность «Ответы», используя Entity Framework Code First DB Migrations.

Функциональные требования:

— возможность создавать/просматривать/редактировать/удалять информацию о ответах;

— возможность добавлять и удалять ответы из карты контроля (одна карта контроля может содержать только 2 ответа);

— для просмотра ответов предусмотреть различные сортировки (по основным полям).

Необходимая для хранения информация:

— название ответа,

— должность пользователя (связь с сущностью «Авторизация»),

— пункт ответа (связь с сущностью «Пункт»).

**Требования к организации входных данных:**

Организованы в виде ввода в текстовые поля или выбора в выпадающем списке

**Требования к организации выходных данных:**

Организованы в виде таблиц, размещённых на формах

**Требования к надёжности:**

Не должна быть утеряна база данных в случае перебоя электросети

**Требования к составу и параметрам технических средств:**

Таблица №7 «Минимальные системные требования»

|  |  |
| --- | --- |
| Процессор | Intel® Core™ i5-6400 CPU @ 2.70GHz |
| Оперативная Память | 2 x 4 ГБ, DDR4, DIMM, 2133 МГц |
| Разрешение экрана | От 1280 × 1024 |
| Устройства ввода | Клавиатура, мышь |
| Дисковое пространство | От 2 Гб |
| Дополнительно | Постоянное подключение к сети, в которой находится база данных |

Таблица №8 «Рекомендуемые системные требования»

|  |  |
| --- | --- |
| Процессор | Intel Core i5 3 GHz |
| Память | От 8 Гб |
| Разрешение экрана | От 1440 × 900 |
| Устройства ввода | Клавиатура, мышь |
| Дисковое пространство | От 8 Гб |
| Дополнительно | Постоянное подключение к сети, в которой находится база данных |

**Требования к информационной и программной совместимости:**

Для корректной работы программы необходимо:

ОС Windows 8.1/10 – операционная система, сделанная корпорацией Microsoft;

Visual Studio 2022 – лучшая интегрированная среда разработки для создания многофункциональных, привлекательных кроссплатформенных приложений для Windows;

Microsoft SQL Server Management Studio 18 – интегрированная среда для управления любой инфраструктурой SQL, от SQL Server до баз данных;

Microsoft Excel 2016 – программа для работы с электронными таблицами, созданная корпорацией Microsoft для Microsoft Windows.

**Условия эксплуатации:**

Программа не требует специального обслуживания. Для ознакомления с полным функционалом пользователь должен прочесть Руководство пользователя. Для работы с программой не требуются каких-либо навыков.

**Климатические условия эксплуатации:**

Климатические условия эксплуатации, при которых должны обеспечиваться заданные характеристики, должны удовлетворять требованиям, предъявляемым к техническим средствам в части условий их эксплуатации.

**Специальные требования**

Программа должна обеспечивать взаимодействие с пользователем (преподавателя) посредством графического пользовательского интерфейса.

**Требования к программной документации**

В ходе разработки программы должны быть подготовлены следующие программные документы: текст программы, описание программы, программа и методика испытаний, руководство пользователя, руководство программиста, технико-экономическое обоснование

## 3.2. Проектирование пользовательского интерфейса

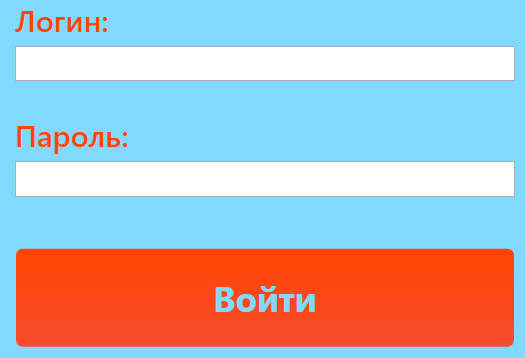


Рис. 19 «Окно авторизации»

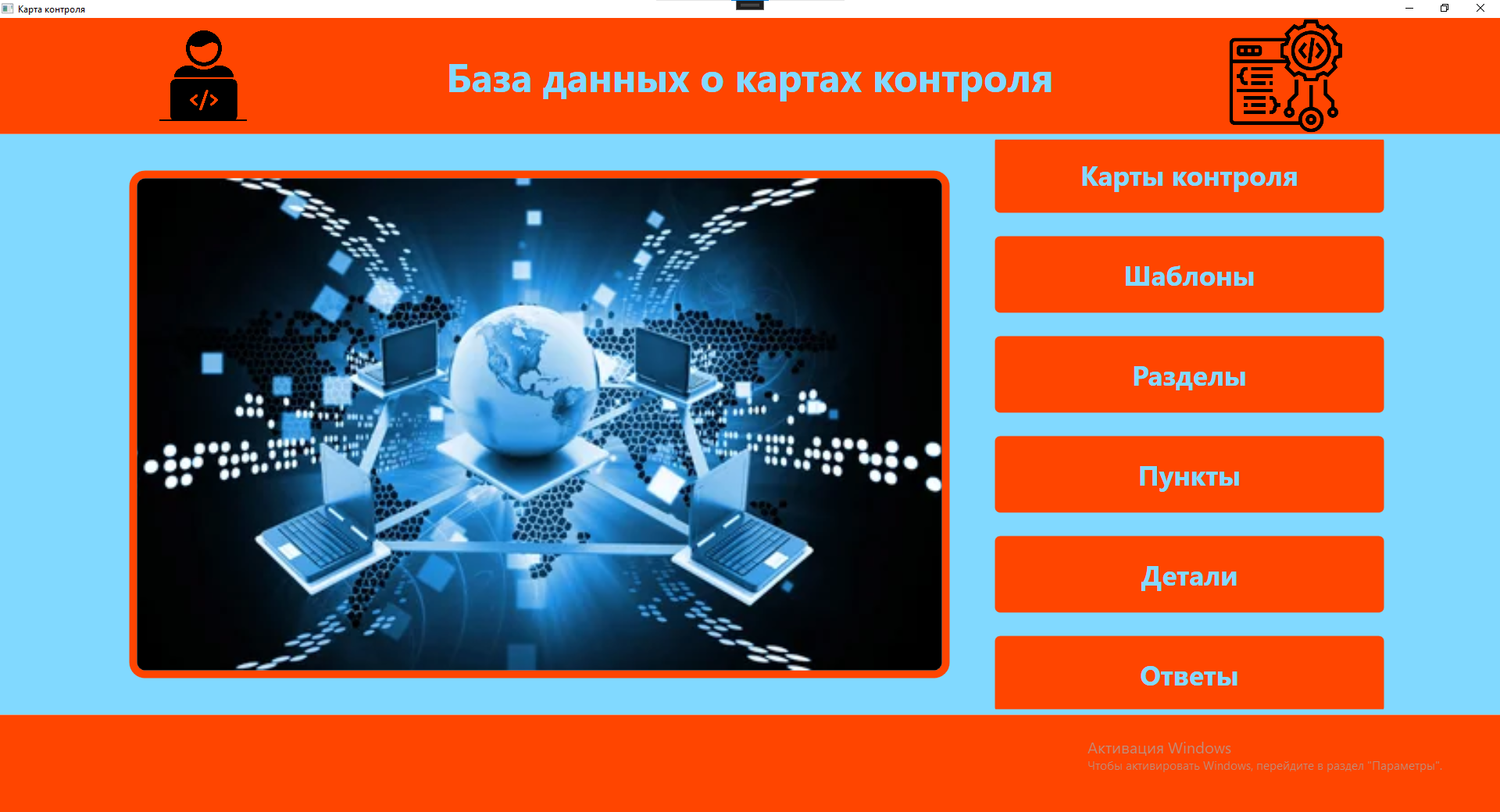


Рис. 20 «Главная страница»

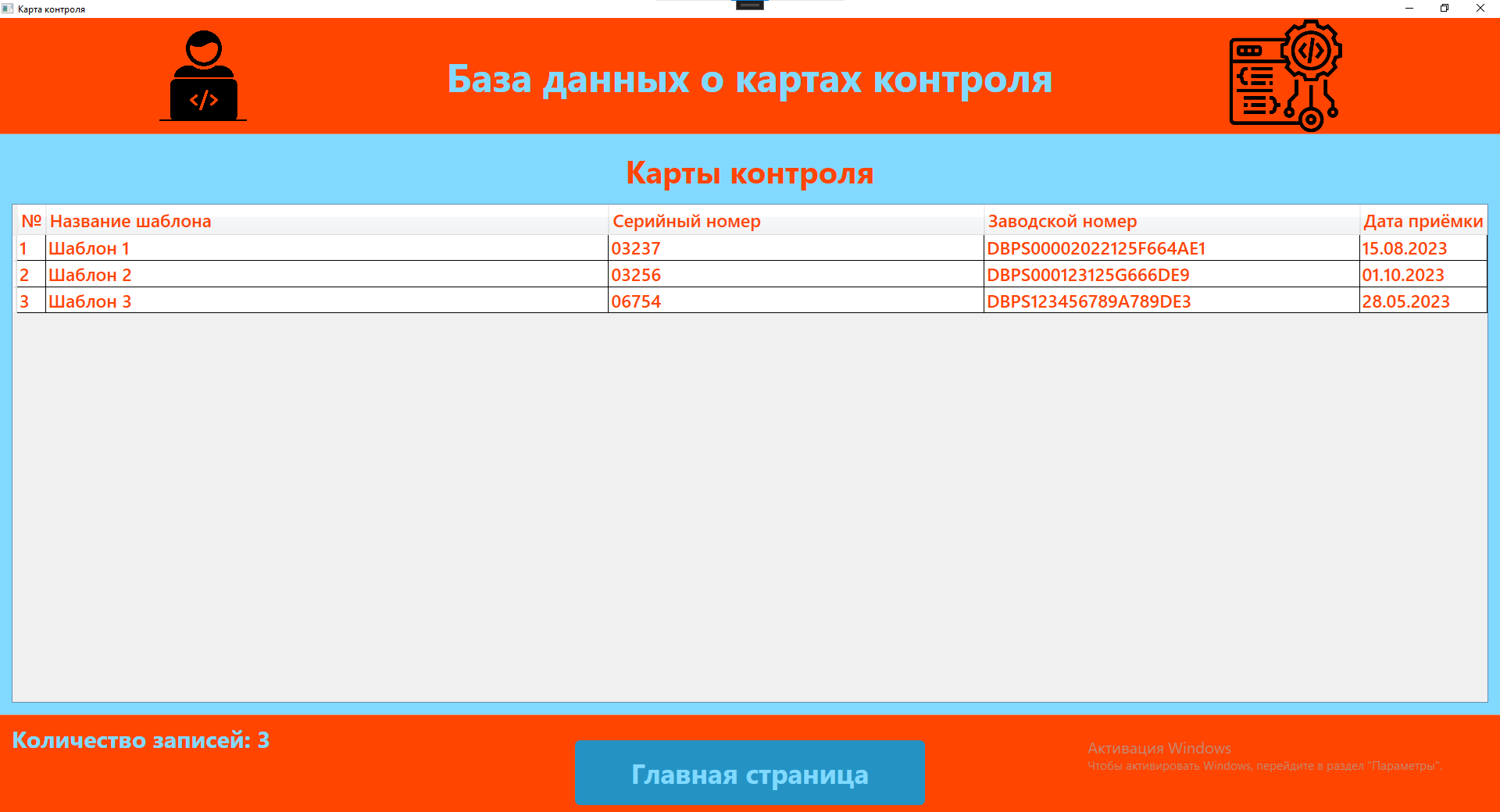


Рис. 21 «Страница карты контроля»

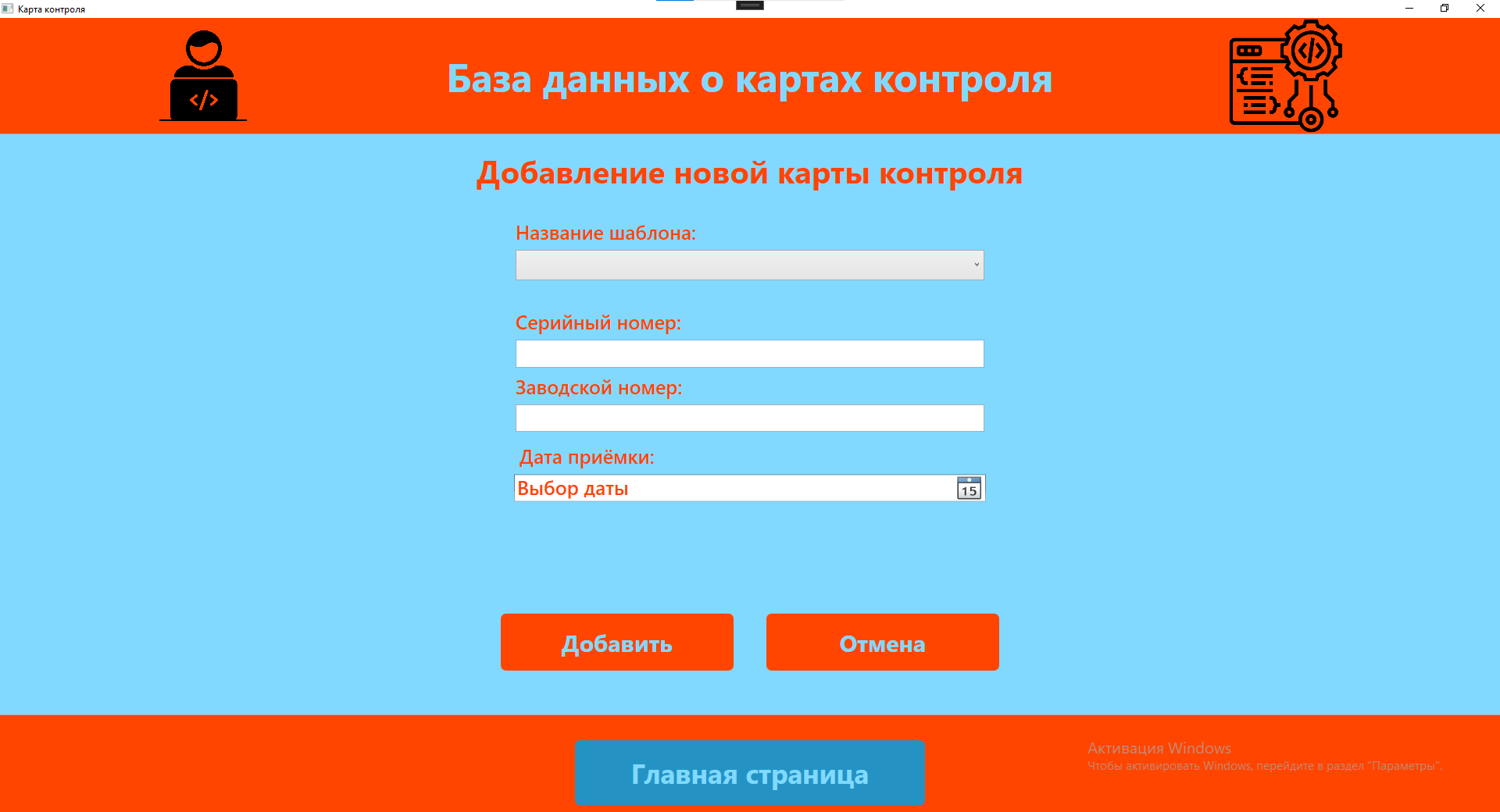


Рис. 22 «Страница добавления карты контроля»

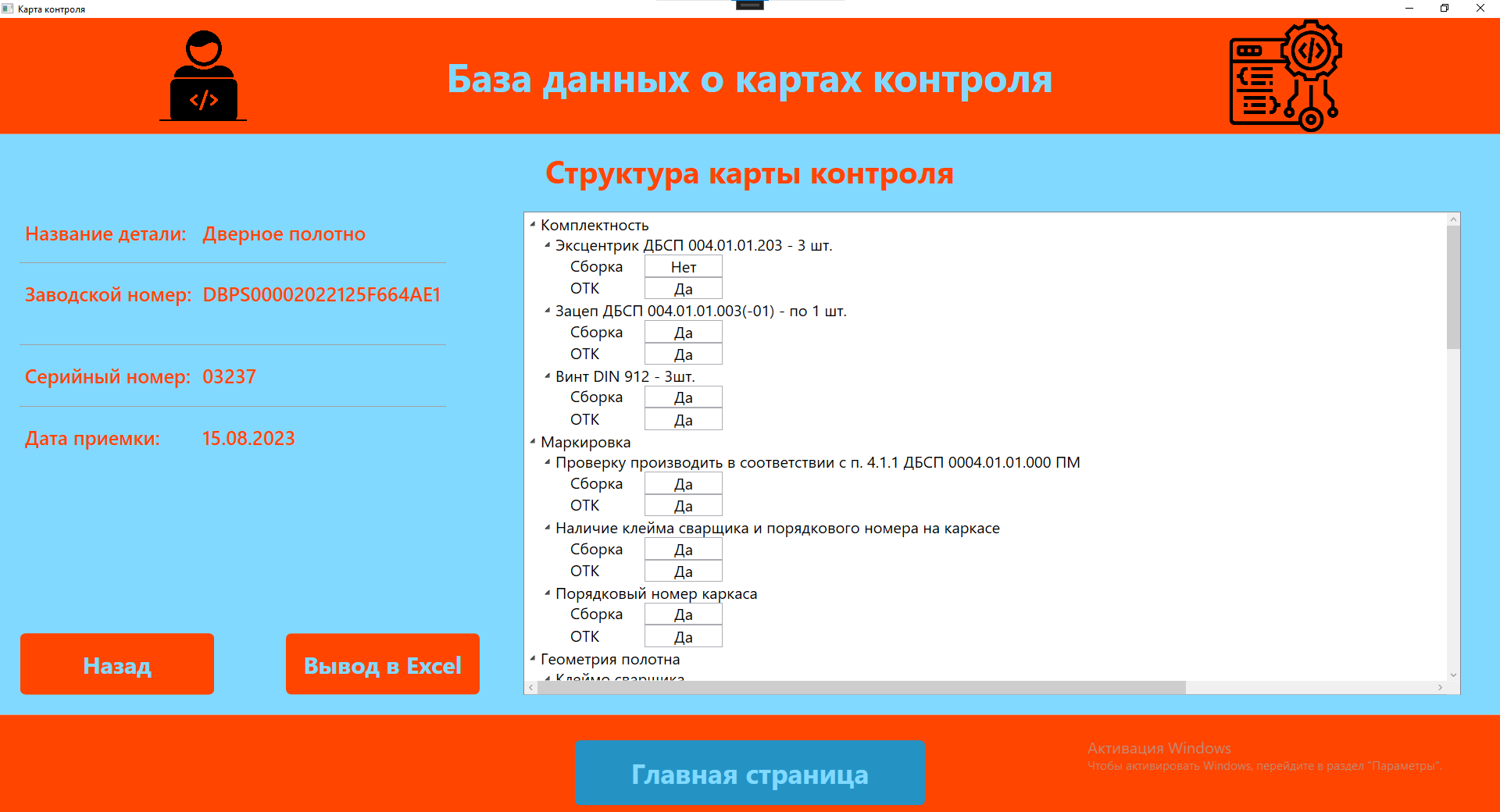


Рис. 23 «Страница структуры карты контроля»

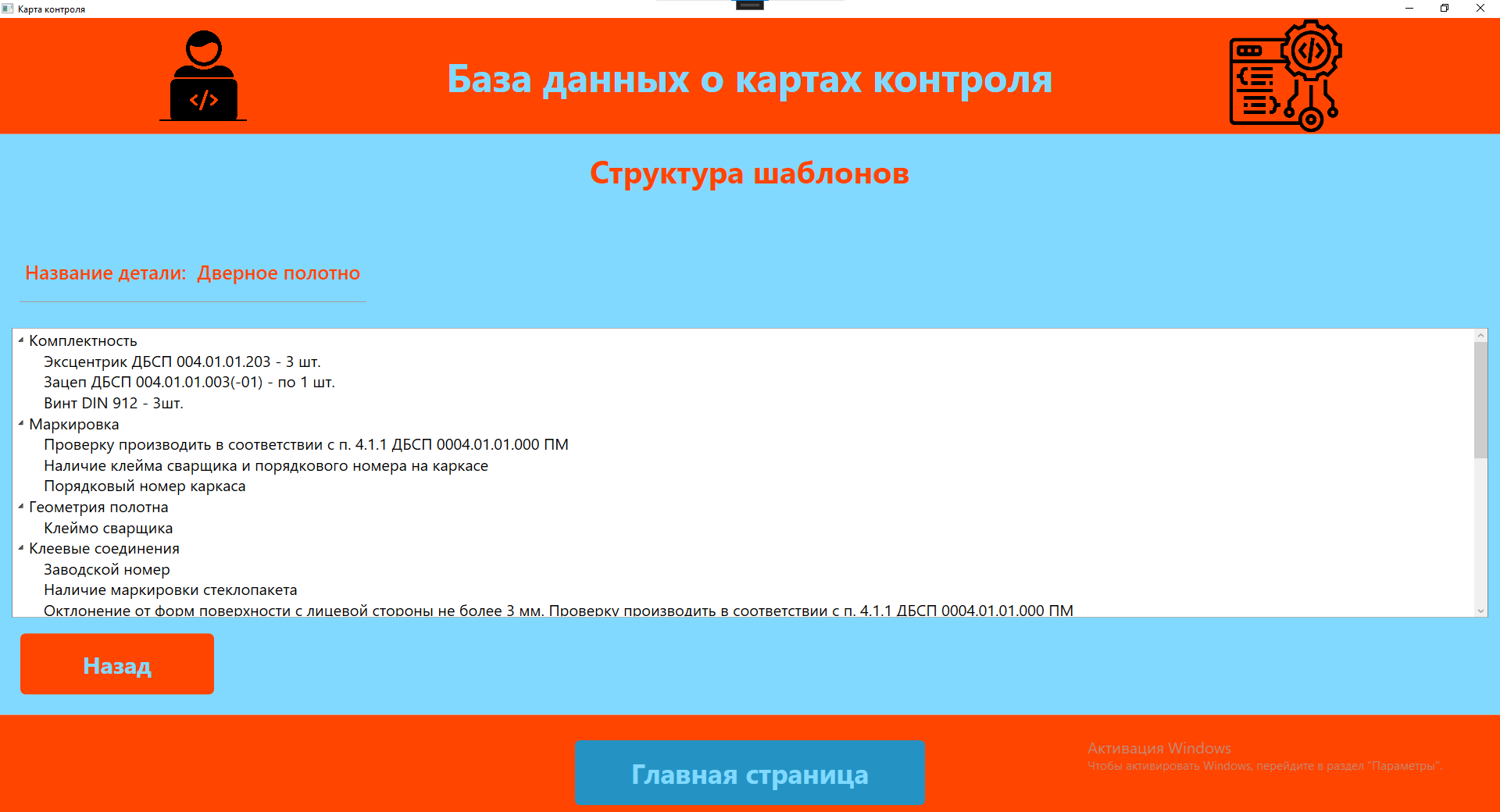


Рис. 24 «Страница структуры шаблона»

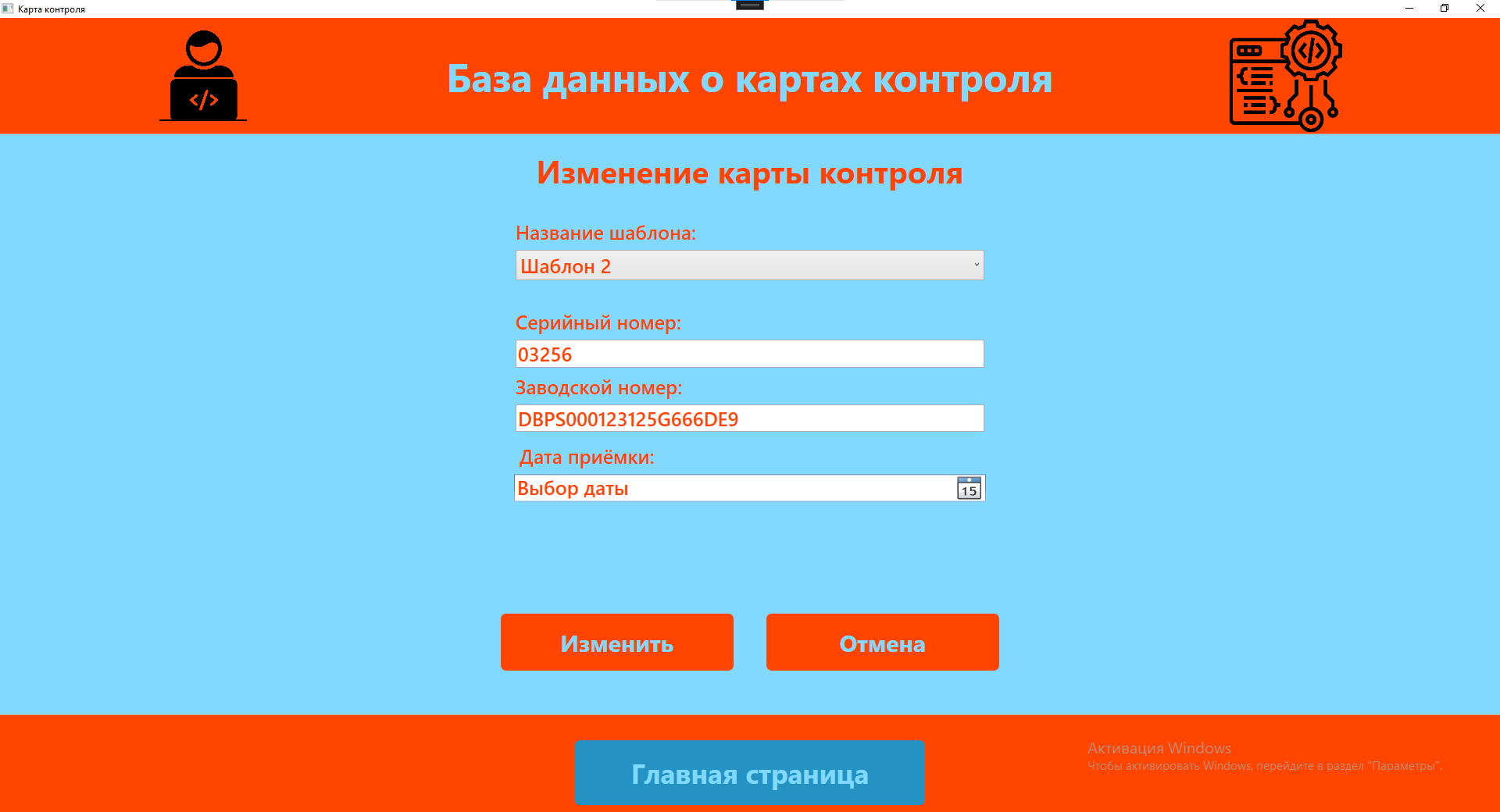


Рис. 25 «Окно редактирования карты контроля»

## 3.3. Блок-схема программы

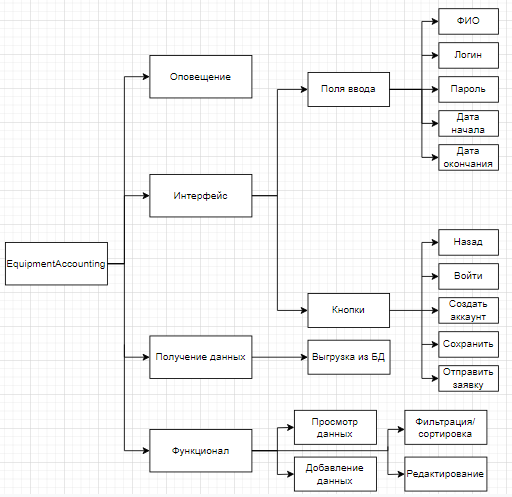


Рис. 26 «Функциональная схема программы»

## 3.4. Руководство программиста

Приложение разработано в Visual Studio 2019

Создание проекта:

|  |  |
| --- | --- |
|  |  |

Рис. 27 «Создание проекта»

Для корректной работы приложения необходимо подключить БД:

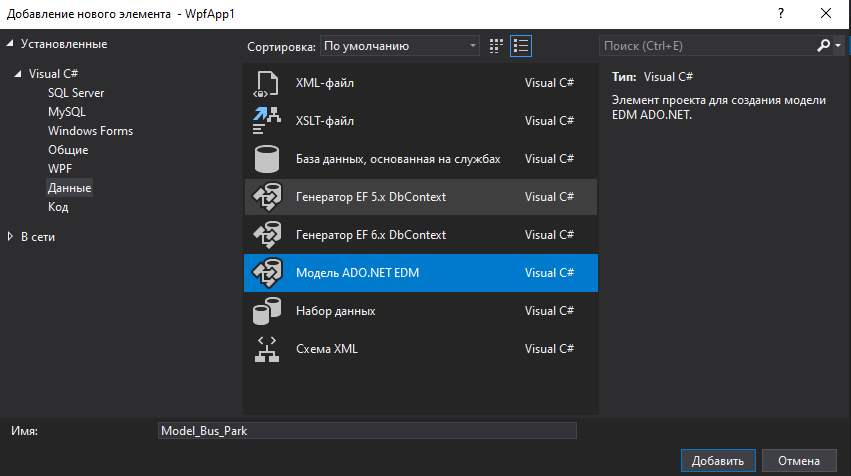


Рис. 29 «Создание модели ADO.NET»

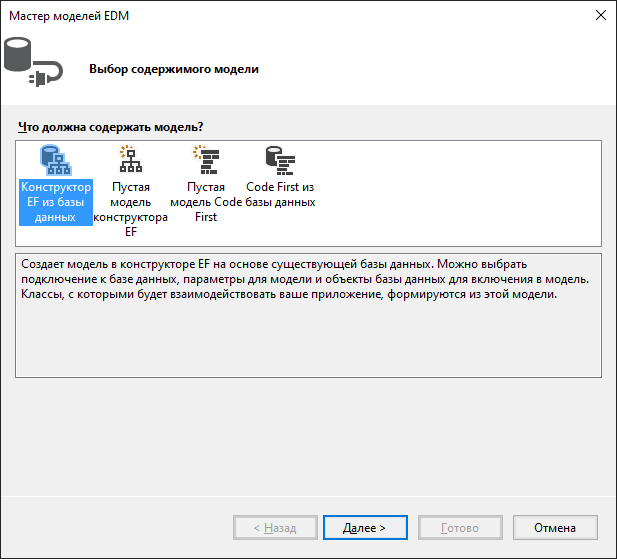


Рис. 30 «Выбор содержимого модели»

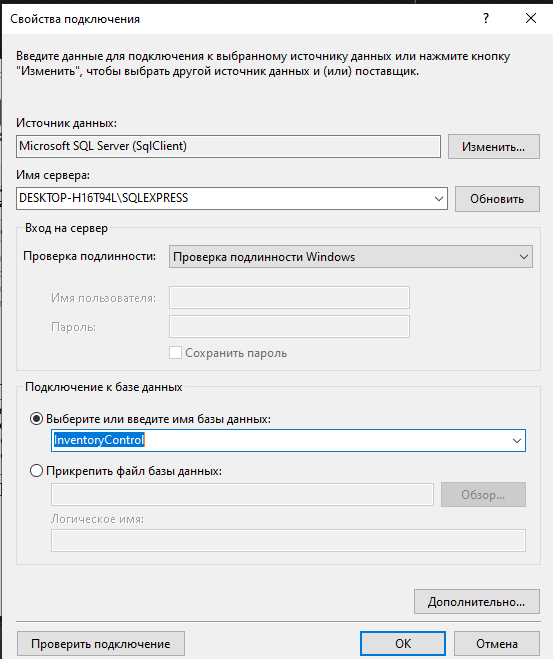


Рис. 31 «Выбор сервера и имени БД»

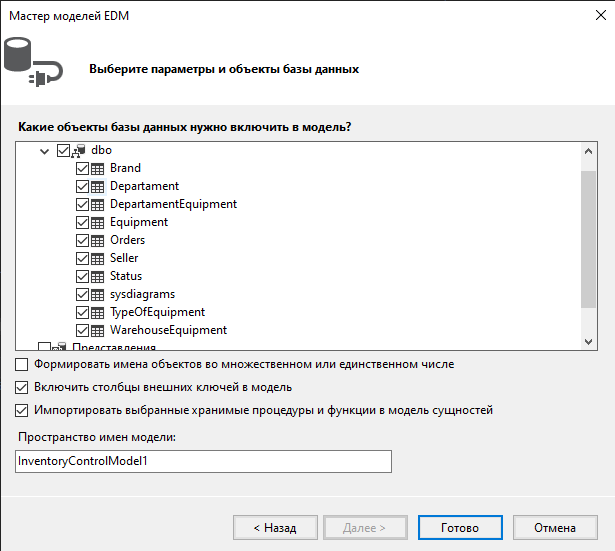


Рис. 32 «Выбор объектов БД»

Диаграмма БД в Visual Studio:

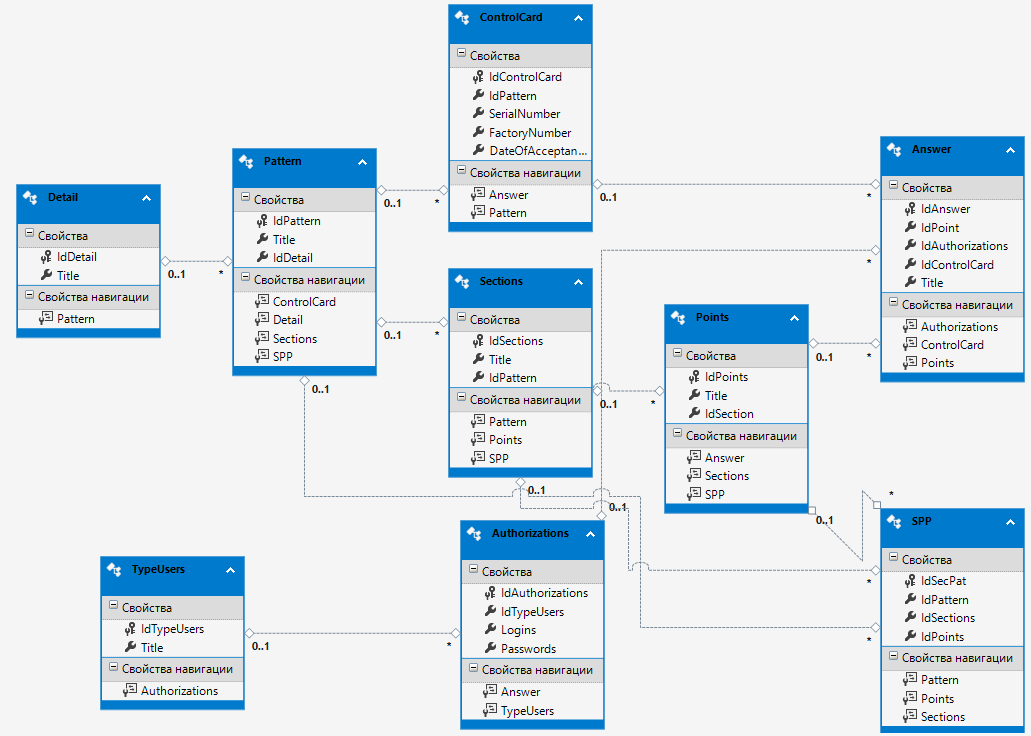


Рис. 33 «Модель данных»

Структура приложения в обозревателе решений:

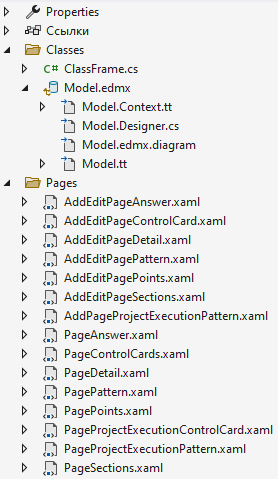


Рис. 34 «Структура приложения»

Класс «Frame», служащий для связи с базой данных:

namespace ControlCards.Classes

{

using System;

using System.Data.Entity;

using System.Data.Entity.Infrastructure;

public partial class ControlCardMalchikEntities : DbContext

{

public ControlCardMalchikEntities()

: base("name=ControlCardMalchikEntities")

{

}

private static ControlCardMalchikEntities \_context;

public static ControlCardMalchikEntities GetContext()

{

if (\_context == null)

\_context = new ControlCardMalchikEntities();

return \_context;

}

protected override void OnModelCreating(DbModelBuilder modelBuilder)

{

throw new UnintentionalCodeFirstException();

}

public virtual DbSet<Answer> Answer { get; set; }

public virtual DbSet<Authorizations> Authorizations { get; set; }

public virtual DbSet<ControlCard> ControlCard { get; set; }

public virtual DbSet<Detail> Detail { get; set; }

public virtual DbSet<Pattern> Pattern { get; set; }

public virtual DbSet<Points> Points { get; set; }

public virtual DbSet<Sections> Sections { get; set; }

public virtual DbSet<SPP> SPP { get; set; }

public virtual DbSet<sysdiagrams> sysdiagrams { get; set; }

public virtual DbSet<TypeUsers> TypeUsers { get; set; }

}

}

## 3.5. Руководство пользователя

При открытии программы ввести логин и пароль, привязанные к аккаунту. Исходя из роли в учётной записи (ОТК, сборка), осуществится переход на определённую страницу. ОТК не может заполнять поле для сборки, сборка не может заполнять поле для ОТК. В случае, если данные для авторизации введены неверно, появляется уведомление.

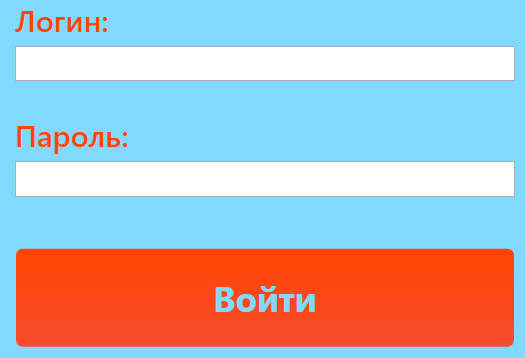


Рис. 1 «Страница авторизации»

После авторизации (исполняемый «.exe» файл) появится страница «Главная страница» с возможностью перехода на страницы «Карты контроля», «Шаблоны», «Разделы», «Пункты», «Детали» и «Ответы».

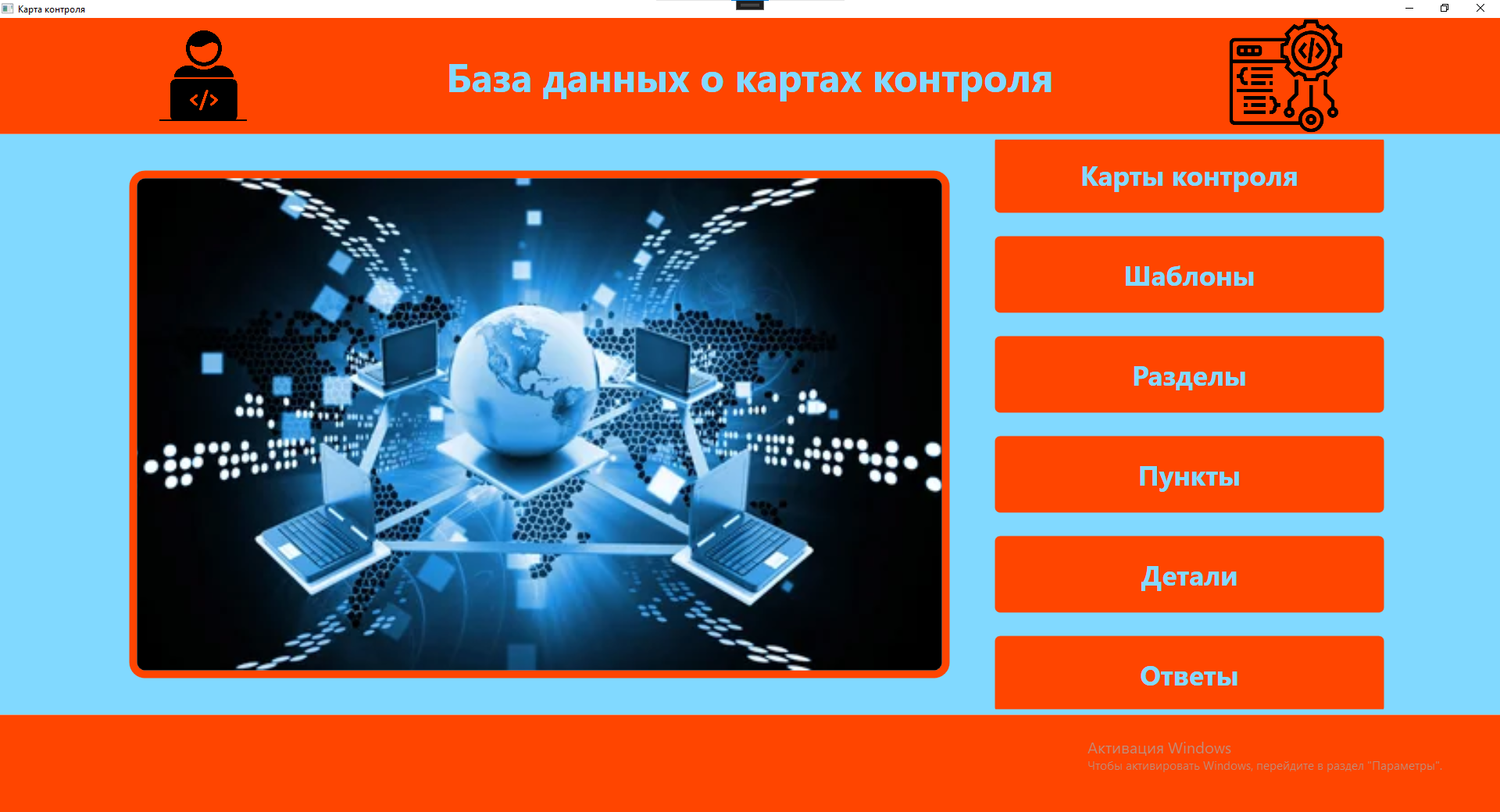


Рис. 2 «Главная страница»

При нажатии в навигационном меню на кнопку «Карты контроля» откроется страница с соответствующим списком карт контроля.

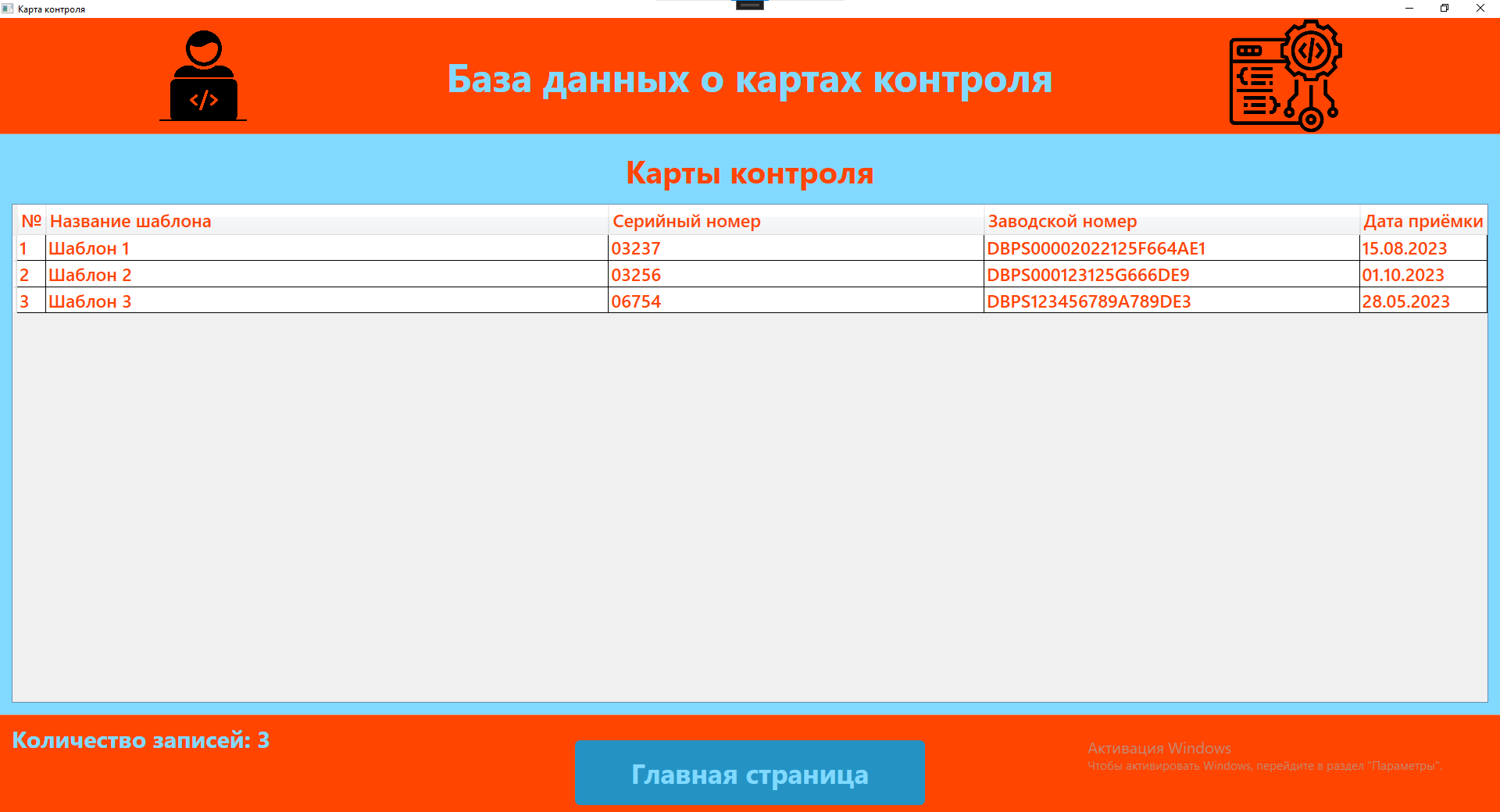


Рис. 3 «Страница карт контроля»

При добавлении информации пользователю необходимо нажать на таблицу правой кнопкой мыши и в контекстном меню выбрать элемент «Добавить». Затем появится страница для добавления с пустыми значениями, которые потребуется заполнить. Обязательные поля: название шаблона, серийный номер, заводской номер и дата приёмки.

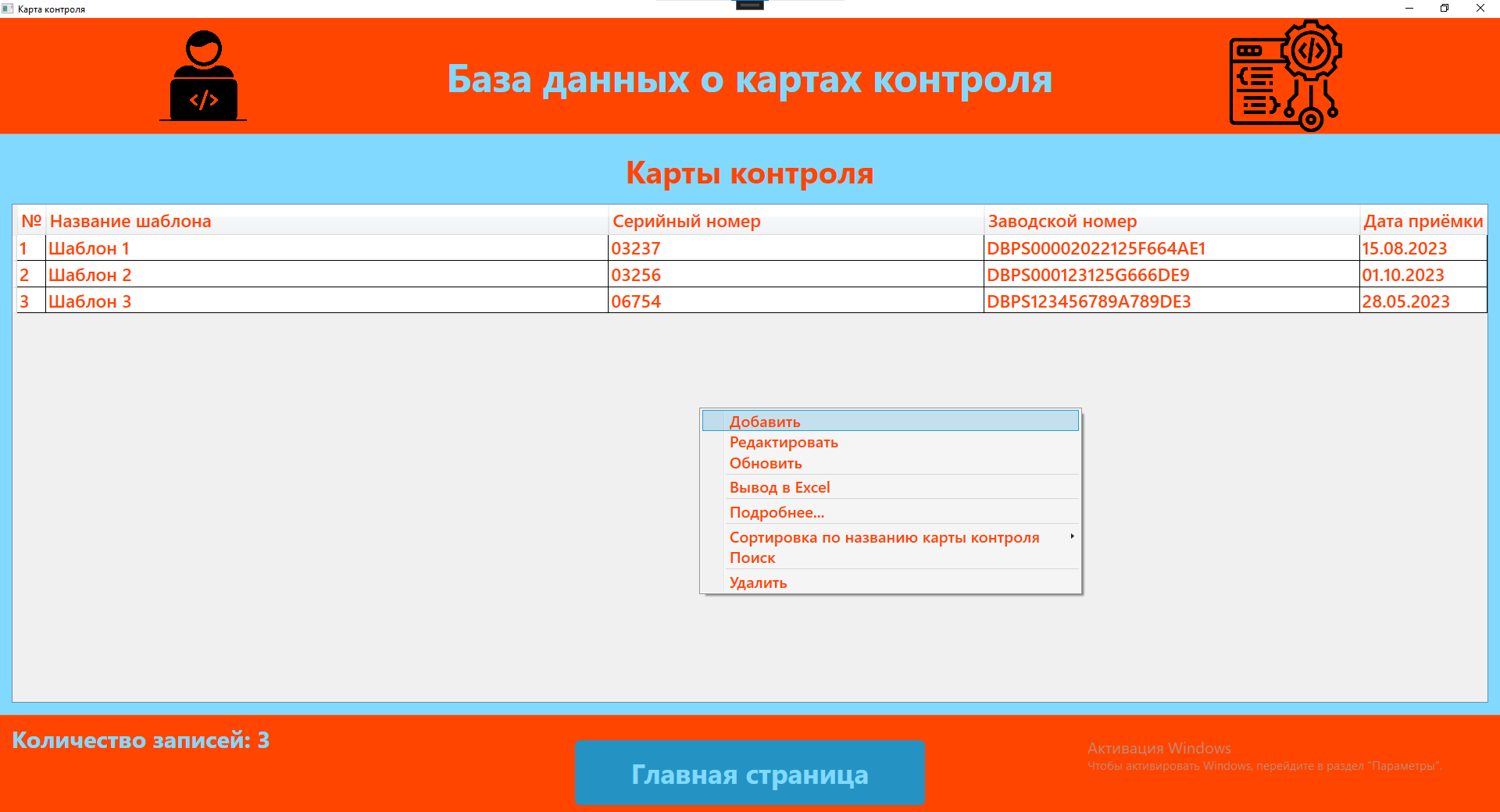


Рис. 4 «Страница карт контроля с вызовом добавления»

Заполним страницу добавления данными.

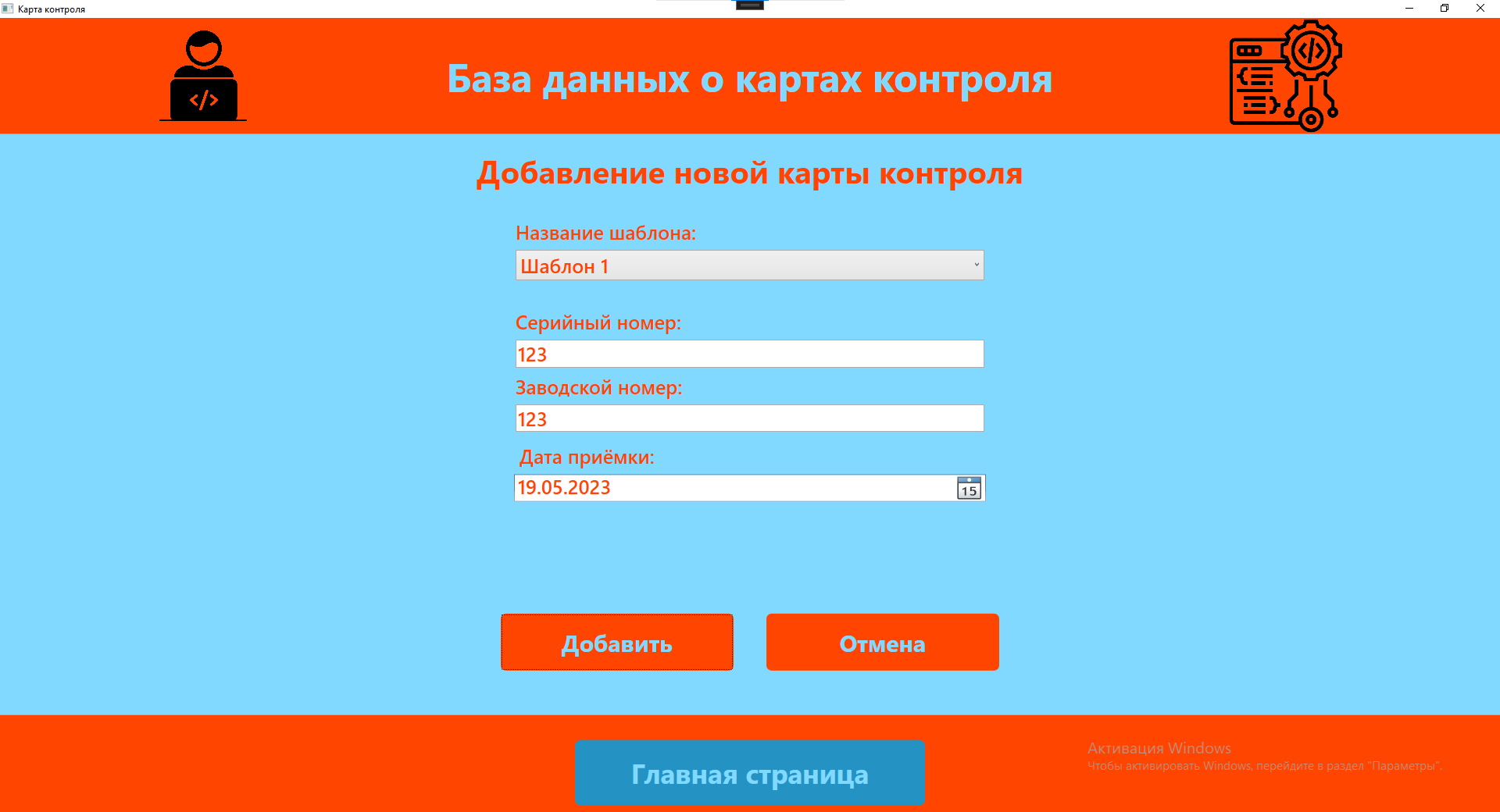


Рис. 5 «Страница добавления карты контроля»

После нажатия кнопки «Добавить» в списке появляется данная запись.

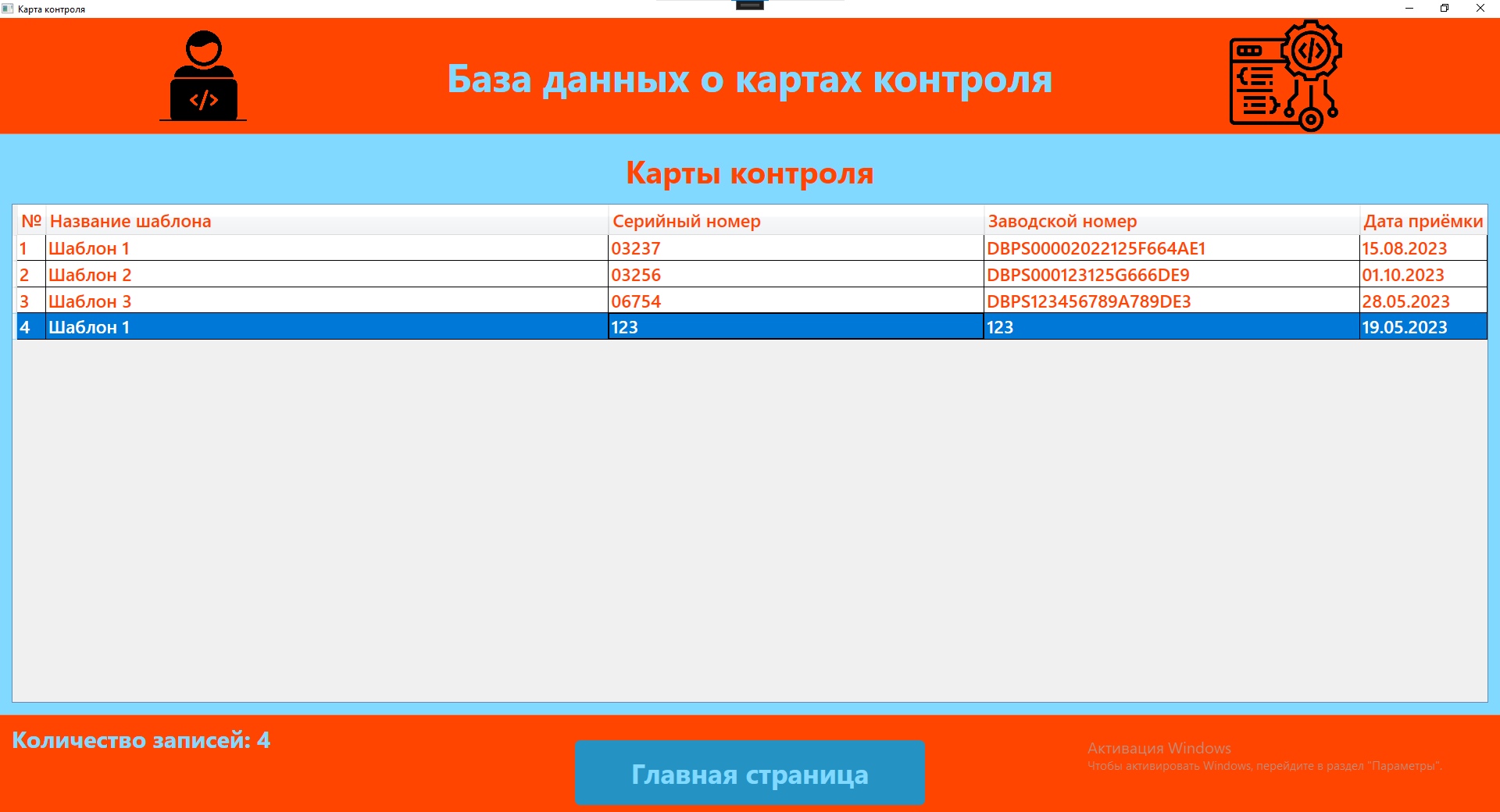


Рис. 6 «Страница карты контроля с добавленной записью»

Чтобы сортировать записи по атрибуту используются вложенное контекстное меню, при нажатии которого выводятся отсортированные данные в таблице, соответствующие условию.

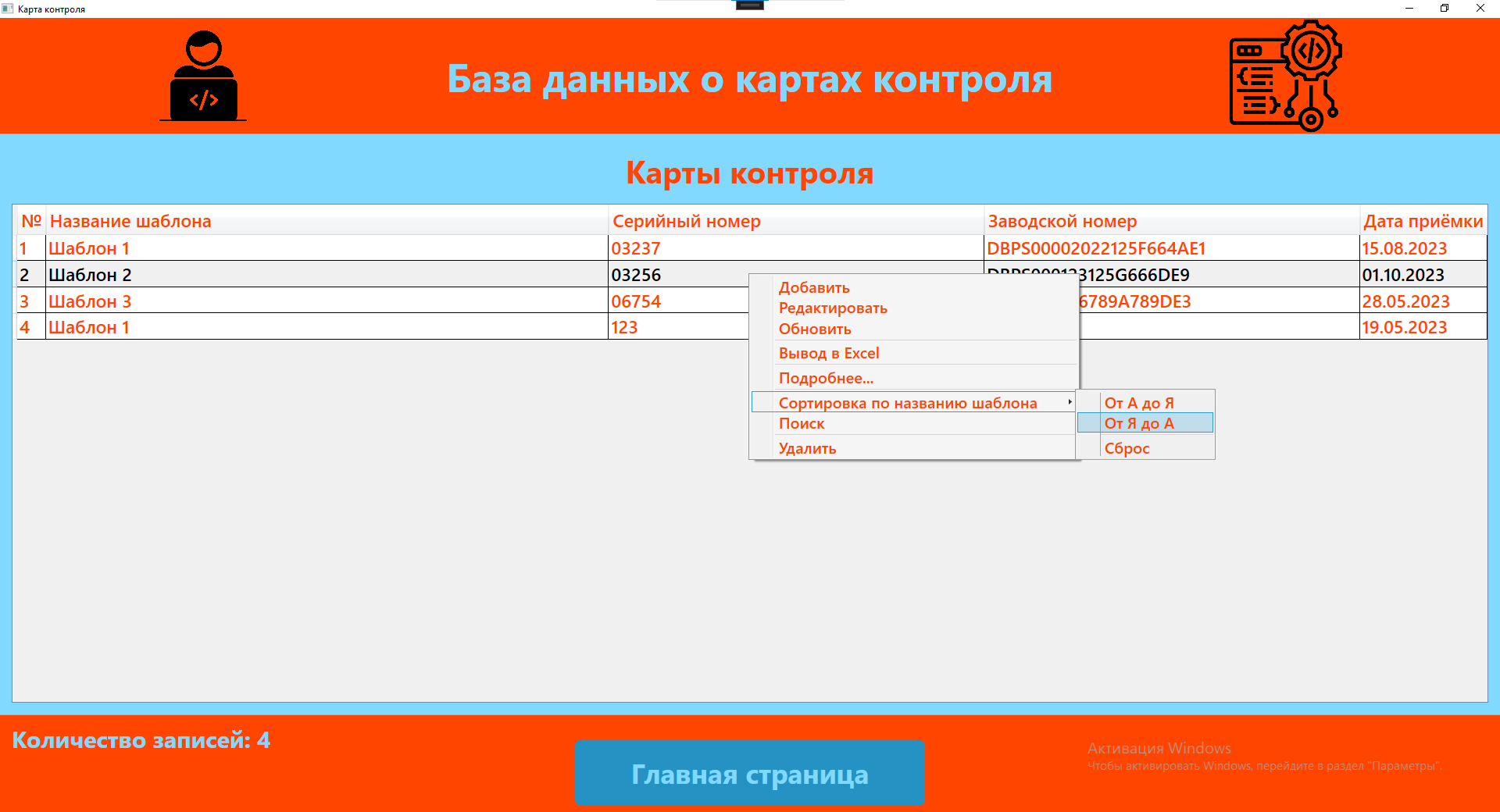


Рис. 7 «Страница карты контроля с вызовом сортировки»

После выбора пункта «От Я до А» из контекстного меню список сортируется по данному критерию.

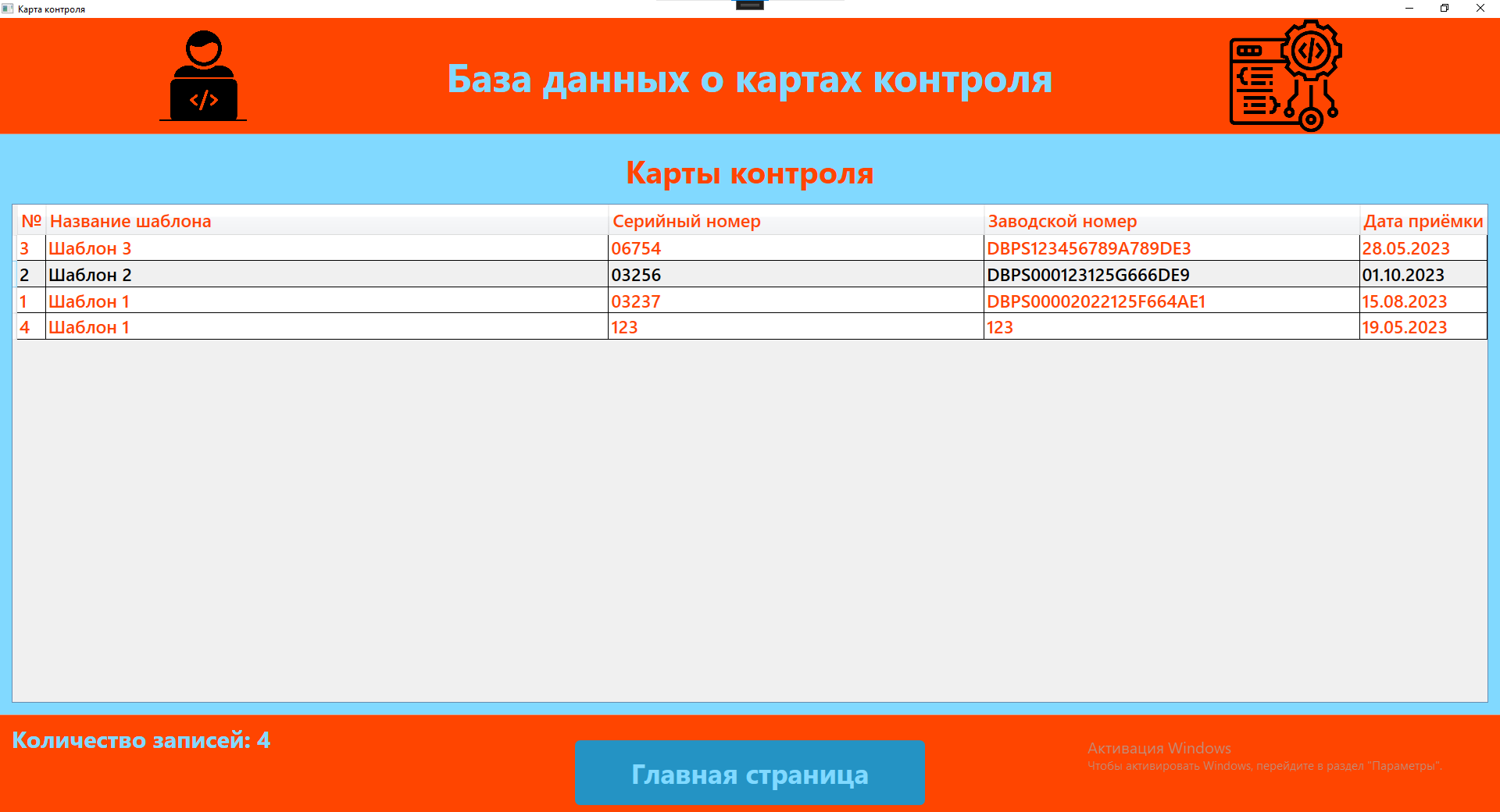


Рис. 8 «Страница карты контроля с сортировкой по названию от Я до А»

В приложении присутствует поиск для нахождения данных по атрибутам таблицы. Чтобы использовать поиск нужно в контекстном меню выбрать пункт «Поиск», после чего откроется поле для ввода. В него нужно ввести цифру, которые присутствуют в названии шаблона.

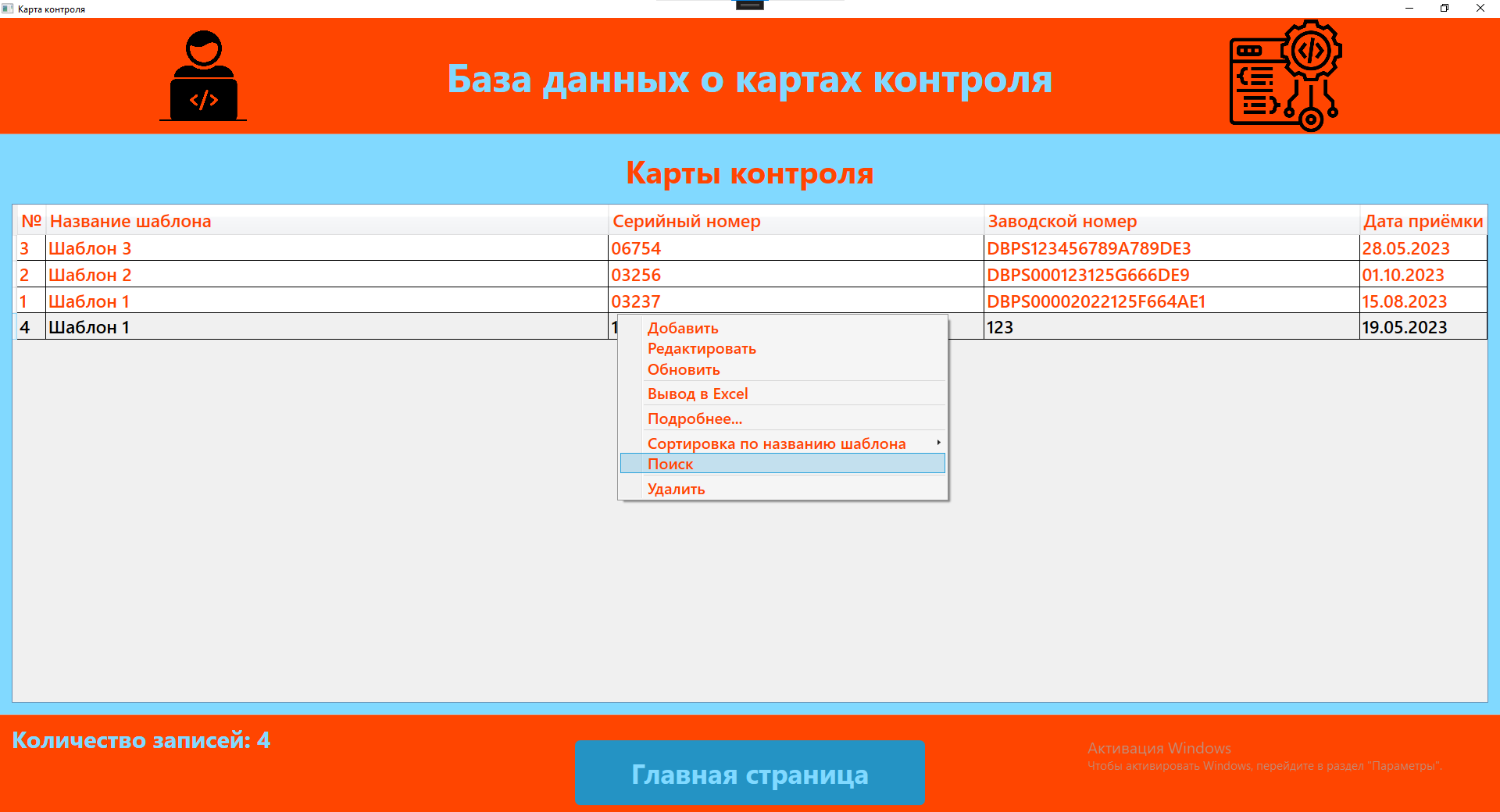


Рис. 9 «Страница карты контроля с вызовом поиска в контекстном меню»

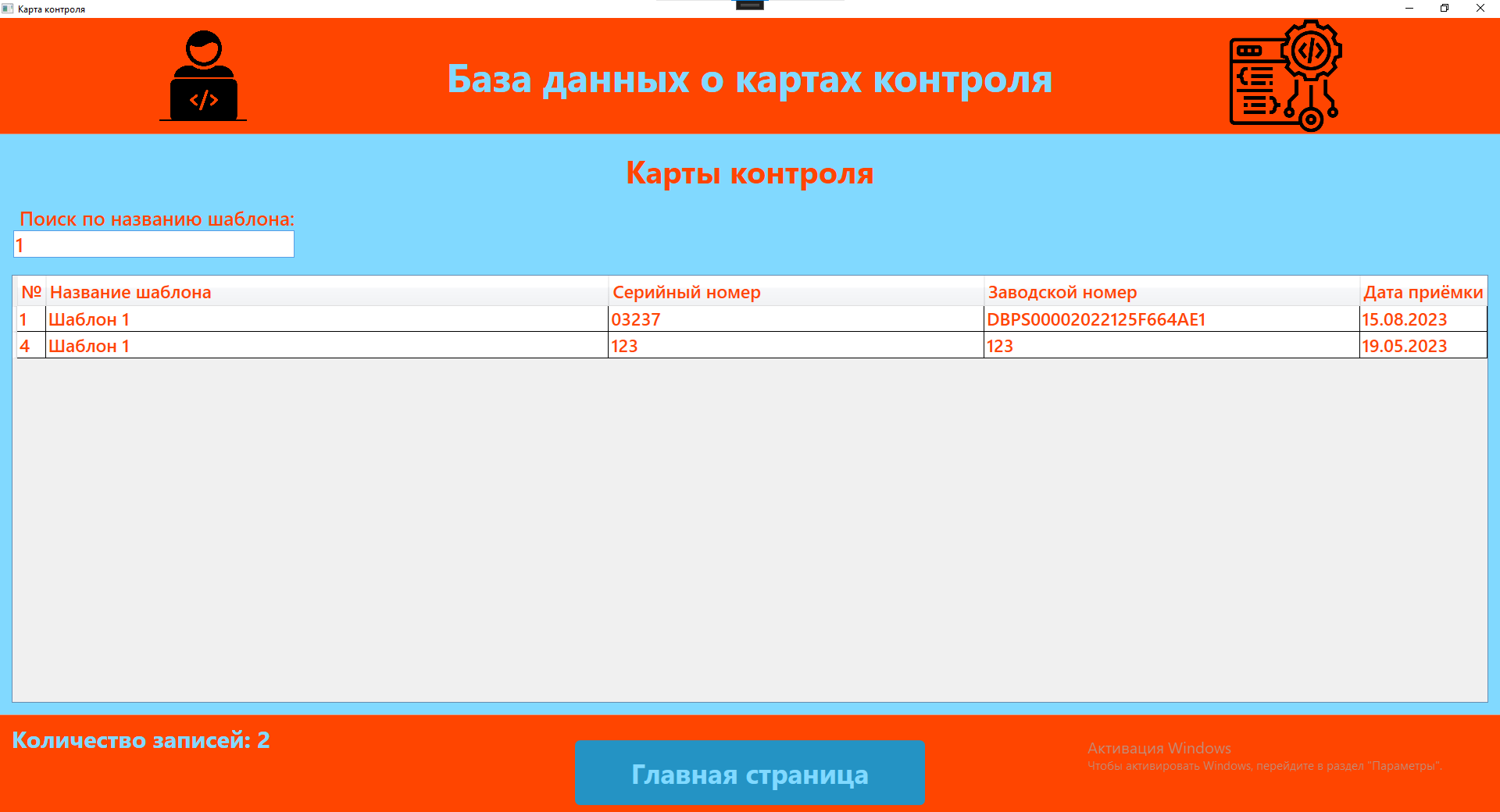


Рис. 10 «Страница карты контроля с поиском»

Чтобы убрать поиск, в контекстном меню нужно нажать пункт «Убрать поиск».

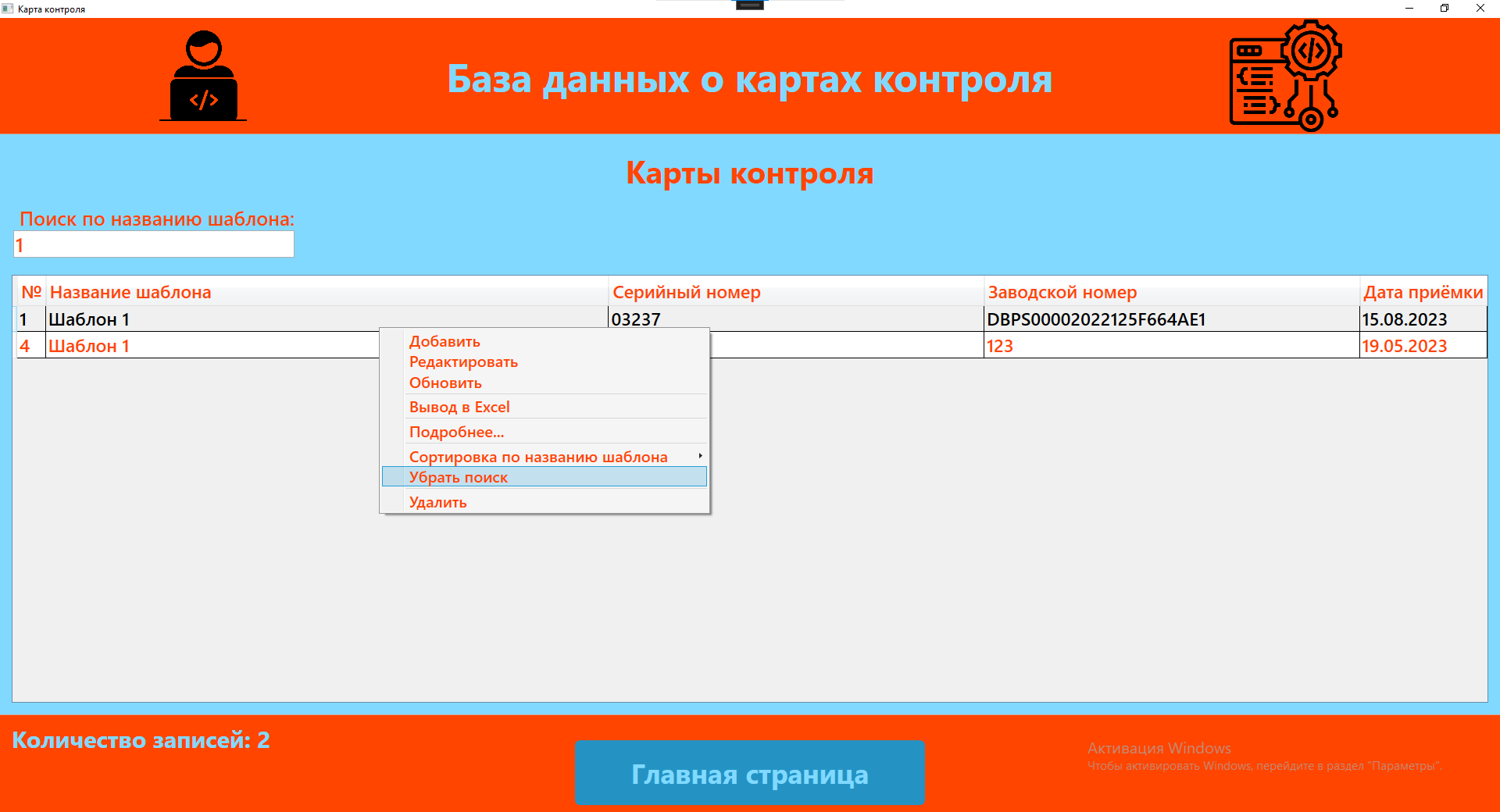


Рис. 11 «Страница карты контроля с вызовом скрытия поиска в контекстном меню»

Чтобы убрать результат операций, пользователю нужно нажать на элемент «Сброс» в контекстном меню в фильтрации или сортировке, или на элемент «Обновить».

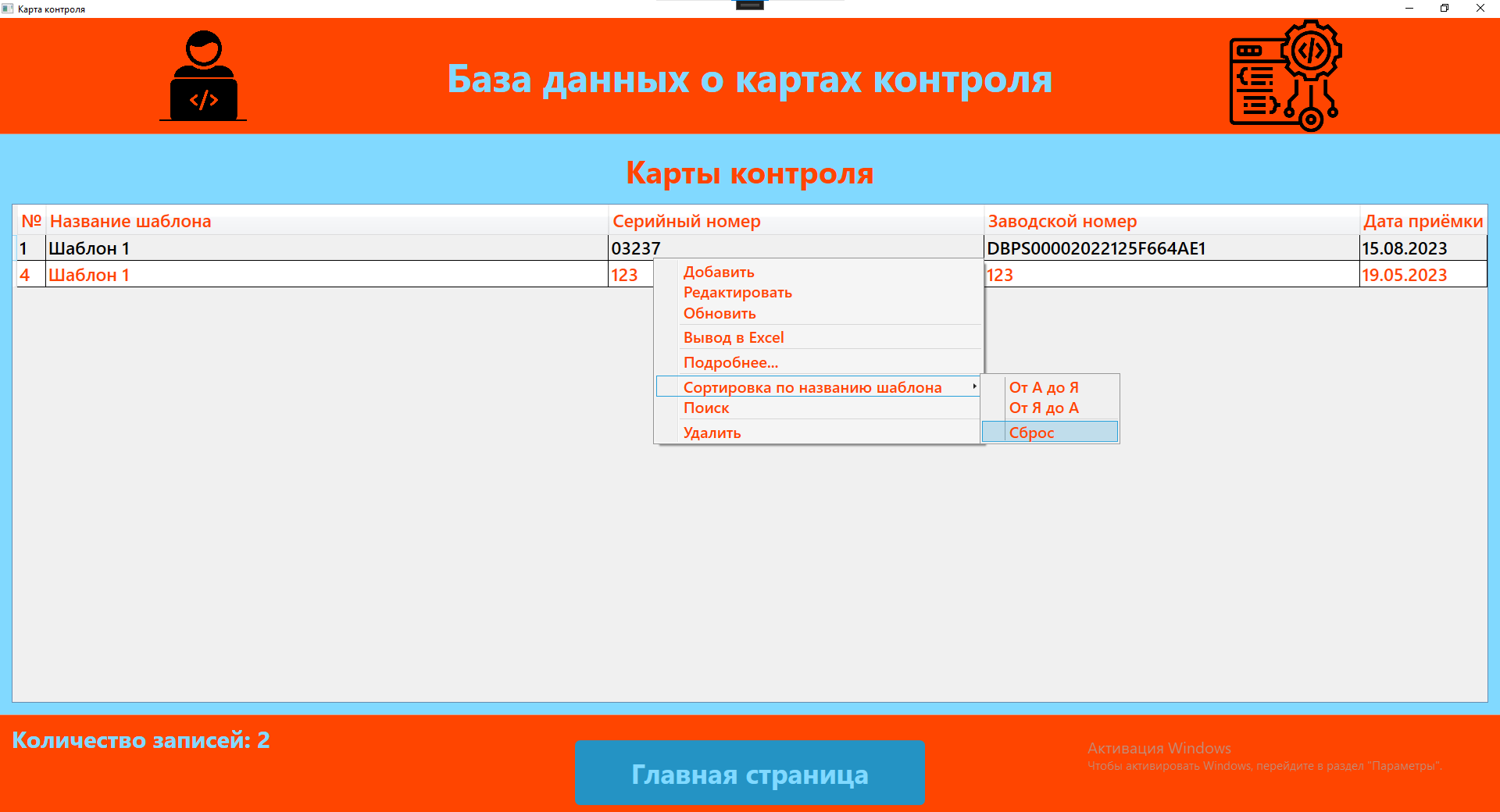


Рис. 12 «Страница карты контроля с вызовом сброса»

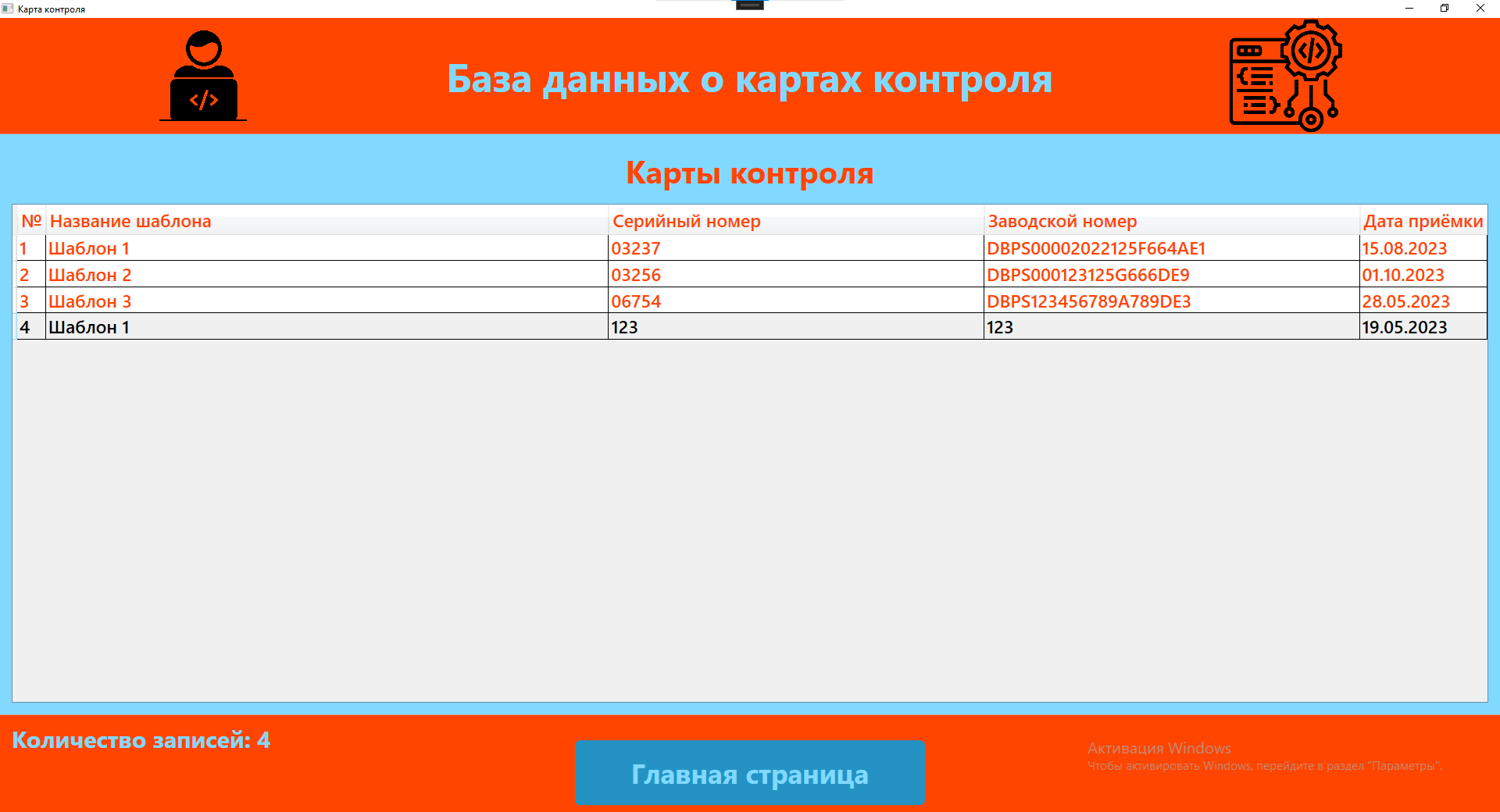


Рис. 13 «Страница карты контроля после вызова сброса»

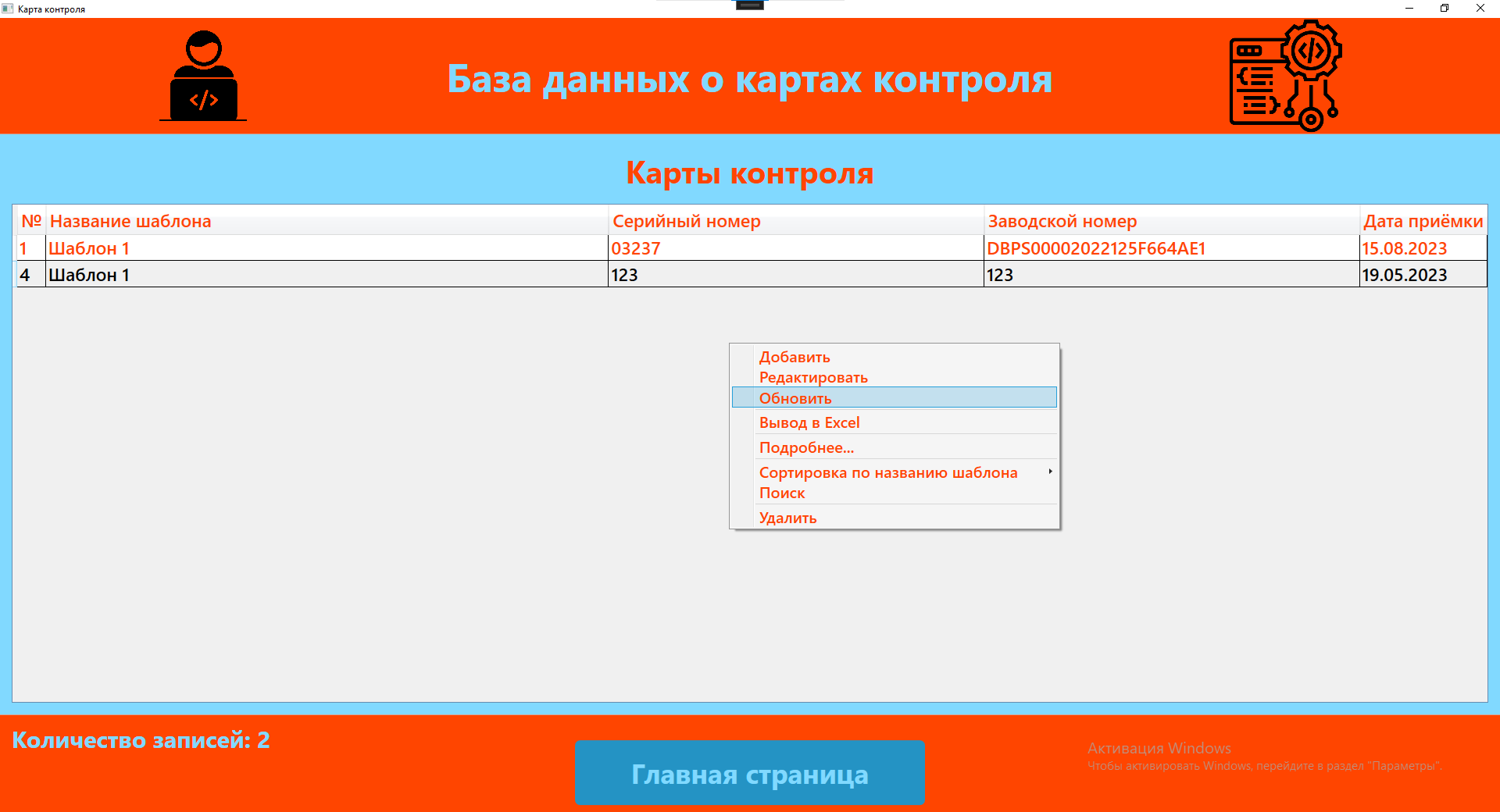


Рис. 14 «Страница карты контроля с вызовом обновления таблицы»

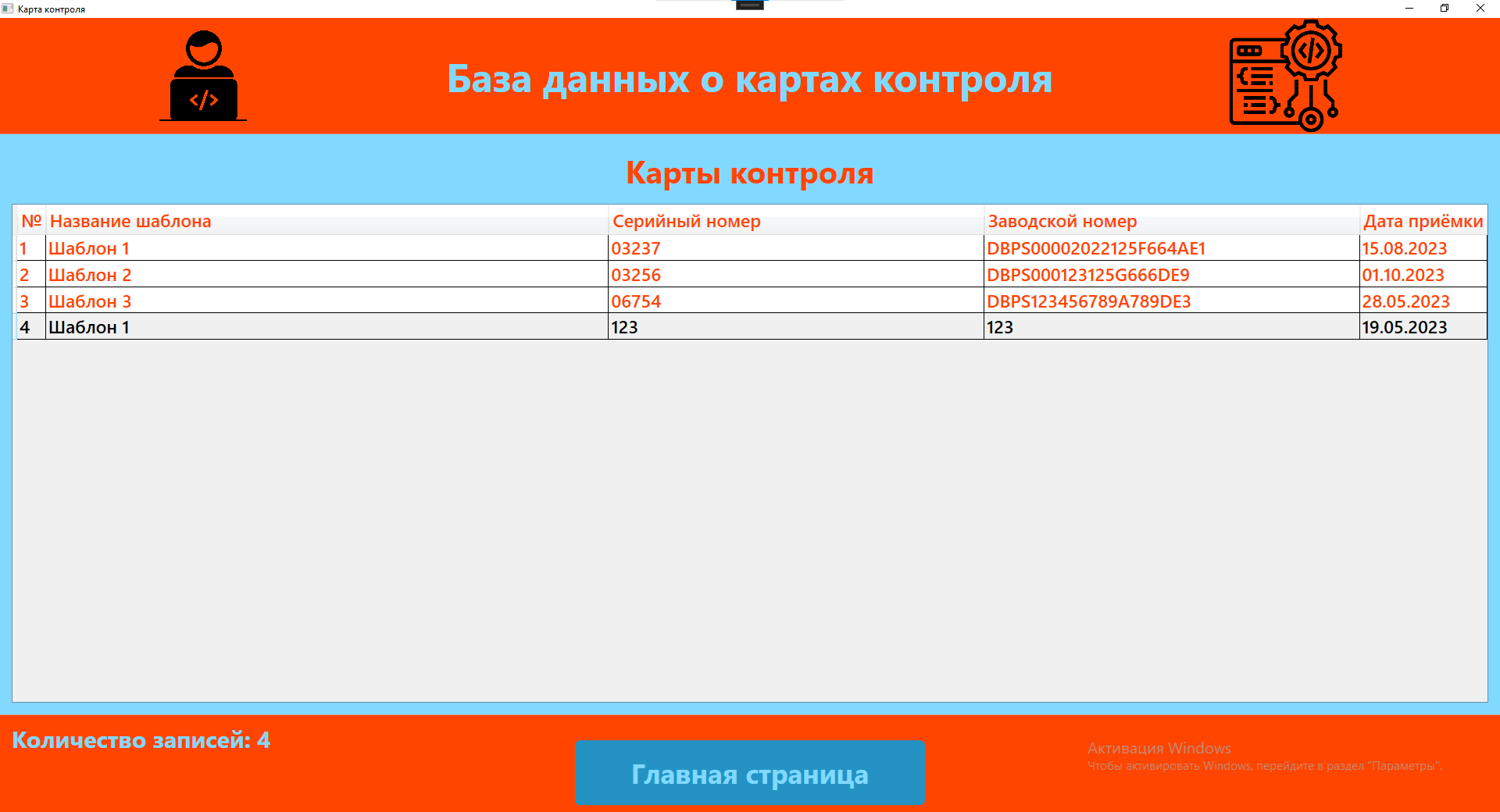


Рис. 15 «Страница карты контроля после вызова обновления таблицы»

**Сообщения пользователю:**

Попытка добавления новой карты контроля в список с неуказанными данными.

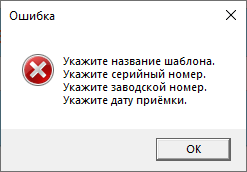


Рис.34 «Попытка добавления карты контроля»

Подтверждение удаления нескольких пунктов.

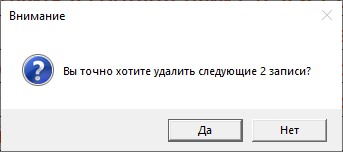


Рис.35 «Подтверждение удаления нескольких пунктов»

Информация после удаления нескольких пунктов.

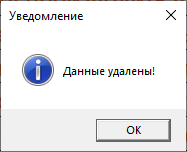
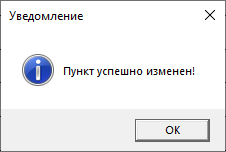


Рис.36 «Информация при удалении нескольких пунктов»

Информация после редактирования проекта.



## Рис.37 «Информация при редактировании пункта»3.6. Отладка ПО

Участок кода с подсчётом количества записей в таблице, работающий некорректно:

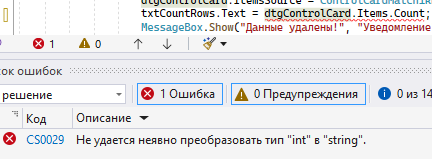


Рис.1 «Неявное преобразование типов»

Исправленный участок кода с подсчётом количества записей в таблице, работающий корректно:

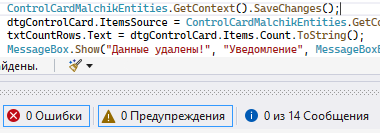


Рис.2 «Преобразование типа «int» в строку»

Участок кода с переходом на страницу добавления сотрудника, работающий некорректно:

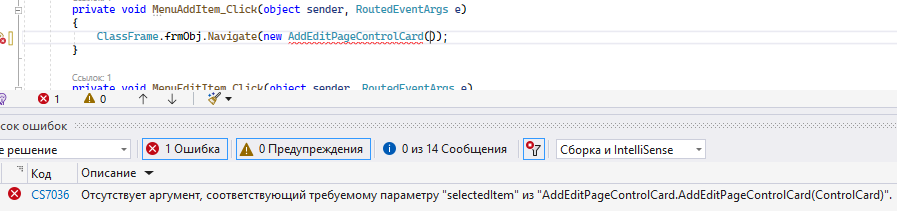


Рис.3 «Отсутствие аргумента, требуемой страницы»

Участок кода с переходом на страницу добавления сотрудника, работающий корректно:

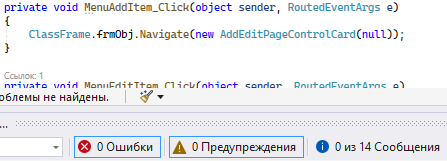


Рис.4 «Добавление пустого аргумента, требуемой странице»

## 3.7. Методика тестирования и испытания задачи

Тестовый сценарий №1:

|  |  |
| --- | --- |
| **Тестовый пример #** | 1 |
| **Приоритет тестирования** | Средний |
| **Заголовок/название теста** | Перейдите на страницу «Карты контроля» и нажмите в контекстном меню пункт «Сортировка по названию» и выберете любой параметр |
| **Краткое изложение теста** | Должно поменяться количество записей при использовании сортировки |
| **Этапы теста** | 1. Запустите программу через Visual Studio.  2. Попадаете в главное меню. Оттуда переходите на страницу «Карты контроля».  3. Затем нажмите в контекстном меню пункт «Сортировка от А до Я» |
| **Тестовые данные** | Сортировка списка карт контроля |
| **Ожидаемый результат** | Если будет использоваться сортировка, то данные должны быть отсортированы в списке |
| **Фактический результат** | Сортировка не применилась |
| **Статус** | Незачет |
| **Предварительное условие** | Исправьте код, чтобы сортировка по названию была выполнена |
| **Постусловие** | Подсчет количества записей работает совместно с фильтрами |
| **Примечания/комментарии** | Данный способ сортировки через LINQ не работал, т.к. для сортировок по количеству был применён другой способ через свойства элементов списка |

Тестовый сценарий №2:

|  |  |
| --- | --- |
| **Тестовый пример #** | 2 |
| **Приоритет тестирования** | Средний |
| **Заголовок/название теста** | Перейдите на страницу «Карты контроля» и нажмите кнопку «Поиск по номеру шаблона» |
| **Краткое изложение теста** | Должен выводиться список карт контроля с параметром, который вы выбрали и количество записей будет изменяться. |
| **Этапы теста** | 1. Запустите программу через Visual Studio.  2. Попадаете в главное меню. Оттуда переходите на страницу «Карты контроля».  3. Затем нажимайте на кнопку с выпадающим списком «Поиск по номеру шаблона». (Вбейте нужный вам формат) |
| **Тестовые данные** | Список карт контроля, количество записей |
| **Ожидаемый результат** | Если будет использоваться поиск, то количество записей должно поменяться. |
| **Фактический результат** | Количество записей карт контроля поменялось |
| **Статус** | Зачет |
| **Предварительное условие** | - |
| **Постусловие** | Подсчет количества записей работает с поиском |
| **Примечания/комментарии** | Пользователь может сам подсчитать количество записей |

## 3.8. Текст программы

Представлен в [Приложении](#_Приложение)

## 3.9 Предложения по улучшению

# Спектр функционала и удобства использования приложением можно расширить. К примеру, можно настроить совместную работу сортировок, поиска и фильтров. Добавить регистрацию новых пользователей. Для страниц добавления и редактирования карт контроля добавить связь со страницей структура карт контроля для мгновенного изменения записей сразу в нескольких таблицах.

# Заключение

Производственную практику проходила в организации АО «Стекломаш», которая является одним из старейших предприятий промышленного комплекса, разработчиком и серийным производителем изделий для рельсового и автомобильного транспорта, промышленных электронных устройств, светопрозрачных конструкций.

Проведен анализ материально-технической базы, который показал, что в ИТ-отделе находится 6 компьютеров и 1 сервер. Установлено следующее программное обеспечение: операционные системы Windows 7 и Windows 10, офисный пакет Microsoft Office 2016, Web-браузер Google Chrome.

Согласно постановке задачи, спроектирована предметная область и разработана база данных. Разработана схема базы данных. Произведена нормализация данных. Для работы была использована СУБД Microsoft SQL Server Management Studio 18.

В соответствии с постановкой задачи было разработано техническое задание. Согласно техническому заданию разработана программа по учёту оборудования на предприятии со следующими функциональными возможностями:

* Добавление данных в таблицы «Карты контроля», «Шаблоны», «Разделы», «Пункты», «Ответы»
* Авторизация с помощью логина и пароля
* Редактирование данных в таблицах «Карты контроля», «Шаблоны», «Разделы», «Пункты», «Ответы»
* Просмотр данных из таблиц «Карты контроля», «Шаблоны», «Разделы», «Пункты», «Ответы»

В техническом задании отражены требования к составу технических средств, информационно-программной совместимости и требования к надёжности, которые должны быть соблюдены во время разработки, внедрении и эксплуатации приложения.

Для разработки приложения использовались следующие средства: Microsoft Visual Studio, SQL Server Management Studio (SSMS).

Приложение обладает удобным графическим интерфейсом.

Для отладки и тестирования приложения использовались встроенные программные средства языка C#.

На основании созданного приложения и в соответствие с ГОСТами ГОСТ «19.505 – 79», ГОСТ «19.503 – 79». Разработана техническая документация: руководство пользователя, руководство программиста, программа и методика испытаний, текст программы.

# Список использованной литературы

1 1. Гохберг, Г.С. Информационные технологии: Учебник / Г.С. Гохберг. - М.: Academia, 2018. - 474 c.

2. Емельянов, С.В. Информационные технологии и вычислительные системы / С.В. Емельянов. - М.: Ленанд, 2015. - 96 c.

3. Дейтел П. Как программировать на Visual C# 2012; Питер - М., 2014. - 316 c.

4. Джозеф Албахари C# 5.0. Справочник. Полное описание языка; Диалектика / Вильямс - М., 2014. - 545 c.

5. Колесников А. П. Методы численного анализа, изложенные на языке формул и алгоритмическом языке C#; Высшая школа - Москва, 2017. - 414 c.

6. Культин Никита Борисович Основы программирования в Microsoft Visual C# 2010; БХВ-Петербург - М., 2011. - 384 c.

7. Бен-Ган, Ицик Microsoft SQL Server 2012. Основы T-SQL / Ицик Бен-Ган. - М.: Эксмо, 2016. - 722 c.

8. Вишневский, Алексей Microsoft SQL Server. Эффективная работа / Алексей Вишневский. - М.: Питер, 2015. - 374 c.

9. Дэвидсон, Луис Проектирование баз данных на SQL Server 2000 / Луис Дэвидсон. - М.: Бином. Лаборатория знаний, 2016. - 662 c.

10. Тернстрем, Т. Microsoft SQL Server 2008. Разработка баз данных. Учебный курс Microsoft / Т. Тернстрем. - М.: Русская Редакция, 2017. - 781 c.

11. Великий Фаберже. Искусство ювелиров придворной фирмы. - Москва: Машиностроение, 2019. - 144 c.

12. Плешев А.М. Классификация и ассортимент ювелирных изделий. Кооперативная торговля ювелирных изделий. - М.: 2008

# Приложение

***MainWindow*.xaml:**

<Grid>

<Frame x:Name="frm" NavigationUIVisibility="Hidden"/>

</Grid>

MainWindow.cs:

public MainWindow()

{

InitializeComponent();

frm.Navigate(new AuthorizationPage());

Manager.frm = frm;

}

***AccPage1.xaml:***

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="50"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="30"/>

</Grid.RowDefinitions>

<Grid Background="#4B0082" Grid.Row="0" Grid.ColumnSpan="3"/>

<Grid Background="#4B0082" Grid.Row="2" Grid.ColumnSpan="3"/>

<StackPanel Orientation="Horizontal" Grid.Row="0">

<materialDesign:PackIcon Kind="Account" Width="40" Height="40"

Margin="5" Background="#E0FFFF"/>

<TextBlock FontWeight="DemiBold" Foreground="#E0FFFF"

FontSize="22" Margin="0 10 0 0"

x:Name="txbFIO" Text="{Binding Account.FIO}"/>

</StackPanel>

<Button x:Name="back" Grid.Row="2" Click="back\_Click"

Margin="0 0 15 0" Height="30"

HorizontalAlignment="Right" Content="Назад"/>

<DataGrid Background="#FFFAF0" AutoGenerateColumns="False" IsReadOnly="True" Height="Auto"

Name="dtgrd" Grid.Row="1" VerticalAlignment="Top">

<DataGrid.Columns>

<DataGridTextColumn Width="20\*" Header="ФИО"

Binding="{Binding FIO}"/>

<DataGridTextColumn Width="15\*" Header="Логин"

Binding="{Binding login}"/>

<DataGridTextColumn Width="20\*" Header="Пароль"

Binding="{Binding password}"/>

<DataGridTextColumn Width="20\*" Header="Роль"

Binding="{Binding Role.role1}"/>

</DataGrid.Columns>

</DataGrid>

</Grid>

***AccPage1.cs:***

public AccPage1()

{

InitializeComponent();

txbFIO.Text = ConnectODB.equipObj.Account.FirstOrDefault(x => x.id == ConnectODB.id\_entered\_fio).FIO.ToString();

dtgrd.ItemsSource = ConnectODB.equipObj.Account.ToList();

}

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

{

Manager.frm.Navigate(new AdminPage1(ConnectODB.id\_entered\_fio));

}

***AdminPage1.xaml:***

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="50"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="30"/>

</Grid.RowDefinitions>

<Grid Background="#4B0082" Grid.Row="0" Grid.ColumnSpan="3"/>

<Grid Background="#4B0082" Grid.Row="2" Grid.ColumnSpan="3"/>

<StackPanel Orientation="Horizontal" Grid.Row="0">

<materialDesign:PackIcon Kind="Account" Width="40" Height="40"

Margin="5" Background="#E0FFFF"/>

<TextBlock FontWeight="DemiBold" Foreground="#E0FFFF"

FontSize="22" Margin="0 10 0 0"

x:Name="txbFIO" Text="{Binding Account.FIO}"/>

</StackPanel>

<Grid Grid.Row="1">

<Grid.ColumnDefinitions>

<ColumnDefinition Width="280"/>

<ColumnDefinition Width="\*"/>

</Grid.ColumnDefinitions>

<StackPanel Orientation="Vertical">

<Grid>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="Auto"/>

<ColumnDefinition Width="\*"/>

</Grid.ColumnDefinitions>

<Grid.RowDefinitions>

<RowDefinition Height="Auto"/>

<RowDefinition Height="Auto"/>

<RowDefinition Height="Auto"/>

<RowDefinition Height="Auto"/>

<RowDefinition Height="Auto"/>

<RowDefinition Height="Auto"/>

</Grid.RowDefinitions>

<TextBox materialDesign:HintAssist.Hint="ФИО сотрудника"

Style="{StaticResource MaterialDesignFloatingHintTextBox}"

x:Name="fio" Margin="0 0 0 5" FontSize="16" Grid.Row="0"

Width="250"/>

<TextBox materialDesign:HintAssist.Hint="Придумайте логин"

Style="{StaticResource MaterialDesignFloatingHintTextBox}"

x:Name="login" Margin="0 0 0 5" FontSize="16" Grid.Row="1"

Width="250"/>

<TextBox materialDesign:HintAssist.Hint="Придумайте пароль"

Style="{StaticResource MaterialDesignFloatingHintTextBox}"

x:Name="password" Margin="0 0 0 5" FontSize="16" Grid.Row="2"

Width="250"/>

<TextBlock Text="Роль сотрудника в приложении:" Foreground="#808080"

x:Name="role" Margin="0 15 0 0" FontSize="15" Grid.Row="3"

Width="250"/>

<ComboBox x:Name="cmbRole" Grid.Column="0" Grid.Row="4"

SelectionChanged="cmbRole\_SelectionChanged" Margin="0 10 0 0 "

IsEditable="True" FontSize="16"/>

<Button x:Name="createAcc" Content="Создать аккаунт"

Grid.Column="0" Grid.Row="5"

Width="160" Margin="0 25 0 0" Click="createAcc\_Click"/>

</Grid>

</StackPanel>

<Grid Grid.Column="1">

<Grid.RowDefinitions>

<RowDefinition Height="145"/>

<RowDefinition Height="\*"/>

</Grid.RowDefinitions>

<Border Grid.RowSpan="2" Background="#E6E6FA" MinWidth="400" MaxWidth="450"

VerticalAlignment="Top" HorizontalAlignment="Center" Margin="50 30 0 0" Padding="15"

CornerRadius="25">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="\*"/>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="\*"/>

<ColumnDefinition Width="\*"/>

</Grid.ColumnDefinitions>

<Button x:Name="request" Content="Заявки"

Click="request\_Click" Width="160" Margin="0"

Grid.Column="0" Grid.Row="0" HorizontalContentAlignment="Center"/>

<Button x:Name="equip" Content="Оборудование"

Click="equip\_Click" Width="160" Margin="0"

Grid.Column="0" Grid.Row="1" HorizontalContentAlignment="Center"/>

<Button x:Name="repair" Content="Ремонт"

Click="repair\_Click" Width="160" Margin="0"

Grid.Column="1" Grid.Row="0" HorizontalContentAlignment="Center"/>

<Button x:Name="acc" Content="Аккаунты"

Click="acc\_Click" Width="160" Margin="5"

Grid.Column="1" Grid.Row="1" HorizontalContentAlignment="Center"/>

<Button x:Name="btnNewPassw" Click="btnNewPassw\_Click"

Content="Сменить пароль" Margin="5 20 5 0"

Width="160" HorizontalAlignment="Center"

Grid.Row="2" Grid.ColumnSpan="2"/>

</Grid>

</Border>

</Grid>

</Grid>

<Button x:Name="back" Grid.Row="2" Click="back\_Click"

Margin="0 0 15 0" Height="30"

HorizontalAlignment="Right" Content="Назад"/>

</Grid>

***AdminPage1.cs:***

public AdminPage1(int args)

{

InitializeComponent();

txbFIO.Text = ConnectODB.equipObj.Account.FirstOrDefault(x => x.id == ConnectODB.id\_entered\_fio).FIO.ToString();

cmbRole.SelectedValuePath = "id";

cmbRole.DisplayMemberPath = "role1";

cmbRole.ItemsSource = ConnectODB.equipObj.Role.ToList();

}

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

{

Manager.frm.Navigate(new AuthorizationPage());

}

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

{

if (ConnectODB.equipObj.Account.Count(x => x.login == login.Text) > 0)

{

MessageBox.Show("Пользователь с таким логином уже существует!", "Уведомление",

MessageBoxButton.OK, MessageBoxImage.Information);

return;

}

else

{

try

{

Account accObj = new Account()

{

FIO = fio.Text,

login = login.Text,

password = password.Text,

id\_role = cmbRole.SelectedIndex+1,

};

ConnectODB.equipObj.Account.Add(accObj);

ConnectODB.equipObj.SaveChanges();

MessageBox.Show("Пользователь создан!", "Уведомление",

MessageBoxButton.OK,

MessageBoxImage.Information);

}

catch (Exception ex)

{

MessageBox.Show("Ошибка работы приложения: "

+ ex.Message.ToString(), "Ошибка",

MessageBoxButton.OK, MessageBoxImage.Warning);

}

}

fio.Clear();

login.Clear();

password.Clear();

cmbRole.Text = "";

}

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

{

Manager.frm.Navigate(new RequestPage1());

}

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

{

Manager.frm.Navigate(new EquipmentPage1());

}

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

{

Manager.frm.Navigate(new RepairPage1());

}

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

{

Manager.frm.Navigate(new AccPage1());

}

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

{

NewPasswordWindow psswrdWndw = new NewPasswordWindow();

psswrdWndw.Show();

}

***AuthorizationPage.xaml:***

<Grid Background="#4B0082">

<Border MinWidth="400" Background="#E6E6FA" MaxWidth="500"

VerticalAlignment="Center" Margin="15" Padding="40"

CornerRadius="30">

<Border.Effect>

<DropShadowEffect BlurRadius="60" ShadowDepth="10"/>

</Border.Effect>

<StackPanel>

<TextBlock Text="Авторизация" Margin="0 0 0 20"

FontSize="30" FontWeight="DemiBold"/>

<TextBox materialDesign:HintAssist.Hint="Введите логин"

Style="{StaticResource MaterialDesignFloatingHintTextBox}"

x:Name="log" Margin="0 0 0 20" FontSize="15"/>

<PasswordBox materialDesign:HintAssist.Hint="Введите пароль"

Style="{StaticResource MaterialDesignFloatingHintPasswordBox}"

x:Name="passw" Margin="0 0 0 40" FontSize="15"/>

<Button x:Name="voiti" Content="Войти" Margin="0 0 0 5"

Width="120" FontSize="18" Height="40"

Click="voiti\_Click"/>

</StackPanel>

</Border>

</Grid>

***AuthorizationPage.cs:***

public AuthorizationPage()

{

InitializeComponent();

}

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

{

try

{

var emplObj = ConnectODB.equipObj.Account.FirstOrDefault(

x => x.login.ToLower() == log.Text.ToLower() &&

x.password == passw.Password);

if (emplObj == null)

{

MessageBox.Show("Такого аккаунта не существует!", "Уведомление",

MessageBoxButton.OK, MessageBoxImage.Information);

log.Clear();

passw.Clear();

}

else

{

ConnectODB.id\_entered\_role = (int)emplObj.id\_role;

ConnectODB.id\_entered\_fio = (int)emplObj.id;

if (ConnectODB.id\_entered\_role == 1)

Manager.frm.Navigate(new AdminPage1(ConnectODB.id\_entered\_fio));

else if (ConnectODB.id\_entered\_role == 2)

Manager.frm.Navigate(new MasterPage1(ConnectODB.id\_entered\_fio));

else

Manager.frm.Navigate(new MechanicPage1(ConnectODB.id\_entered\_fio));

}

}

catch (Exception ex)

{

MessageBox.Show("Критический сбой в работе приложения: " + ex.Message.ToString(), "Уведомление",

MessageBoxButton.OK, MessageBoxImage.Warning);

}

}

***EquipmentPage1.xaml:***

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="50"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="30"/>

</Grid.RowDefinitions>

<Grid Background="#4B0082" Grid.Row="0" Grid.ColumnSpan="3"/>

<Grid Background="#4B0082" Grid.Row="2" Grid.ColumnSpan="3"/>

<StackPanel Orientation="Horizontal" Grid.Row="0">

<materialDesign:PackIcon Kind="Account" Width="40" Height="40"

Margin="5" Background="#E0FFFF"/>

<TextBlock FontWeight="DemiBold" Foreground="#E0FFFF"

FontSize="22" Margin="0 10 0 0"

x:Name="txbFIO" Text="{Binding Account.FIO}"/>

</StackPanel>

<Button x:Name="back" Grid.Row="2" Click="back\_Click"

Margin="0 0 15 0" Height="30"

HorizontalAlignment="Right" Content="Назад"/>

<Grid Grid.Row="1">

<Grid.RowDefinitions>

<RowDefinition Height="10\*"/>

<RowDefinition Height="10\*"/>

<RowDefinition Height="80\*"/>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition/>

<ColumnDefinition/>

<ColumnDefinition/>

<ColumnDefinition/>

<ColumnDefinition/>

</Grid.ColumnDefinitions>

<DataGrid Background="#FFFAF0" AutoGenerateColumns="False" IsReadOnly="True"

Height="Auto" Name="dtgrd" VerticalAlignment="Top"

Grid.Row="3" Grid.ColumnSpan="5">

<DataGrid.Columns>

<DataGridTextColumn Width="20\*" Header="Название"

Binding="{Binding title}"/>

<DataGridTextColumn Width="15\*" Header="Инвентарный номер"

Binding="{Binding inventory\_number}"/>

<DataGridTextColumn Width="20\*" Header="Год выпуска"

Binding="{Binding year\_of\_issue}"/>

<DataGridTextColumn Width="20\*" Header="Тип"

Binding="{Binding Type\_of\_equipment.type}"/>

<DataGridTextColumn Width="20\*" Header="Статус"

Binding="{Binding Repair\_status.status}"/>

<DataGridTextColumn Width="20\*" Header="Цех"

Binding="{Binding Manufactory.number}"/>

</DataGrid.Columns>

</DataGrid>

<TextBlock Text="Поиск по названию:" Grid.Row="0" Grid.Column="0"

HorizontalAlignment="Center" FontSize="13" VerticalAlignment="Center"

Foreground="#4B0082"/>

<ComboBox x:Name="cmb\_ttl" Grid.Column="0" Grid.Row="1"

IsEditable="True" FontSize="16"

SelectionChanged="cmb\_ttl\_SelectionChanged"/>

<TextBlock Text="Поиск по инвент. номеру:" Grid.Row="0" Grid.Column="1"

HorizontalAlignment="Center" FontSize="13" VerticalAlignment="Center"

Foreground="#4B0082"/>

<ComboBox x:Name="cmb\_invN" Grid.Column="1" Grid.Row="1"

IsEditable="True" FontSize="16"

SelectionChanged="cmb\_invN\_SelectionChanged"/>

<TextBlock Text="Поиск по типу оборудования:" Grid.Row="0" Grid.Column="2"

HorizontalAlignment="Center" FontSize="13" VerticalAlignment="Center"

Foreground="#4B0082" TextWrapping="Wrap"/>

<ComboBox x:Name="cmb\_type" Grid.Column="2" Grid.Row="1"

IsEditable="True" FontSize="16"

SelectionChanged="cmb\_type\_SelectionChanged"/>

<TextBlock Text="Поиск по статусу:" Grid.Row="0" Grid.Column="3"

HorizontalAlignment="Center" FontSize="13" VerticalAlignment="Center"

Foreground="#4B0082"/>

<ComboBox x:Name="cmb\_status" Grid.Column="3" Grid.Row="1"

IsEditable="True" FontSize="16"

SelectionChanged="cmb\_status\_SelectionChanged"/>

<TextBlock Text="Поиск по цеху:" Grid.Row="0" Grid.Column="4"

HorizontalAlignment="Center" FontSize="13" VerticalAlignment="Center"

Foreground="#4B0082"/>

<ComboBox x:Name="cmb\_manuf" Grid.Column="4" Grid.Row="1"

IsEditable="True" FontSize="16"

SelectionChanged="cmb\_manuf\_SelectionChanged"/>

</Grid>

</Grid>

***EquipmentPage1.cs:***

public EquipmentPage1()

{

InitializeComponent();

txbFIO.Text = ConnectODB.equipObj.Account.FirstOrDefault(x => x.id == ConnectODB.id\_entered\_fio).FIO.ToString();

dtgrd.ItemsSource = ConnectODB.equipObj.Equipment.ToList();

cmb\_invN.SelectedValuePath = "id";

cmb\_invN.DisplayMemberPath = "inventory\_number";

cmb\_invN.ItemsSource = ConnectODB.equipObj.Equipment.ToList();

cmb\_type.SelectedValuePath = "id";

cmb\_type.DisplayMemberPath = "type";

cmb\_type.ItemsSource = ConnectODB.equipObj.Type\_of\_equipment.ToList();

cmb\_status.SelectedValuePath = "id";

cmb\_status.DisplayMemberPath = "status";

cmb\_status.ItemsSource = ConnectODB.equipObj.Repair\_status.ToList();

cmb\_manuf.SelectedValuePath = "id";

cmb\_manuf.DisplayMemberPath = "number";

cmb\_manuf.ItemsSource = ConnectODB.equipObj.Manufactory.ToList();

cmb\_ttl.SelectedValuePath = "id";

cmb\_ttl.DisplayMemberPath = "title";

cmb\_ttl.ItemsSource = ConnectODB.equipObj.Equipment.ToList();

}

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

{

if (ConnectODB.id\_entered\_role == 1)

Manager.frm.Navigate(new AdminPage1(ConnectODB.id\_entered\_fio));

else if (ConnectODB.id\_entered\_role == 2)

Manager.frm.Navigate(new MasterPage1(ConnectODB.id\_entered\_fio));

else

Manager.frm.Navigate(new MechanicPage1(ConnectODB.id\_entered\_fio));

}

private void cmb\_invN\_SelectionChanged(object sender, SelectionChangedEventArgs e)

{

int EqId = (int)cmb\_invN.SelectedValue;

dtgrd.ItemsSource = ConnectODB.equipObj.Equipment.Where(x => x.id == EqId).ToList();

}

private void cmb\_type\_SelectionChanged(object sender, SelectionChangedEventArgs e)

{

int EqId = (int)cmb\_type.SelectedValue;

dtgrd.ItemsSource = ConnectODB.equipObj.Equipment.Where(x => x.id\_type == EqId).ToList();

}

private void cmb\_status\_SelectionChanged(object sender, SelectionChangedEventArgs e)

{

int EqId = (int)cmb\_status.SelectedValue;

dtgrd.ItemsSource = ConnectODB.equipObj.Equipment.Where(x => x.id\_status == EqId).ToList();

}

private void cmb\_manuf\_SelectionChanged(object sender, SelectionChangedEventArgs e)

{

int EqId = (int)cmb\_manuf.SelectedValue;

dtgrd.ItemsSource = ConnectODB.equipObj.Equipment.Where(x => x.id\_manufact == EqId).ToList();

}

private void cmb\_ttl\_SelectionChanged(object sender, SelectionChangedEventArgs e)

{

int EqId = (int)cmb\_ttl.SelectedValue;

dtgrd.ItemsSource = ConnectODB.equipObj.Equipment.Where(x => x.id == EqId).ToList();

}

***MasterPage1.xaml:***

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="50"/>

<RowDefinition Height="50"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="30"/>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="Auto"/>

<ColumnDefinition Width="\*"/>

<ColumnDefinition Width="300"/>

</Grid.ColumnDefinitions>

<Grid Background="#4B0082" Grid.Column="0" Grid.Row="0" Grid.ColumnSpan="3"/>

<Grid Background="#4B0082" Grid.Column="0" Grid.Row="3" Grid.ColumnSpan="3"/>

<StackPanel Orientation="Horizontal" Grid.Row="0" Grid.ColumnSpan="3">

<materialDesign:PackIcon Kind="Account" Width="40" Height="40" Margin="5" Background="#E0FFFF"/>

<TextBlock FontWeight="DemiBold" Foreground="#E0FFFF"

FontSize="22" Margin="0 10 0 0"

x:Name="txbFIO" Text="{Binding Account.FIO}"/>

<TextBox Text="Цех: " FontSize="22" FontWeight="DemiBold"

Margin="100 5 0 0" Foreground="#E0FFFF"/>

<TextBlock x:Name="txbManufact" FontSize="22" FontWeight="DemiBold"

Margin="0 10 0 0" Text="{Binding Manufactory.number}"

Foreground="#E0FFFF"/>

</StackPanel>

<StackPanel Grid.Row="1" Grid.RowSpan="2" Orientation="Vertical" Margin="5 15 0 0">

<TextBlock Text="Название оборудования:" Margin="5 0 0 10" FontSize="16"

Grid.Row="2"/>

<ComboBox x:Name="cmb\_eq" IsEditable="True" FontSize="16" Margin="5 0 0 10"/>

<TextBlock Text="Кратко изложите суть проблемы:"

Margin="5 0 0 10" FontSize="16"/>

<TextBox Margin="5 0 0 10" Padding="5" FontSize="16"

MinHeight="60" x:Name="descr"/>

<Button Content="Оставить заявку на ремонт" x:Name="btnRequest" Margin="20 20 20 0"

Click="btnRequest\_Click" Grid.Row="2"/>

</StackPanel>

<Border Grid.Row="1" Grid.Column="1" Grid.ColumnSpan="2" Grid.RowSpan="2"

Background="#E6E6FA" VerticalAlignment="Center" HorizontalAlignment="Center"

Margin="190 -60 0 0" Padding="30" CornerRadius="25">

<StackPanel Orientation="Vertical">

<Button x:Name="equip" Content="Оборудование"

Click="equip\_Click" Width="160" Margin="5"

HorizontalContentAlignment="Center"/>

<Button x:Name="repair" Content="Ремонт"

Click="repair\_Click" Width="160" Margin="5"

HorizontalContentAlignment="Center"/>

<Button x:Name="btnNewPassw" Click="btnNewPassw\_Click"

Content="Сменить пароль" Margin="5 20 5 0"

Width="160" HorizontalAlignment="Right"/>

</StackPanel>

</Border>

<Button x:Name="back" Grid.Row="3" Grid.Column="2" Click="back\_Click"

Margin="0 0 15 0" Height="30"

HorizontalAlignment="Right" Content="Назад"/>

</Grid>

***MasterPage1.cs:***

public MasterPage1(int args)

{

InitializeComponent();

txbFIO.Text = ConnectODB.equipObj.Account.FirstOrDefault(x => x.id == ConnectODB.id\_entered\_fio).FIO.ToString();

txbManufact.Text = ConnectODB.equipObj.Manufactory.FirstOrDefault(x => x.id\_responsible == ConnectODB.id\_entered\_fio).number.ToString();

cmb\_eq.SelectedValuePath = "id";

cmb\_eq.DisplayMemberPath = "title";

cmb\_eq.ItemsSource = ConnectODB.equipObj.Equipment.ToList();

}

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

{

Request req = new Request()

{

status = "В обработке",

id\_equip = cmb\_eq.SelectedIndex+1,

short\_description = descr.Text,

};

Repair rep = new Repair()

{

id\_request = req.id,

};

ConnectODB.equipObj.Request.Add(req);

ConnectODB.equipObj.SaveChanges();

ConnectODB.equipObj.Repair.Add(rep);

ConnectODB.equipObj.SaveChanges();

MessageBox.Show("Ваша заявка отправлена на рассмотрение", "Уведомление",

MessageBoxButton.OK,

MessageBoxImage.Information);

}

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

{

Manager.frm.Navigate(new AuthorizationPage());

}

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

{

NewPasswordWindow psswrdWndw = new NewPasswordWindow();

psswrdWndw.Show();

}

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

{

Manager.frm.Navigate(new EquipmentPage1());

}

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

{

Manager.frm.Navigate(new RepairPage1());

}

***MechanicPage1.xaml:***

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="50"/>

<RowDefinition Height="50"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="30"/>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="Auto"/>

<ColumnDefinition Width="\*"/>

<ColumnDefinition Width="300"/>

</Grid.ColumnDefinitions>

<Grid Background="#4B0082" Grid.Column="0" Grid.Row="0" Grid.ColumnSpan="3"/>

<Grid Background="#4B0082" Grid.Column="0" Grid.Row="3" Grid.ColumnSpan="3"/>

<StackPanel Orientation="Horizontal" Grid.Row="0" Grid.ColumnSpan="3">

<materialDesign:PackIcon Kind="Account" Width="40" Height="40" Margin="5" Background="#E0FFFF"/>

<TextBlock FontWeight="DemiBold" Foreground="#E0FFFF"

FontSize="22" Margin="0 10 0 0"

x:Name="txbFIO" Text="{Binding Account.FIO}"/>

</StackPanel>

<Border Grid.Row="1" Grid.Column="1" Grid.ColumnSpan="2" Grid.RowSpan="2"

Background="#E6E6FA" VerticalAlignment="Center" HorizontalAlignment="Center"

Padding="30" CornerRadius="25">

<StackPanel Orientation="Vertical">

<Button x:Name="equip" Content="Оборудование"

Click="equip\_Click" Width="160" Margin="5"

HorizontalContentAlignment="Center"/>

<Button x:Name="repair" Content="Ремонт"

Click="repair\_Click" Width="160" Margin="5"

HorizontalContentAlignment="Center"/>

<Button x:Name="request" Content="Заявки"

Click="request\_Click" Width="160" Margin="5"

Grid.Column="0" Grid.Row="0" HorizontalContentAlignment="Center"/>

<Button x:Name="btnNewPassw" Click="btnNewPassw\_Click"

Content="Сменить пароль" Margin="5 20 5 0"

Width="160" HorizontalAlignment="Right"/>

</StackPanel>

</Border>

<Button x:Name="back" Grid.Row="3" Grid.Column="2" Click="back\_Click"

Margin="0 0 15 0" Height="30"

HorizontalAlignment="Right" Content="Назад"/>

</Grid>

***MechanicPage1.cs:***

public MechanicPage1(int args)

{

InitializeComponent();

txbFIO.Text = ConnectODB.equipObj.Account.FirstOrDefault(x => x.id == ConnectODB.id\_entered\_fio).FIO.ToString();

}

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

{

NewPasswordWindow psswrdWndw = new NewPasswordWindow();

psswrdWndw.Show();

}

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

{

Manager.frm.Navigate(new MechRepairPage1());

}

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

{

Manager.frm.Navigate(new EquipmentPage1());

}

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

{

Manager.frm.Navigate(new AuthorizationPage());

}

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

{

Manager.frm.Navigate(new MechReqPage1());}

***MechRepairPage1.xaml:***

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="50"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="40"/>

<RowDefinition Height="30"/>

</Grid.RowDefinitions>

<Grid Background="#4B0082" Grid.Row="0" Grid.ColumnSpan="3"/>

<Grid Background="#4B0082" Grid.Row="3" Grid.ColumnSpan="3"/>

<StackPanel Orientation="Horizontal" Grid.Row="0">

<materialDesign:PackIcon Kind="Account" Width="40" Height="40"

Margin="5" Background="#E0FFFF"/>

<TextBlock FontWeight="DemiBold" Foreground="#E0FFFF"

FontSize="22" Margin="0 10 0 0"

x:Name="txbFIO" Text="{Binding Account.FIO}"/>

</StackPanel>

<DataGrid Background="#FFFAF0" AutoGenerateColumns="False" Height="Auto"

Name="dtgrd" Grid.Row="1" VerticalAlignment="Top">

<DataGrid.Columns>

<DataGridTextColumn Width="20\*" Header="Дата начала"

Binding="{Binding start\_date, StringFormat=d, ConverterCulture='en-GB'}"

IsReadOnly="False"/>

<DataGridTextColumn Width="15\*" Header="Дата окончания"

Binding="{Binding end\_date, StringFormat=d, ConverterCulture='en-GB'}"

IsReadOnly="False"/>

<DataGridTextColumn Width="20\*" Header="Механик"

Binding="{Binding Account.FIO}" IsReadOnly="True"/>

<DataGridTextColumn Width="20\*" Header="Заявка"

Binding="{Binding Request.short\_description}"

IsReadOnly="True"/>

</DataGrid.Columns>

</DataGrid>

<Button x:Name="back" Grid.Row="3" Click="back\_Click"

Margin="0 0 15 0" Height="30"

HorizontalAlignment="Right" Content="Назад"/>

<Button x:Name="btn\_Save" Grid.Row="2" Click="btn\_Save\_Click"

HorizontalAlignment="Center" VerticalAlignment="Center"

Width="120" Content="Сохранить"/>

</Grid>

***MechRepairPage1.cs:***

public MechRepairPage1()

{

InitializeComponent();

txbFIO.Text = ConnectODB.equipObj.Account.FirstOrDefault(x => x.id == ConnectODB.id\_entered\_fio).FIO.ToString();

dtgrd.ItemsSource = ConnectODB.equipObj.Repair.Where(x => x.id\_responsible == ConnectODB.id\_entered\_fio).ToList();

}

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

{

Manager.frm.Navigate(new MechanicPage1(ConnectODB.id\_entered\_fio));

}

private void btn\_Save\_Click(object sender, RoutedEventArgs e)

{

ConnectODB.equipObj.SaveChanges();

MessageBox.Show(

"Заявка принята!",

"Уведомление",

MessageBoxButton.OK,

MessageBoxImage.Information);

}

***MechReqPage1.xaml:***

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="50"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="30"/>

</Grid.RowDefinitions>

<Grid Background="#4B0082" Grid.Row="0" Grid.ColumnSpan="3"/>

<Grid Background="#4B0082" Grid.Row="2" Grid.ColumnSpan="3"/>

<StackPanel Orientation="Horizontal" Grid.Row="0">

<materialDesign:PackIcon Kind="Account" Width="40" Height="40"

Margin="5" Background="#E0FFFF"/>

<TextBlock FontWeight="DemiBold" Foreground="#E0FFFF"

FontSize="22" Margin="0 10 0 0"

x:Name="txbFIO" Text="{Binding Account.FIO}"/>

</StackPanel>

<DataGrid Background="#FFFAF0" AutoGenerateColumns="False" IsReadOnly="True" Height="220"

Name="dtgrd" Grid.Row="1">

<DataGrid.Columns>

<DataGridTextColumn Width="20\*" Header="Статус"

Binding="{Binding status}"

FontFamily="Times New Roman"/>

<DataGridTextColumn Width="15\*" Header="Оборудование"

Binding="{Binding Equipment.title}"

FontFamily="Times New Roman"/>

<DataGridTextColumn Width="20\*" Header="Краткое описание"

Binding="{Binding short\_description}"

FontFamily="Times New Roman"/>

</DataGrid.Columns>

</DataGrid>

<Button x:Name="back" Grid.Row="3" Grid.Column="2" Click="back\_Click"

Margin="0 0 15 0" Height="30"

HorizontalAlignment="Right" Content="Назад"/>

</Grid>

***MechReqPage1.cs:***

public MechReqPage1()

{

InitializeComponent();

txbFIO.Text = ConnectODB.equipObj.Account.FirstOrDefault(x => x.id == ConnectODB.id\_entered\_fio).FIO.ToString();

dtgrd.ItemsSource = ConnectODB.equipObj.Request.ToList();

}

private void btn\_edit\_Click(object sender, RoutedEventArgs e)

{

Manager.frm.Navigate(new MechRepairPage1());

}

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

{

Manager.frm.Navigate(new MechanicPage1(ConnectODB.id\_entered\_fio));

}

***RepairPage1.xaml:***

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="50"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="30"/>

</Grid.RowDefinitions>

<Grid Background="#4B0082" Grid.Row="0" Grid.ColumnSpan="3"/>

<Grid Background="#4B0082" Grid.Row="2" Grid.ColumnSpan="3"/>

<StackPanel Orientation="Horizontal" Grid.Row="0">

<materialDesign:PackIcon Kind="Account" Width="40" Height="40"

Margin="5" Background="#E0FFFF"/>

<TextBlock FontWeight="DemiBold" Foreground="#E0FFFF"

FontSize="22" Margin="0 10 0 0"

x:Name="txbFIO" Text="{Binding Account.FIO}"/>

</StackPanel>

<Button x:Name="back" Grid.Row="2" Click="back\_Click"

Margin="0 0 15 0" Height="30"

HorizontalAlignment="Right" Content="Назад"/>

<DataGrid Background="#FFFAF0" AutoGenerateColumns="False" IsReadOnly="True" Height="Auto"

Name="dtgrd" Grid.Row="1" VerticalAlignment="Top">

<DataGrid.Columns>

<DataGridTextColumn Width="20\*" Header="Дата начала"

Binding="{Binding start\_date, StringFormat=d, ConverterCulture='en-GB'}"/>

<DataGridTextColumn Width="15\*" Header="Дата окончания"

Binding="{Binding end\_date, StringFormat=d, ConverterCulture='en-GB'}"/>

<DataGridTextColumn Width="20\*" Header="Механик"

Binding="{Binding Account.FIO}"/>

<DataGridTextColumn Width="20\*" Header="Номер запроса"

Binding="{Binding Request.id}"/>

</DataGrid.Columns>

</DataGrid>

</Grid>

***RepairPage1.cs:***

public RepairPage1()

{

InitializeComponent();

txbFIO.Text = ConnectODB.equipObj.Account.FirstOrDefault(x => x.id == ConnectODB.id\_entered\_fio).FIO.ToString();

dtgrd.ItemsSource = ConnectODB.equipObj.Repair.ToList();

}

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

{

if (ConnectODB.id\_entered\_role == 1)

Manager.frm.Navigate(new AdminPage1(ConnectODB.id\_entered\_fio));

else if (ConnectODB.id\_entered\_role == 2)

Manager.frm.Navigate(new MasterPage1(ConnectODB.id\_entered\_fio));

else

Manager.frm.Navigate(new MechanicPage1(ConnectODB.id\_entered\_fio));

}

***RequestPage1.xaml:***

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="50"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="30"/>

</Grid.RowDefinitions>

<Grid Background="#4B0082" Grid.Row="0" Grid.ColumnSpan="3"/>

<Grid Background="#4B0082" Grid.Row="2" Grid.ColumnSpan="3"/>

<StackPanel Orientation="Horizontal" Grid.Row="0">

<materialDesign:PackIcon Kind="Account" Width="40" Height="40"

Margin="5" Background="#E0FFFF"/>

<TextBlock FontWeight="DemiBold" Foreground="#E0FFFF"

FontSize="22" Margin="0 10 0 0"

x:Name="txbFIO" Text="{Binding Account.FIO}"/>

</StackPanel>

<Button x:Name="back" Grid.Row="2" Click="back\_Click"

Margin="0 0 15 0" Height="30"

HorizontalAlignment="Right" Content="Назад"/>

<DataGrid Background="#FFFAF0" AutoGenerateColumns="False" IsReadOnly="True" Height="Auto"

Name="dtgrd" Grid.Row="1" VerticalAlignment="Top">

<DataGrid.Columns>

<DataGridTextColumn Width="20\*" Header="Статус заявки"

Binding="{Binding status}"/>

<DataGridTextColumn Width="15\*" Header="Оборудование"

Binding="{Binding Equipment.title}"/>

<DataGridTextColumn Width="20\*" Header="Мастер"

Binding="{Binding Account.FIO}"/>

</DataGrid.Columns>

</DataGrid>

</Grid>

***RequestPage1.cs:***

public RequestPage1()

{

InitializeComponent();

txbFIO.Text = ConnectODB.equipObj.Account.FirstOrDefault(x => x.id == ConnectODB.id\_entered\_fio).FIO.ToString();

dtgrd.ItemsSource = ConnectODB.equipObj.Request.ToList();

}

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

{

if (ConnectODB.id\_entered\_role == 1)

Manager.frm.Navigate(new AdminPage1(ConnectODB.id\_entered\_fio));

else if (ConnectODB.id\_entered\_role == 2)

Manager.frm.Navigate(new MasterPage1(ConnectODB.id\_entered\_fio));

else

Manager.frm.Navigate(new MechanicPage1(ConnectODB.id\_entered\_fio));

}

***ConnectODB.cs:***

class ConnectODB

{

public static EquipmentEntities equipObj = new EquipmentEntities();

public static int id\_entered\_role;

public static int id\_entered\_fio;

}

***Manager.cs:***

class Manager

{

public static Frame frm { get; set; }

}