**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.

**Microsoft Excel -** программа для работы с электронными таблицами, созданная корпорацией Microsoft для Microsoft Windows, Windows NT и Mac OS, а также Android, iOS и Windows Phone. Она предоставляет возможности экономико-статистических расчетов, графические инструменты и, за исключением Excel 2008 под Mac OS X, язык макропрограммирования VBA. Microsoft Excel входит в состав Microsoft Office.

## 2.2. Состав технических средств

Таблица №1 «Технические средства ПК»

|  |  |  |
| --- | --- | --- |
| **Название** | **Фото** | **Характеристики** |
| **Процессор** | | |
| Процессор AMD Ryzen 5 5600G, SocketAM4 |  | Сокет: AM4Линейка процессоров: AMD Ryzen 5Количество ядер: 6 шт.Техпроцесс: 7 нмЧастота процессора: 3900 МГцТип памяти: DDR4Вид поставки: BOX, OEMЯдро: CezanneОбъем кэша L3: 16 МБТепловыделение: 65 ВтНазначение процессора:настольный ПК, игровойПроизводитель: AMD |
| **Оперативная память** | | |
| DIMM Kingston KVR26N19D8/16 |  | Тип: DDR4  Объем одного модуля: 16 ГБ  Тактовая частота: 2666 МГц  Форм-фактор: DIMM Количество модулей в комплекте: 1 шт.  CL: 19  Особенности: офисная, Unregistered  Производитель: Kingston |
| **Материнская плата** | | |
| ASUS H110M-C2/CSM |  | Сокет: LGA1151Название чипсета: Intel H110Форм-фактор: microATXПроизводитель процессора: IntelТип памяти: DDR4 DIMMПоддержка PCI Express: 3.0, 2.0Беспроводные интерфейсы: без Wi-FiМаксимальная частота памяти: 2400 МГцРазъемы на задней панели: VGA, PS/2 (клавиатура/мышь), LPT, DVI, COM-порт, HDMIПоддержка SLI:  нетПроизводитель:  ASUS |
| **Видеокарта** | | |
| Nvidia Geforce GTX1060 на 6GB |  | Объем видеопамяти: 6 ГБ  Тип памяти: GDDR5  Тип подключения: PCI Express 3.0  Разъемы и интерфейсы: выход HDMI, выход DVI, выход DisplayPort x 3  Разработчик видеокарты: NVIDIA  Линейка: GeForce  Название видеокарты: NVIDIA GeForce GTX 1060  Производитель: NVIDIA  Разрядность шины памяти: 192 бит  Тип HDMI: 2.0b |
| **Внутренняя звуковая карта** | | |
| ASIA PCI-E CMEDIA CMI8738 5.1 |  | Особенности: возможность вывода многоканального звукаРазрядность ЦАП: 16Поддержка ASIO: нетМаксимальная частота ЦАП (стерео): 44.1Производитель: C-media |
| **Устройство охлаждение(куллер)** | | |
| DeepCool THETA 21 PWM, 92мм, Ret |  | Бренд: DEEPCOOLМодель: THETA 21 PWMТип охлаждения: активное воздушноеКоличество вентиляторов: 1Размер вентилятора: 92 ммНаправление выдува: перпендикулярно МПВоздушный поток вентилятора:45.91 cfmУровень шума вентилятора: 18 - 33 дБСкорость вращения вентилятора:900 - 2400 об/минТип подшипника: скольжения (гидродинамический), Hydro Bearing |
| **Блок питания** | | |
| DEEPCOOL PK800D 800W, 80 PLUS Bronze, (R-PK800D-FA0B-EU) |  | Бренд**:** DEEPCOOL  Модель**:** PM800D  Форм-фактор**:** ATX  Версия ATX**:** 12V v2.4  Мощность**:** 800 Вт  АктивныйPFC**:** есть  Производительность(КПД) **:** 90 %  Сертифицирован встандарте**:** 80 PLUS GOLD  Цвет**:** черный |
| **Жесткий диск** | | |
| WD Purple WD10PURZ, 1ТБ |  | Емкость: 1 ТБФорм-фактор: 3.5"Назначение: для офисаТип: HDDСкорость вращения: 5400 об/минИнтерфейсы: SATA 6Gb/sОбъем буфера: 64 МБЛинейка: WD PurpleПроизводитель: Western Digital |
| **Корпус** | | |
| ATX Accord ACC-B301, Midi-Tower, |  | Типоразмер: Midi-TowerБлок питания: отсутствуетФорм-фактор материнской платы: ATXМаксимальная длина видеокарты: 300 ммПроизводитель: ACCORD |

Таблица №2 «Периферийные устройства»

|  |  |  |
| --- | --- | --- |
| **Монитор** | | |
| Монитор Acer K243YBMIX |  | Диагональ: 23.1"-24"  Макс. разрешение: 1920x1080  Тип матрицы экрана: IPS  Яркость: 250 кд/м2  Макс. частота обновления кадров: 61-99 Гц  Интерфейсы видео: вход VGA, вход HDMI  Соотношение сторон: 16:9  Особенности: встроенные колонки, подсветка без мерцания (Flicker-Free)  Изогнутый экран: Нет  Время отклика: 1 мс  Игровой монитор: Нет  Цвет: черный  Производитель: Acer |
| **Клавиатура** | | |
| HP 125 (266C9AA) |  | Форм фактор: MaxiТип: МембраннаяСпособ подключения: ПроводнаяИнтерфейс подключения: Type-AПодсветка: НетЦифровой блок: ЕстьПоддержка макросов: НетБренд: HPКоличество клавиш: 104 |
| **Мышь** | | |
| HP 125 (265A9AA) |  | Тип соединения мыши: проводнаяИнтерфейс подключения: USBПодключение по USB: естьДлина провода: 1.8 м |
| **Наушники** | | |
| Logitech Stereo Headset H150 |  | Тип наушников**:** [накладные](https://market.yandex.ru/catalog--kompiuternye-garnitury/26913190?hid=6368403&glfilter=21194330%3A36779387" \t "_blank)  Минимальная воспроизводимая частота**:** 20 Гц  Максимальная воспроизводимая частота**:** 20000 Гц  Импеданс**:** 22 Ом  Чувствительность**:** 122 дБ  Вес**:** 80 г  Тип микрофона**:** динамический  Крепление микрофона**:** подвижное  Импеданс микрофона**:** 22 Ом  Минимальная частота микрофона**:** 100 Гц |
| **Принтер** | | |
| HP Laser 135w (4ZB83A) |  | Область применения: малый офис  Цветность печати: черно-белая  Тип печати: лазерный  Максимальный формат: A4  Интерфейсы: Wi-Fi, USB, AirPrint  Функции сканера:  сканирование, копирование  Цвет: черный, белый  Производитель: HP |
| **Маршрутизатор** | | |
| Mikrotik RB3011UiAS-RM |  | Тип устройства:  маршрутизатор  Количество LAN-портов: 10  Базовая скорость передачи данных: 1 Гбит/с  Количество WAN-портов: 10  Особенности: межсетевой экран (Firewall), консольный порт, DHCP-сервер  Сетевые стандарты:  автоопределение MDI/MDIX  Производитель: MikroTik |
| **Сервер** | | |
| Quanta Computer 1S3AZZZ0ST5 |  | Тип корпуса: Rack  Количество установленных процессоров: 0  Размер установленной оперативной памяти: без ОЗУ  Количество слотов оперативной памяти: 4  Сокет процессора: LGA1151  Тип оперативной памяти: DDR4  Производитель: Quanta Computer |
| Коммутатор | | |
| D-Link DES-1024A/E1B |  | * Тип устройства: коммутатор * Количество LAN-портов: 24 * Базовая скорость передачи данных: 100 Мбит/с * Сетевые стандарты: автоопределение MDI/MDIX * Производитель: D-link |

## 2.3. Структура локальной сети предприятия

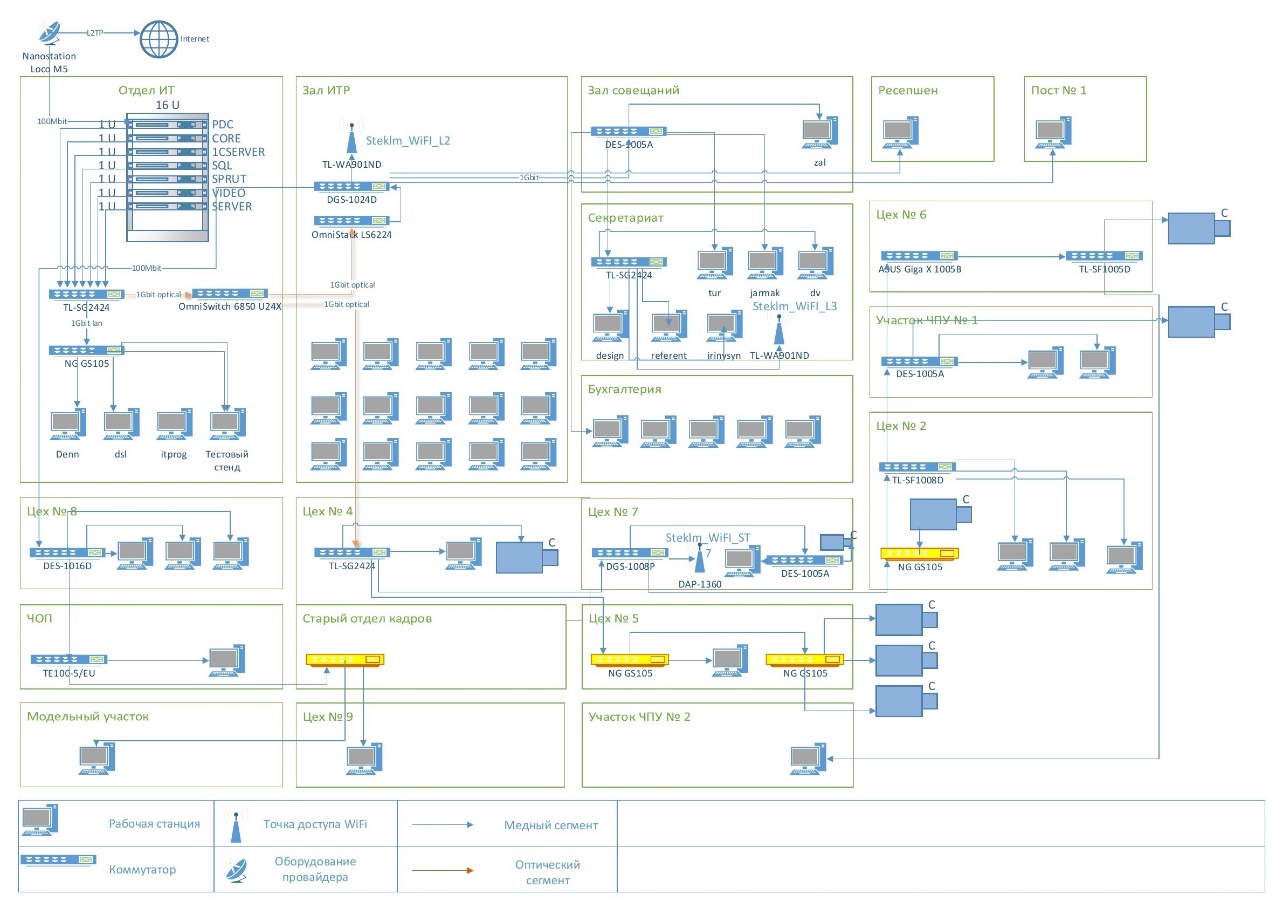


Рис. 2 «Структура локальной сети»

# 3. Выполнение индивидуального задания

## 3.1. Разработка технического задания

**Назначение разработки:**

Программа предназначена для упрощения и автоматизации работы со сбытом товара, 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 по сбыту товаров в базу данных и дальнейшей работы с этими данными.

Функциональные требования:

* Добавление, удаление и редактирование материалов, заказов, складов, движение, хранения, состав материала, сбыт.
* Поиск материалов, склада, сбыта, хранящихся материалов, пользователей по всем атрибутам
* Фильтрация материалов, склада, сбыта, хранящихся материалов, по всем
* Вывод на печать в Excel состава заказа
* Авторизация с помощью логина и пароля
* Регистрация пользователя

**Требования к организации входных данных:**

Входные данные программы организованы в БД в приложении «Microsoft SQL» со следующими полями: Пользователи – ФИО пользователя, логин пользователя, пароль пользователя, номер вида пользователя. Вид пользователя – название вида пользователя. Контрагенты – название контрагентов, ИНН, адрес контрагентов. Заказ – номер контрагента, номер пользователя, дата отправки, распределение. Состав заказа – номер заказа, номер материала, количество, цена. Материалы – название материала, номер чертежа, номер типа материала. Тип материала – название типа материала. Иерархия – номер родителя, номер дочернего, количество. Склад– название склада, адрес. Хранение – номер склада, номер материала, количество. Движение – номер склада, номер состава заказа, номер пользователя, количество, приход или сбыт.

**Требования к организации выходных данных:**

Выходные данные организованы в виде таблицы и вывода на печать через Excel состава заказа.

**Требования к надёжности:**

Программа должна быть в достаточной степени надёжна от сбоев. На крайний случай предусмотрено сохранение данных БД в приложении «Microsoft SQL» или восстановление данных в случае завершения работы.

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

Таблица №3 «Системные требования»

|  |  |
| --- | --- |
| Процессор | AMD Ryzen 5 5600G, SocketAM4 |
| Оперативная Память | DIMM Kingston KVR26N19D8/16 |
| Разрешение экрана | 1920x1080 |
| Размер экрана | 24 дюйма |
| Устройства ввода | Мышь, клавиатура |
| Дисковое пространство | 1Гб |
| Подключаемые устройства | Принтер |
| Операционная система | Windows 10, 11 |

**Требования к информационной и программной совместимости:**

Для корректной работы программы необходимо:

ОС Windows 10/11 – 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 - программа для работы с электронными таблицами, созданная корпорацией Microsoft для Microsoft Windows, Windows NT и Mac OS, а также Android, iOS и Windows Phone. Она предоставляет возможности экономико-статистических расчетов, графические инструменты и, за исключением Excel 2008 под Mac OS X, язык макропрограммирования VBA. Microsoft Excel входит в состав Microsoft Office.

**Условия эксплуатации:**

Программа не требует специального обслуживания. Для ознакомления с полным функционалом пользователь должен прочесть Руководство пользователя. Для работы с программой требуются хотя бы малейшие навыки работы с приложениями с похожим интерфейсом, содержащими информацию в виде таблиц БД и Excel.

**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 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

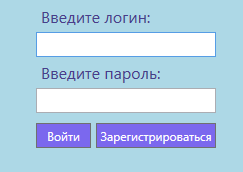


Рис. 3 «Страница авторизации»

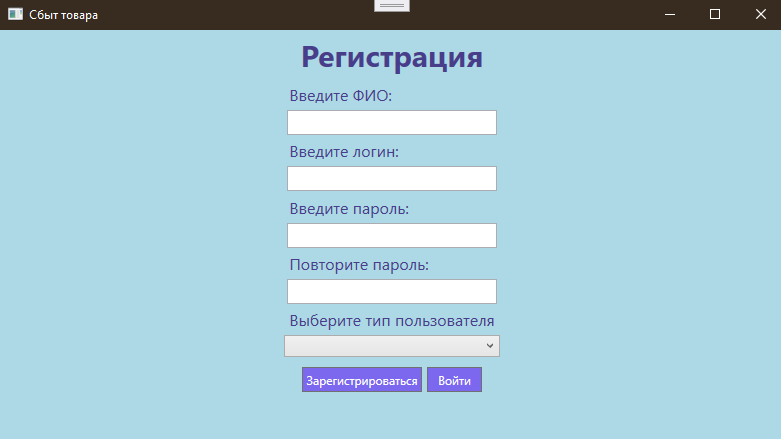


Рис. 4 «Страница регистрации»



Рис. 5 «Главная страница»

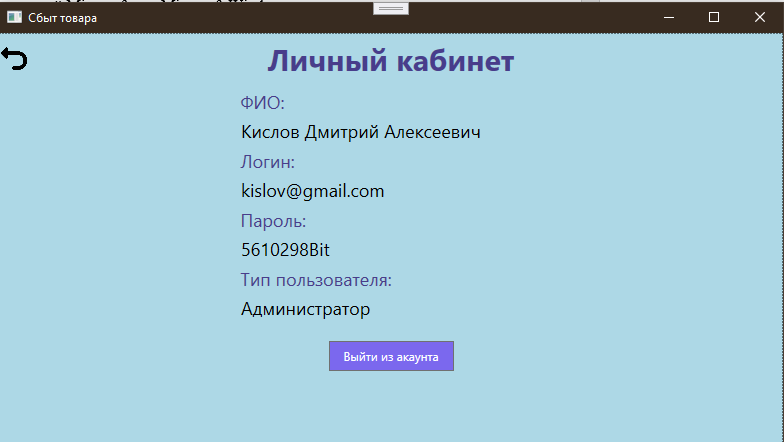


Рис. 6 «Страница c личным кабинетом»

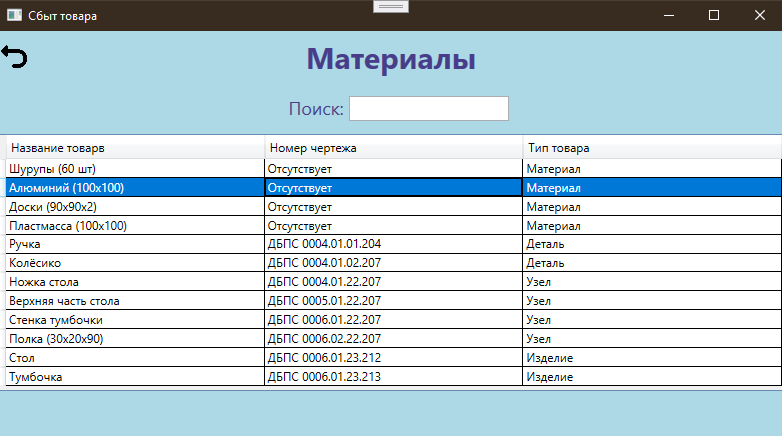


Рис. 7 «Страница материалов»

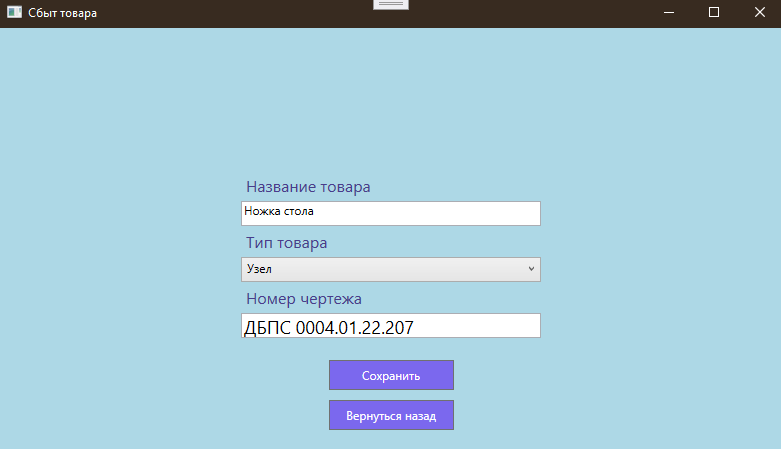


Рис. 8 «Страница добавления и редактирования материалов»

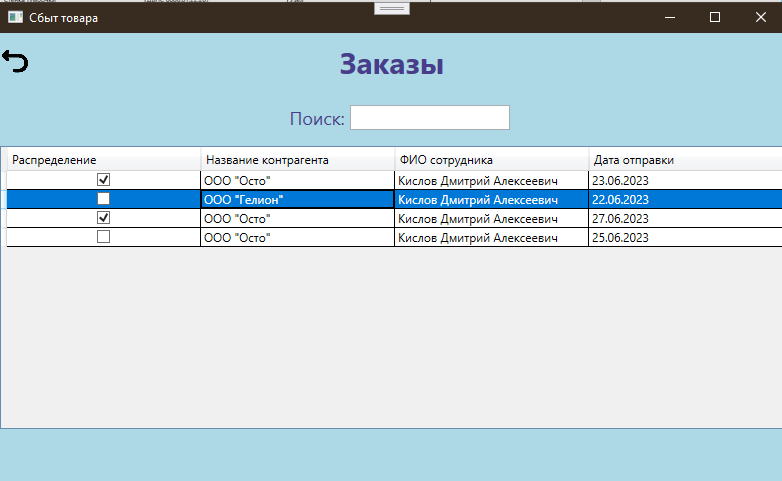


Рис. 9 «Страница заказов»

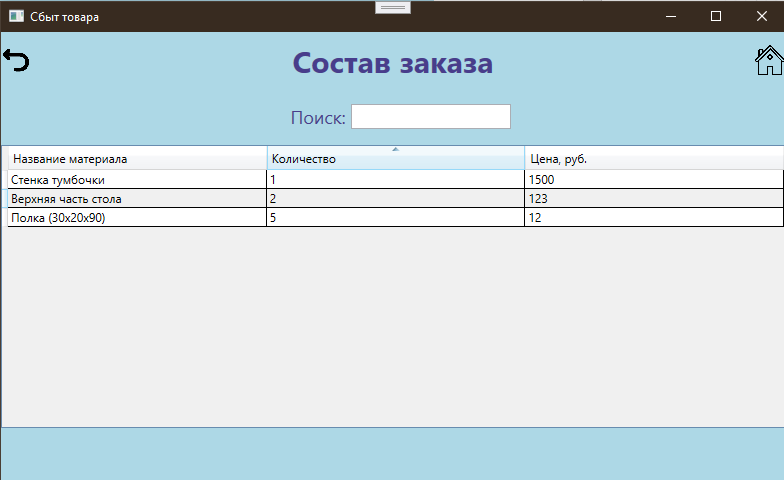


Рис. 10 «Страница состава заказа»

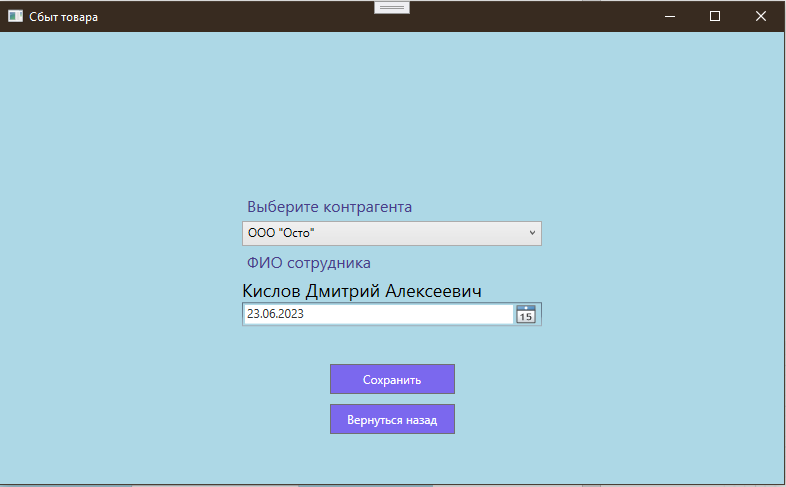


Рис. 11 «Страница добавления и редактирования заказов»

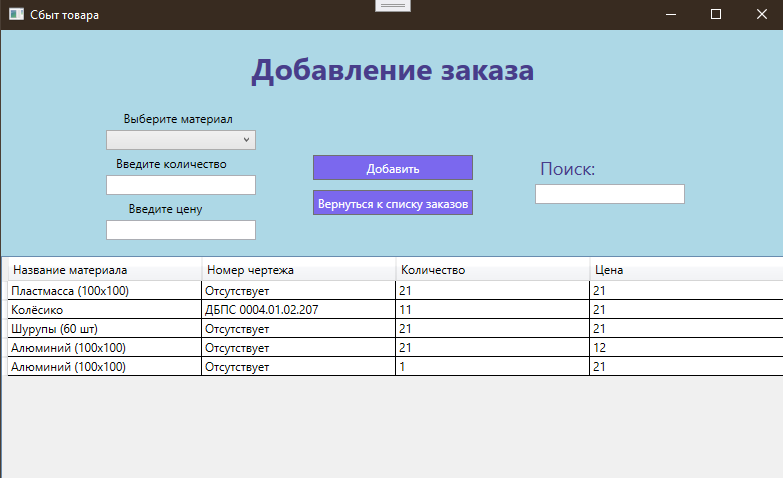


Рис. 12 «Страница добавления и редактирования состава заказов»

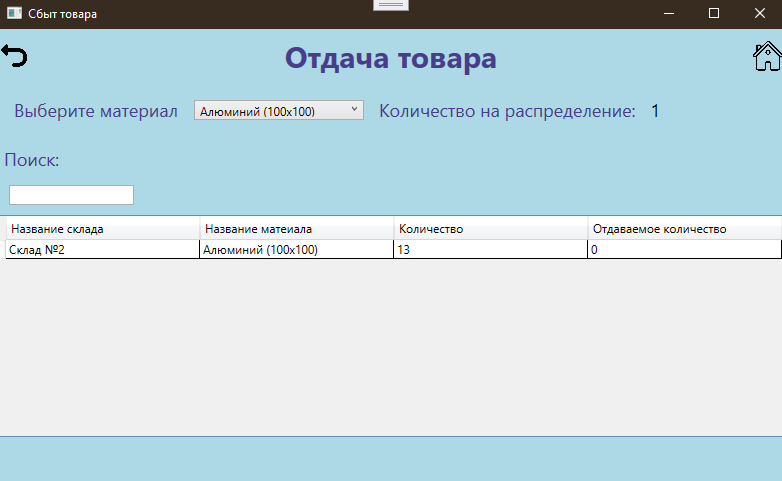


Рис. 13 «Страница отдачи товара»

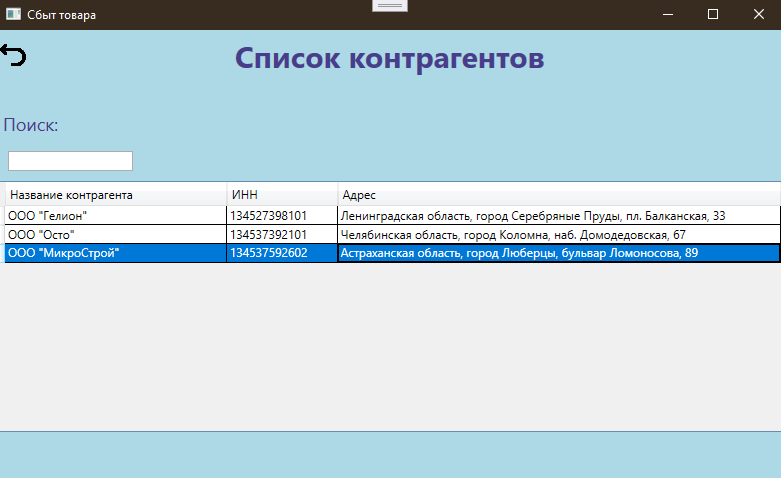


Рис. 14 «Страница контрагентов»

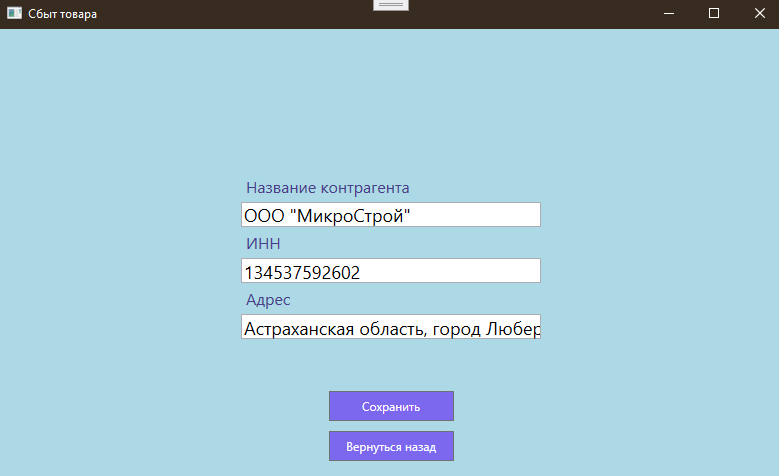


Рис. 15 «Страница добавления и редактирования контрагентов»

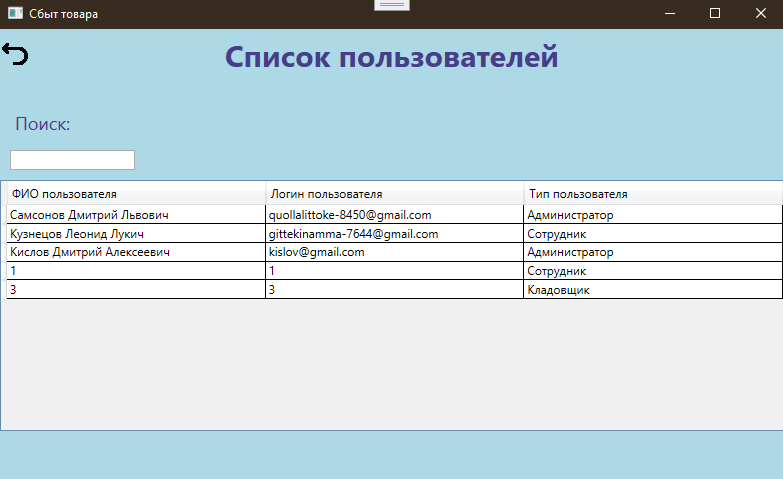


Рис. 16 «Страница пользователей»

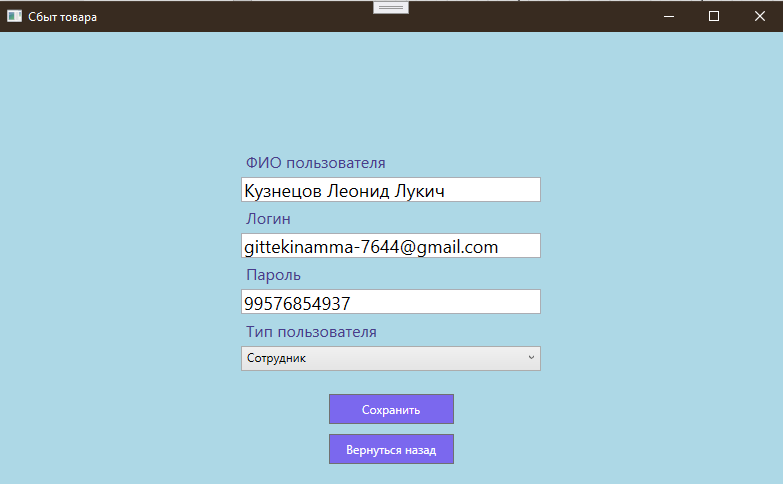


Рис. 17 «Страница добавления и редактирования пользователей»

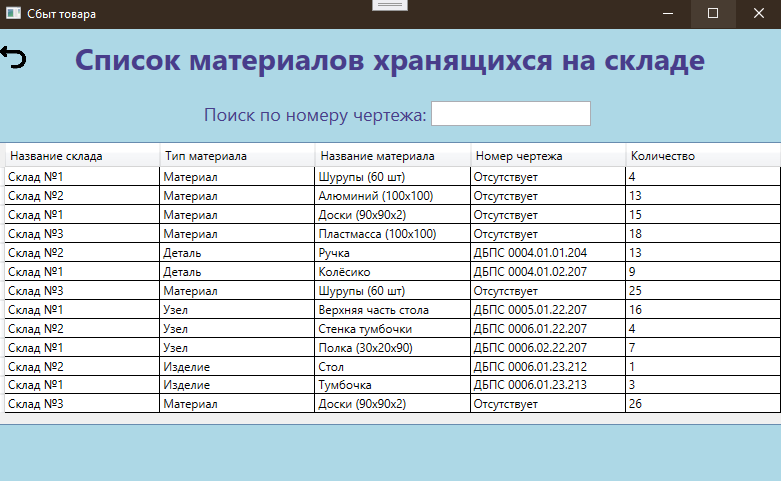


Рис. 18 «Страница материалов со склада»



Рис. 19 «Страница сбыта товара»

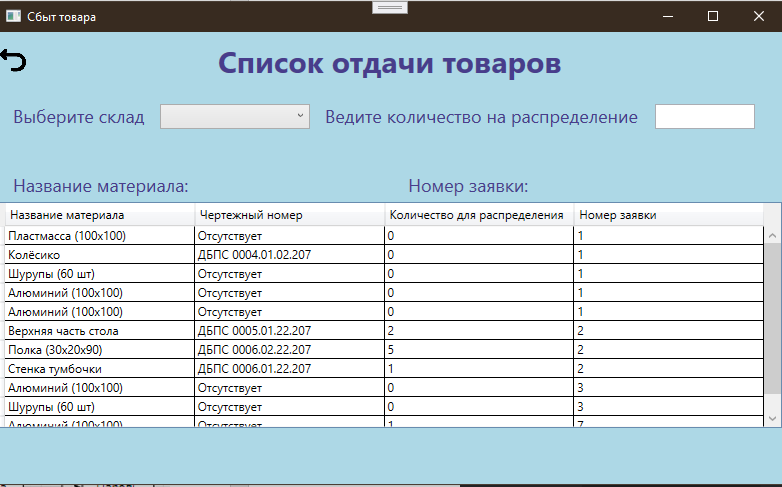


Рис. 20 «Страница отдачи товара»

## 3.3. Блок-схема программы

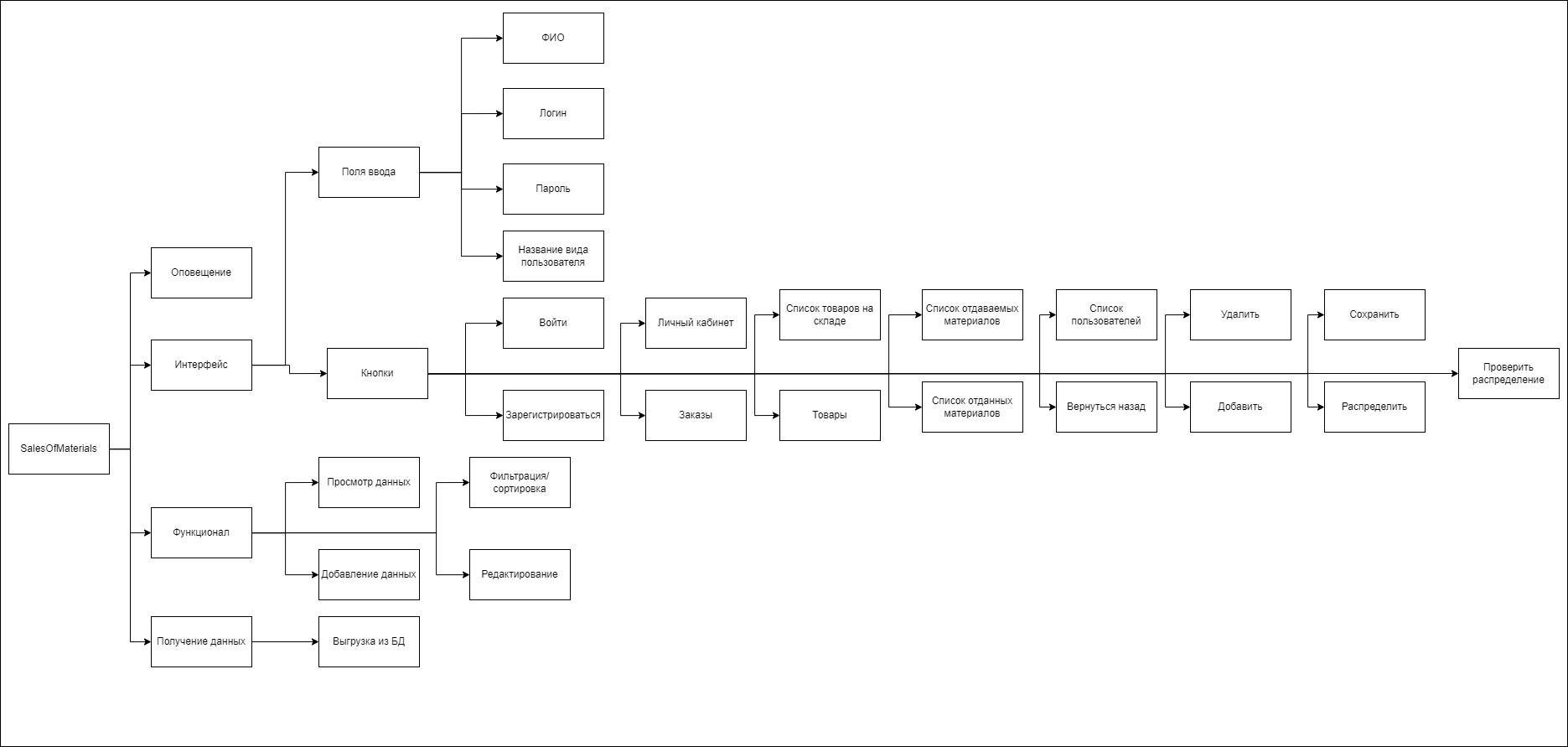


Рис. 21 «Функциональная схема программы»

## 3.4. Руководство программиста

Приложение разработано в Visual Studio 2019

Создание проекта:

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

Рис. 22 «Создание проекта»

Для корректной работы приложения необходимо подключить БД:

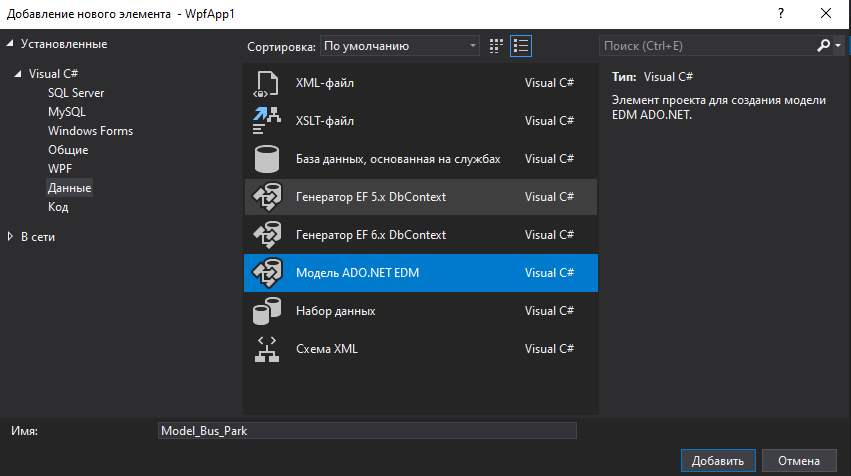


Рис. 23 «Создание модели ADO.NET»

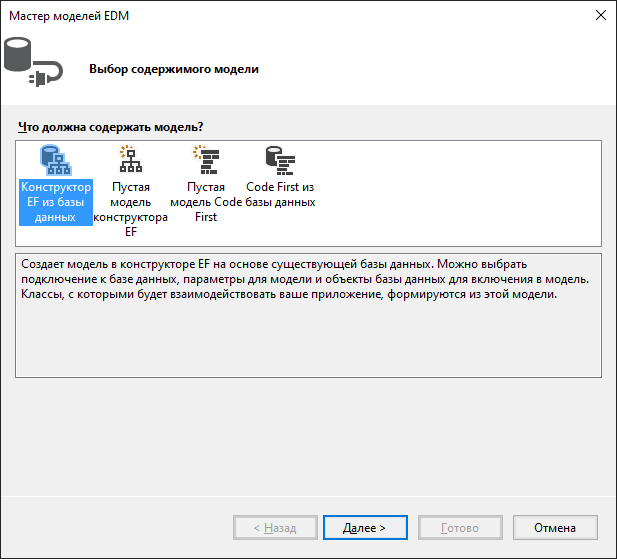


Рис. 24 «Выбор содержимого модели»

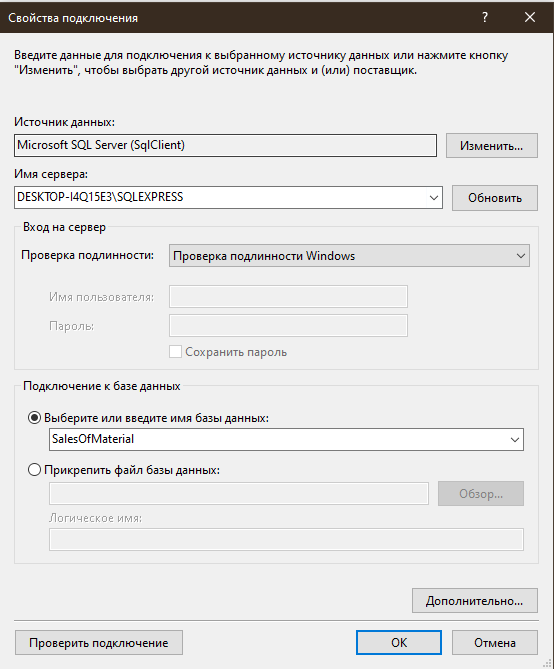


Рис. 25 «Выбор сервера и имени БД»

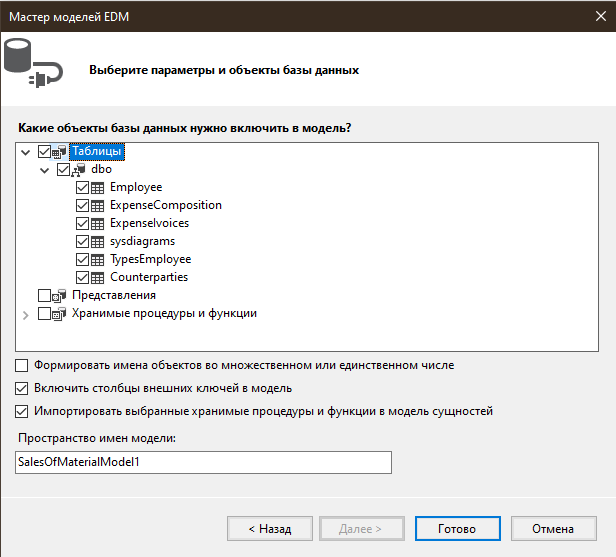


Рис. 26 «Выбор объектов БД»

Диаграмма БД в Visual Studio:

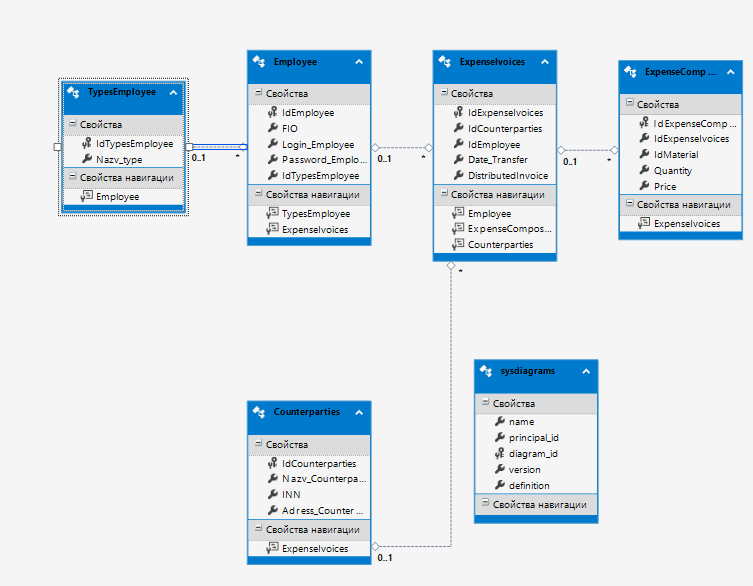


Рис. 33 «Модель данных»

Структура приложения в обозревателе решений:

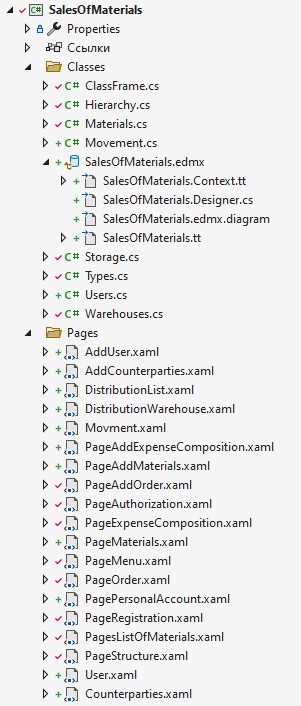


Рис. 34 «Структура приложения»

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

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Controls;

namespace SalesOfMaterials.Classes

{

public class ClassFrame

{

public static Frame frmObj;

public static SalesOfMaterialEntities db;

}

}

**Входные и выходные данные**

Таблица №4 «Таблица данных»

|  |  |
| --- | --- |
| **Поля** | **Тип данных** |
|  |  |
| ФИО пользователя | Текстовый |
| Логин пользователя | Текстовый |
|  |  |
| Пароль пользователя | Текстовый |
| Тип пользователя | Текстовый |
| Дата отправки | Дата |
| Количество с заказа | Числовой |
| Цена заказа | Числовой |
| Название контрагента | Текстовый |
| ИНН контрагента | Текстовый |
| Адрес контрагента | Текстовый |
| Название материала | Текстовый |
| Номер чертежа | Текстовый |
| Тип материала | Текстовый |
| Количество материала | Числовой |
| Название склада | Текстовый |
| Адрес склада | Текстовый |
| Количество на складе | Числовой |
| Отправленное количество | Числовой |

**Выходная информация:**

Выходной информацией является экспорт списка/таблицы в виде нового Excel-файла.

## 3.5. Руководство пользователя

При открытии программы ввести логин и пароль, привязанные к аккаунту. Исходя из роли в учётной записи (Администратора, Сотрудника, Кладовщика), осуществится переход на определённую страницу. Администратор появляются все кнопки. Сотрудник видит все кнопки кроме: «Распределения», «Список отдаваемых материалов», «Список отданных материалов», «Список пользователей». Кладовщик видит все кнопки кроме: «Добавить новый заказ», «Список пользователей». В случае, если данные для авторизации введены неверно, появляется уведомление. Если аккаунта ещё нет, то необходима нажать на кнопку «Зарегистрироваться».

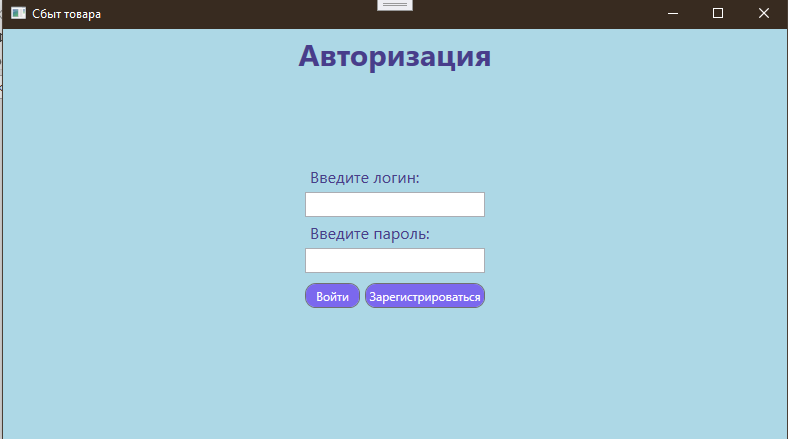


Рис. 35 «Страница авторизации»

На странице регистрацию нужно заполнить поля «ФИО», «Логин», «Пароль», «Повторить пароль», «Тип пользователя». После того как все данные в полях заполнены нужно нажать на кнопку «Зарегистрироваться».

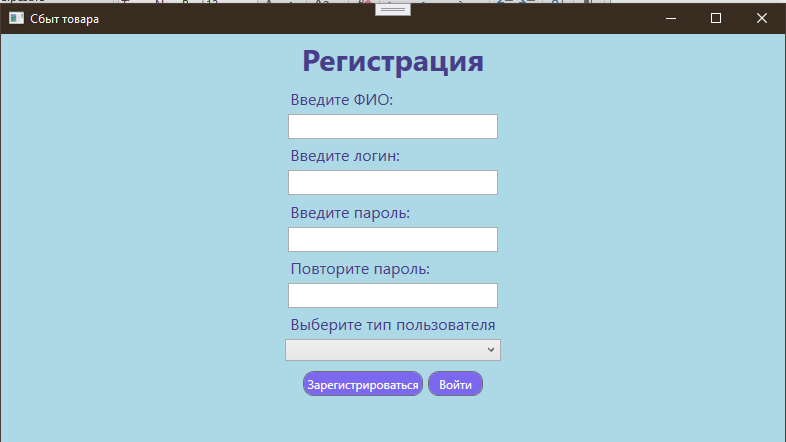


Рис. 36 «Страница регистрации»

После авторизации (исполняемый «.exe» файл) появится страница «Меню» с возможностью перехода на страницы «Личный кабинет», «Заказы», «Список товаров на складе», «Товары», «Список отдаваемых материалов», «Список отданных материалов» и «Список пользователей».



Рис. 37 «Главная страница»

После нажатия на одну из кнопок в навигационном меню пользователь попадает на страницу, на которой есть список с заполненными данными. Для возвращения на предыдущую страницу с переходами нужно нажать на кнопку «Вернуться назад», расположенную в нижней части страницы. Элементы управления для взаимодействия с информацией располагаются в нижней части страницы, а также функции для добавления, редактирования и удаления информации из таблицы/списка.

При нажатии в навигационном меню на кнопку «Заказы» откроется страница с соответствующим списком заказов.

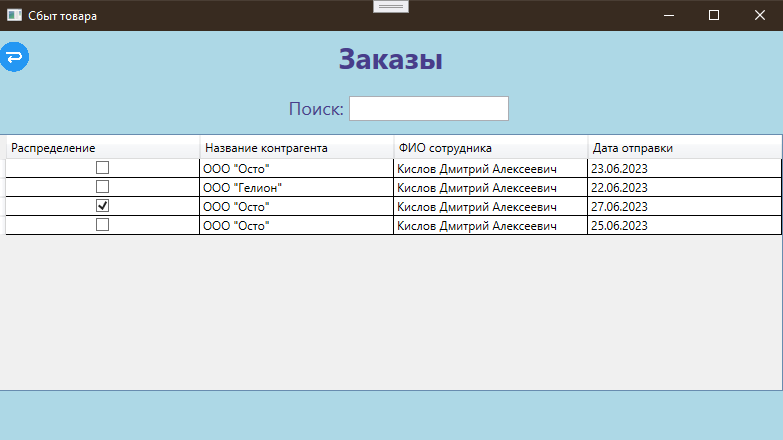


Рис. 38 «Страница заказов»

При добавлении информации пользователю необходимо нажать ПКМ по таблице и выбрать из выпадающего списка «Добавить новый заказ». Затем появится страница для добавления с пустыми значениями, которые потребуется заполнить. Обязательные поля: название контрагента, ФИО сотрудника, Дата отправки.

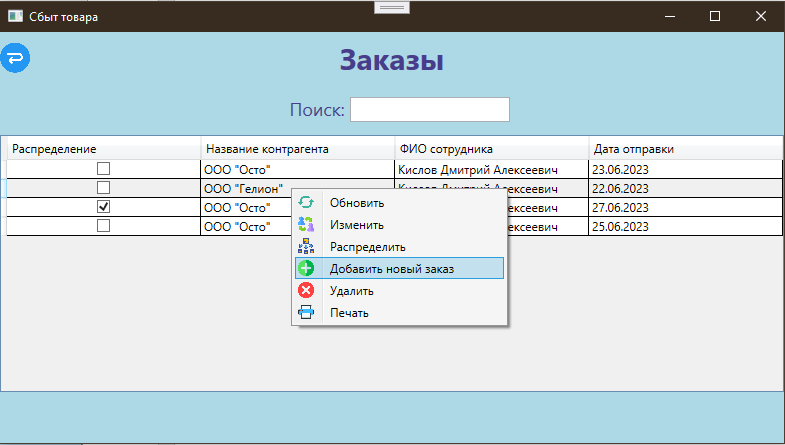


Рис. 39 «Страница заказов с вызовом добавления»

Заполним страницу добавления данными.

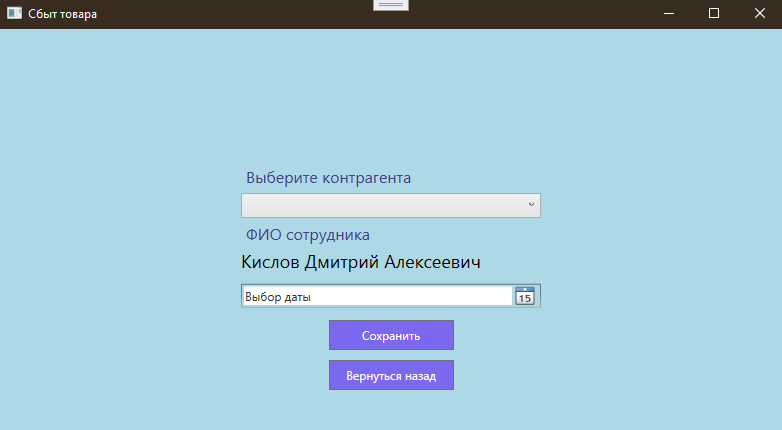


Рис. 40 «Страница добавления заказов»

После нажатия кнопки «Сохранить» пользователь переходит на страницу добавления состава заказа.

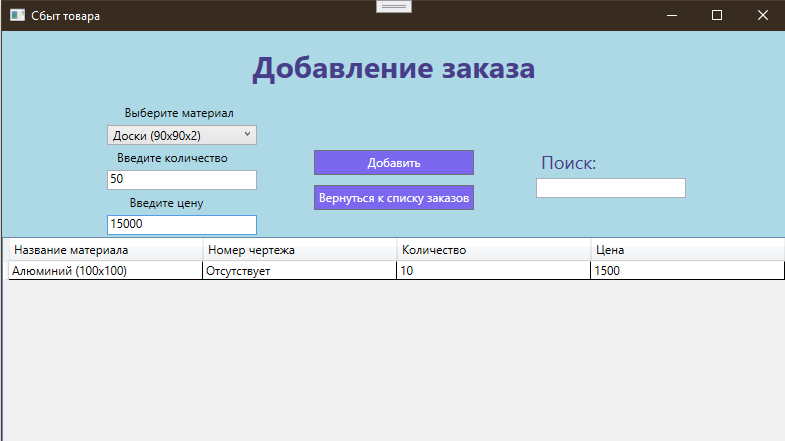


Рис. 41 «Страница добавления состава товара»

Для добавление нужных товаров необходимо выбрать материал, количество и цену и после нажать на кнопку «Добавить». Затем, когда нужный товар добавлен нужно нажать на кнопку «Вернуться к списку заказов»

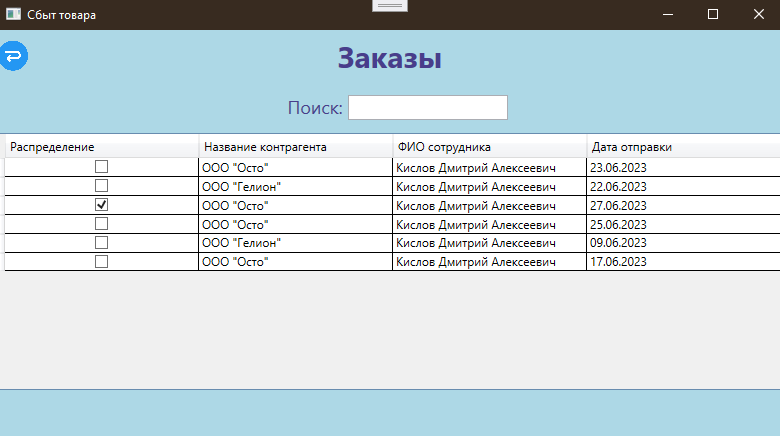


Рис. 42 «Страница заказы с добавленной записью»

При изменении информации пользователю необходимо нажать ПКМ по таблице и выбрать из выпадающего списка «Изменить». Затем появится страница для изменения с заполненными значениями, которые потребуется изменить. Обязательные поля: название контрагента, ФИО сотрудника, Дата отправки.

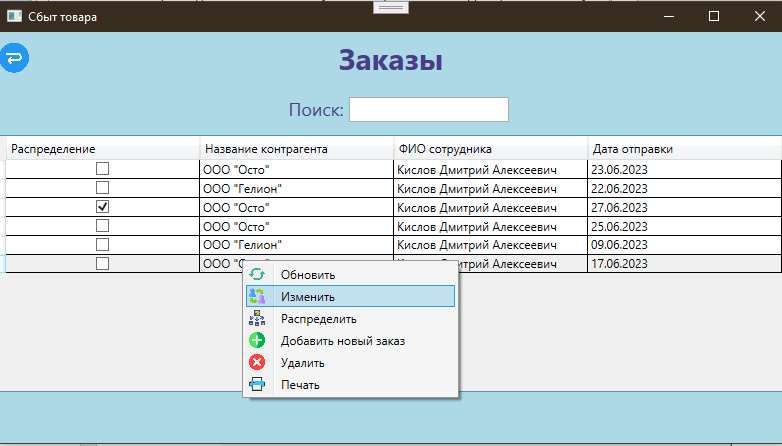


Рис. 43 «Страница заказов с вызовом изменения»

Изменим данные на странице редактирование.

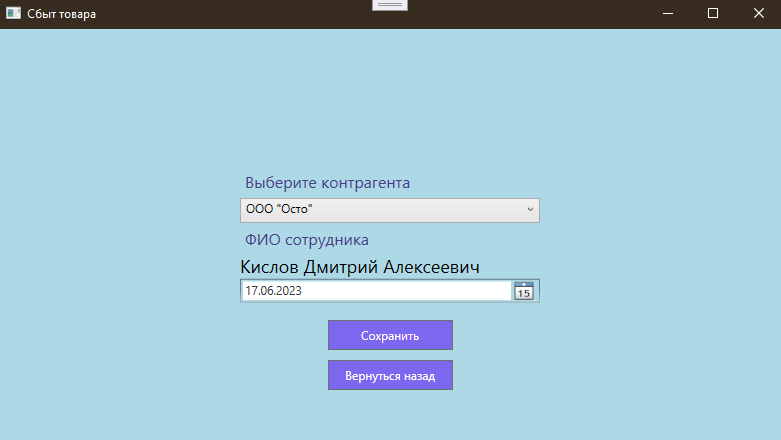


Рис. 44 «Страница редактирование заказов»

После нажатия кнопки «Сохранить» пользователь переходит на страницу добавления состава заказа.

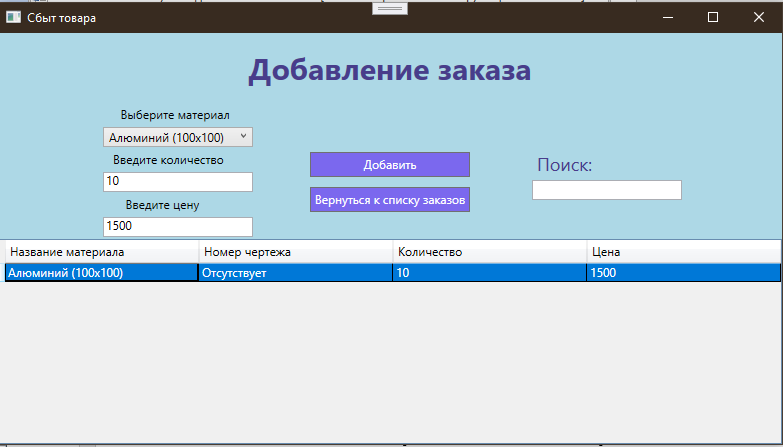


Рис. 45 «Страница редактирование состава заказа»

Для редактирования нужных товаров необходимо нажать двойным щелчком на нужный материал из таблицы и изменить данные. Затем, когда нужные товары изменены нужно нажать на кнопку «Вернуться к списку заказов»

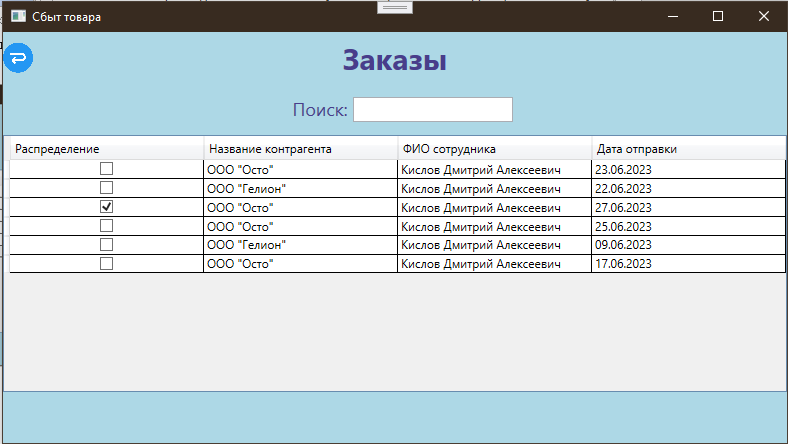


Рис. 46 «Страница заказы после редактирования»

Для просмотра состава заказа необходимо выбрать заказ и с помощью двойного щелчка по мыши прейдем на страницу.

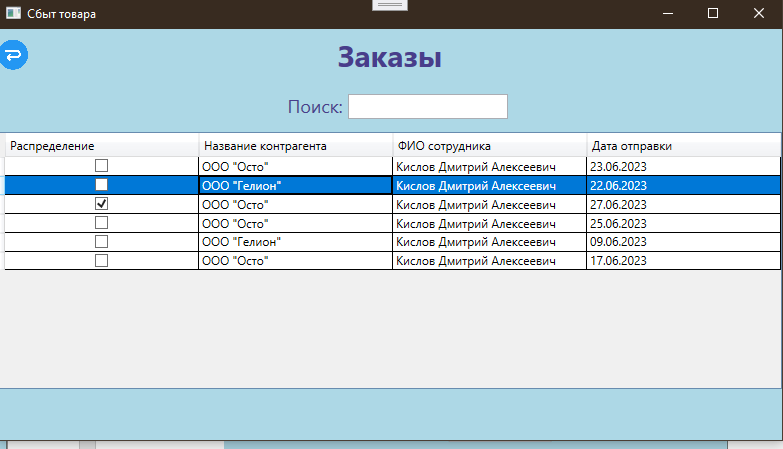


Рис. 47 «Страница заказы переход на состав заказа»

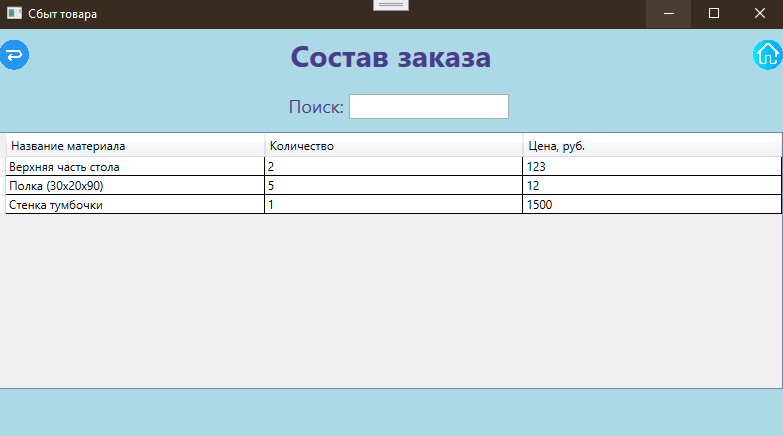


Рис. 48 «Страница состав товара»

При удалении информации пользователю необходимо нажать ПКМ и из списка выбрать «Удалить». Затем появится уведомление «Удалить состав заказа».

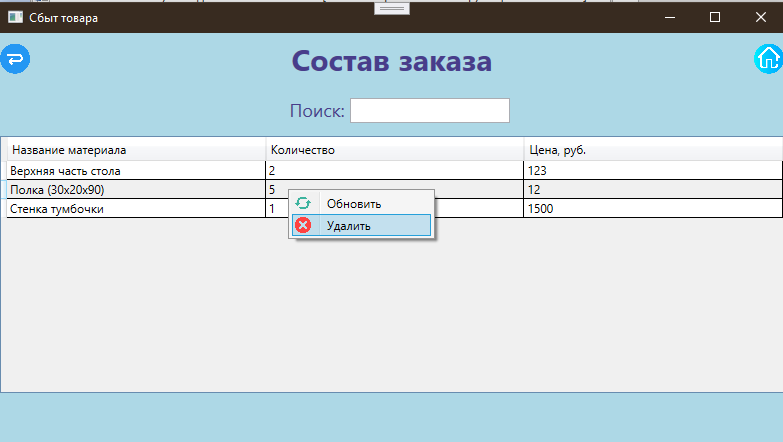


Рис. 49 «Страница состава заказов с удалением записей»

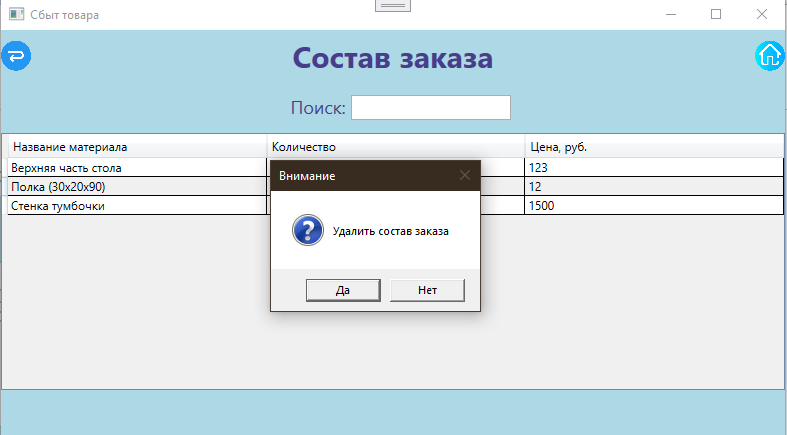


Рис. 50 «Страница состава заказов с уведомление на удаление»

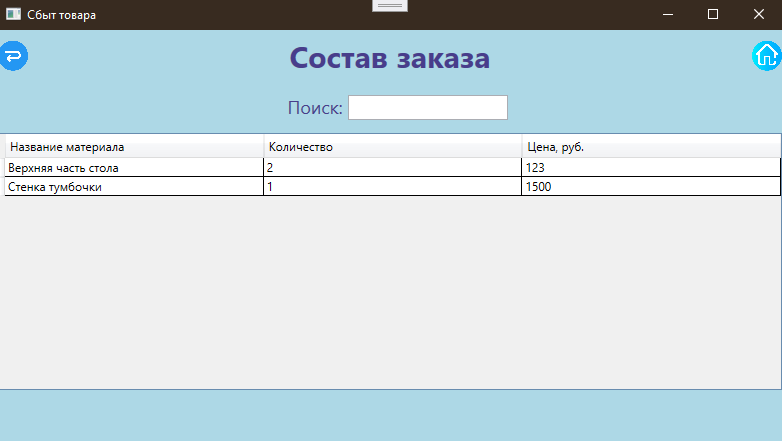


Рис. 51 «Страница состава после удаления»

В приложении присутствует поиск для нахождения данных по атрибутам таблицы. В него нужно ввести текст, которые присутствуют в названии контрагента или ФИО сотрудника.



Рис. 52 «Страница заказов с поиском»

Чтобы убрать результат операций, пользователю нужно удалить данные из поиска.

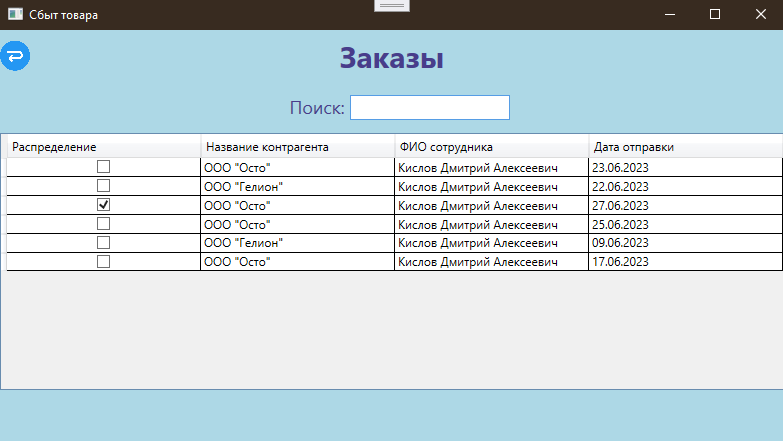


Рис. 53 «Страница заказов после сброса данных»

При удалении информации пользователю необходимо нажать ПКМ и из выпадающего списка выбрать «Удалить». Затем появится уведомление «Удалить заказ».

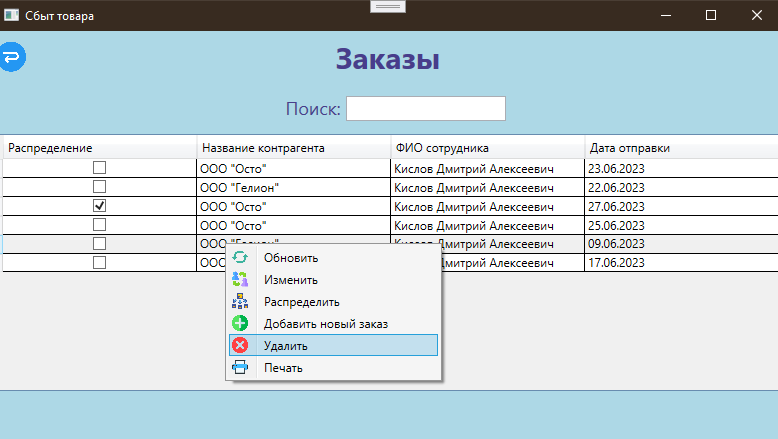


Рис. 54 «Страница заказов с удалением записей»

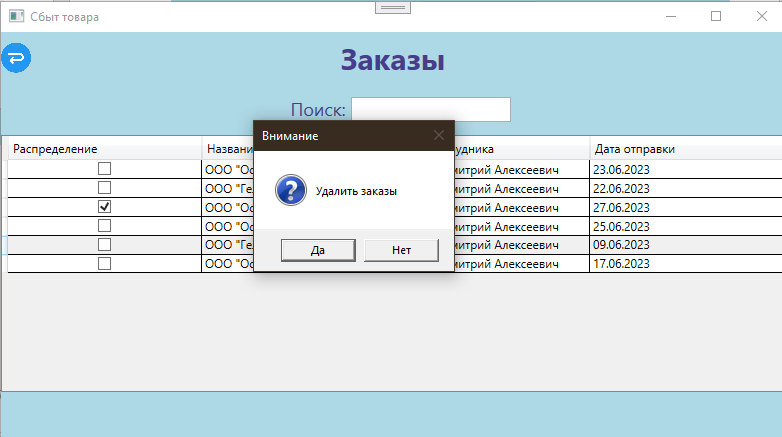


Рис. 55 «Страница заказов с уведомление на удаление»

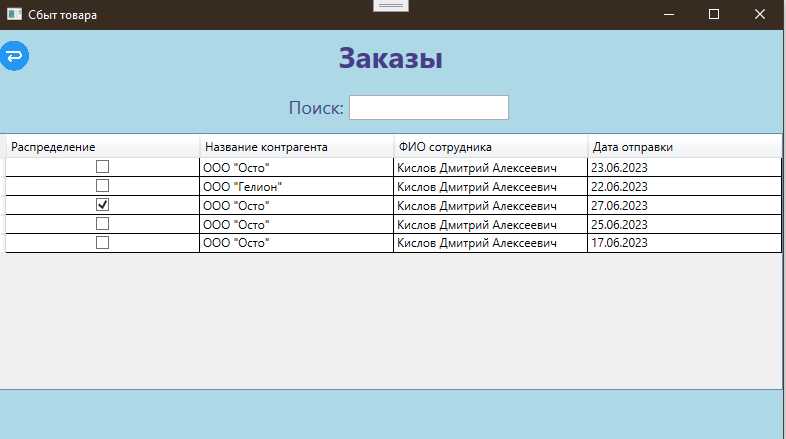


Рис. 56 «Страница заказов после удаления»

При распределении товара пользователю необходимо выбрать заказ и нажать ПКМ и выбрать из списка «Распределить». Затем появится страница для распределения с пустыми значениями, которые потребуется заполнить. Обязательные поля: название материала, отдаваемое количество.

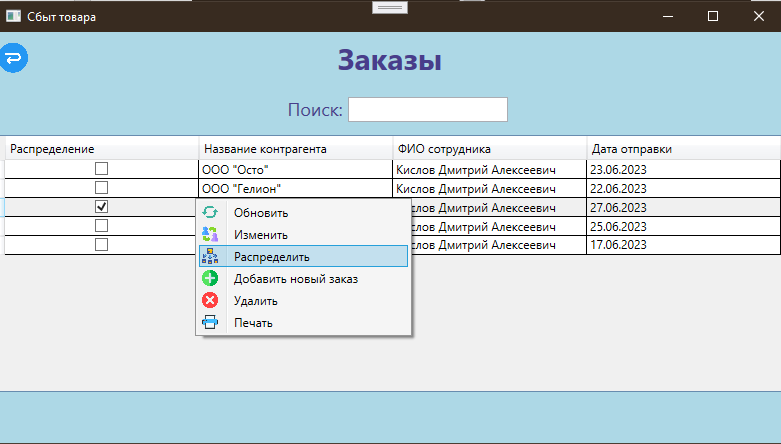


Рис. 57 «Страница заказов с распределением товара»

На странице «Отдача товаров» пользователю нужно ввести количество отдаваемого товара в таблицу. Затем после ввода нужного количества необходимо нажать на ПКМ и выбрать «Распределить». После, когда все товары распределены нужно нажать на ПКМ и выбрать «Проверить распределение».

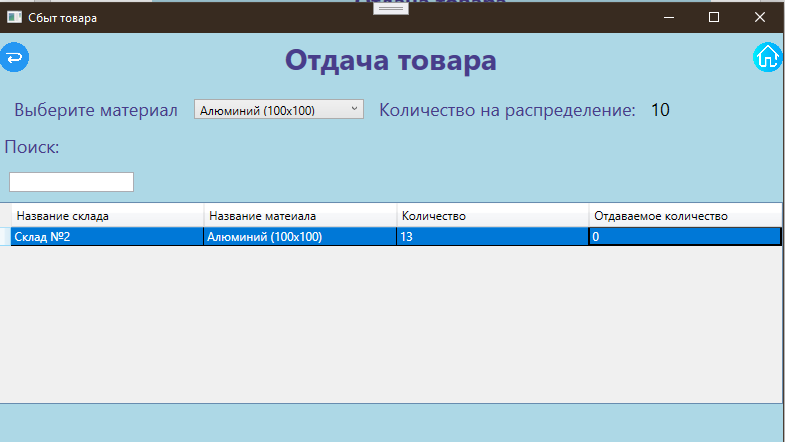


Рис. 58 «Страница отдача товара»

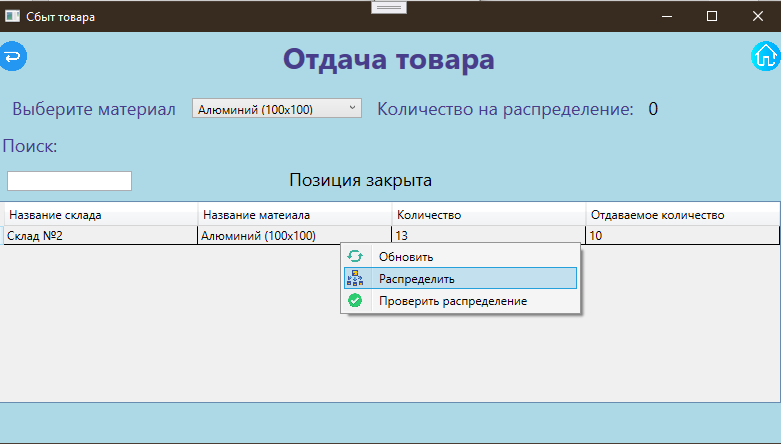


Рис. 59 «Страница отдача товаров распределение»



Рис. 60 «Страница отдача товаров уведомление на распределение»

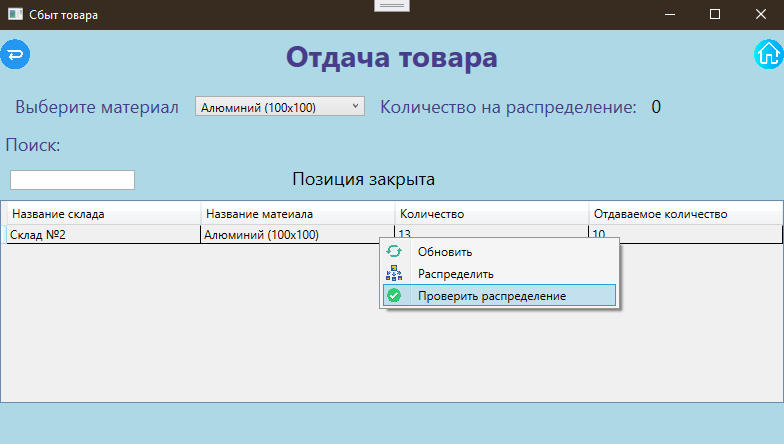


Рис. 61 «Страница отдача товаров проверка распределение»

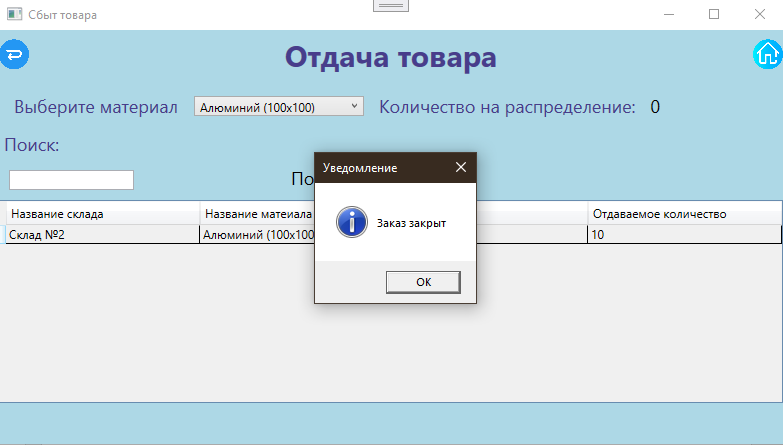


Рис. 62 «Страница отдача товаров уведомление на проверку распределение»

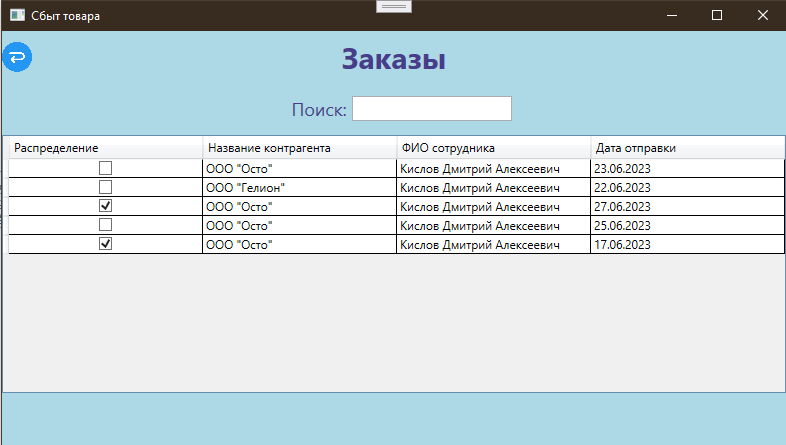


Рис. 63 «Страница заказов после распределения»

При нажатии в навигационном меню на кнопку «Личный кабинет» откроется страница с соответствующими данными о пользователе.

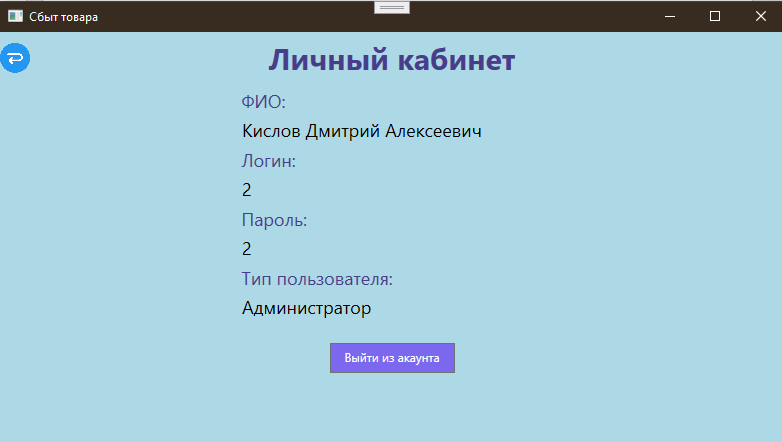


Рис. 64 «Страница с личным кабинетом»

Для редактирования личных данных пользователя необходимо нажать двойным щелчком по полю. Если данные введены верно, то нужно нажать Enter для сохранения данных. Если данные не нужно было редактировать, то нужно нажать Ctrl для отмены редактирования.

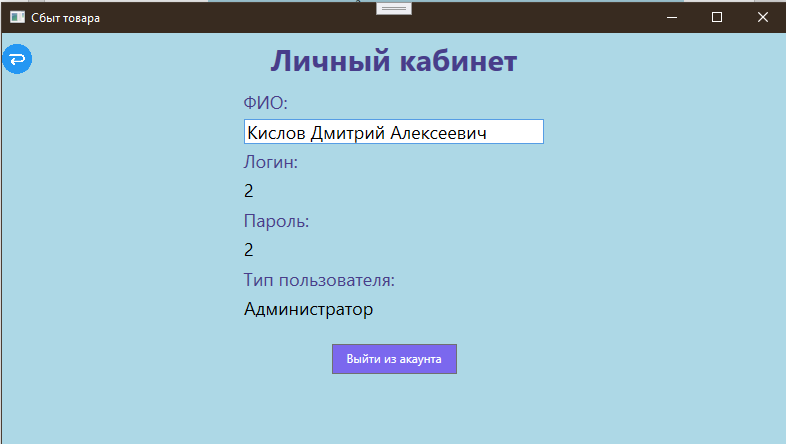


Рис. 65 «Страница редактирование личного кабинета»

На данной странице так же можно выйти из аккаунта.

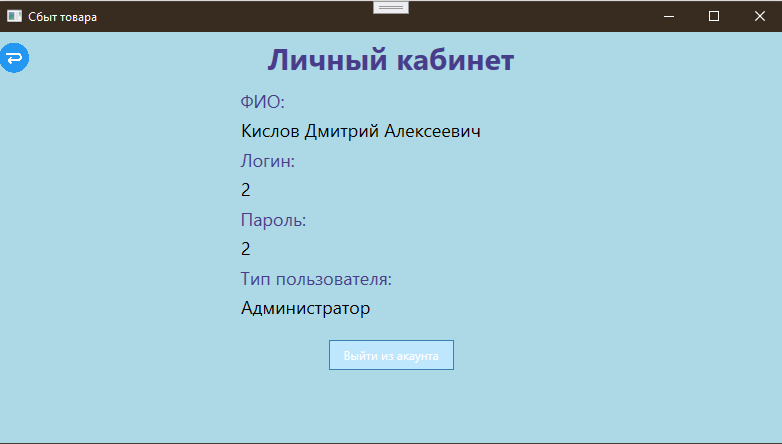


Рис. 66 «Выход из личного кабинета»

При нажатии в навигационном меню на кнопку «Товары» откроется страница с соответствующим списком товаров.

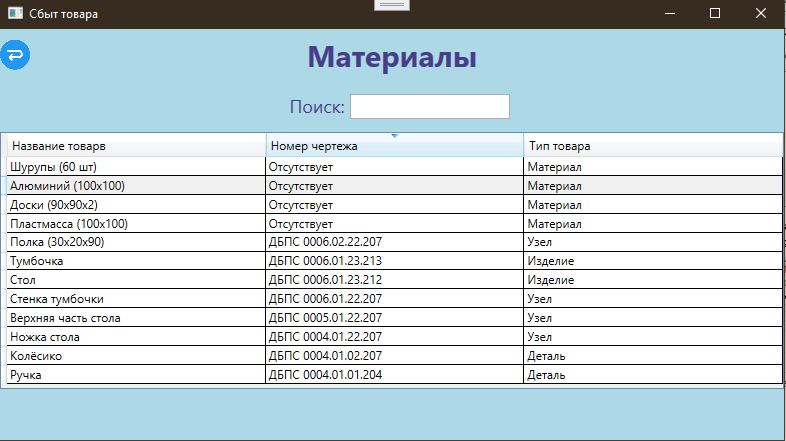


Рис. 67 «Страница материалов»

При добавлении информации пользователю необходимо нажать ПКМ по таблице и выбрать из выпадающего списка «Добавить». Затем появится страница для добавления с пустыми значениями, которые потребуется заполнить. Обязательные поля: Название товара, Тип товара, Номер чертежа.

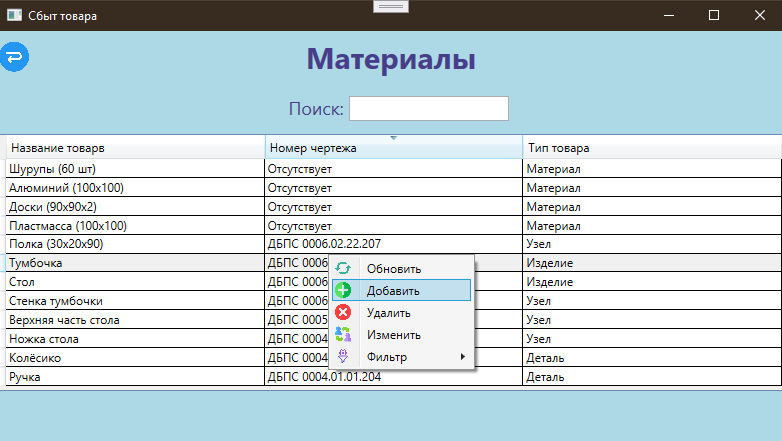


Рис. 68 «Страница материалов с вызовом добавления»

Заполним страницу добавления данными.

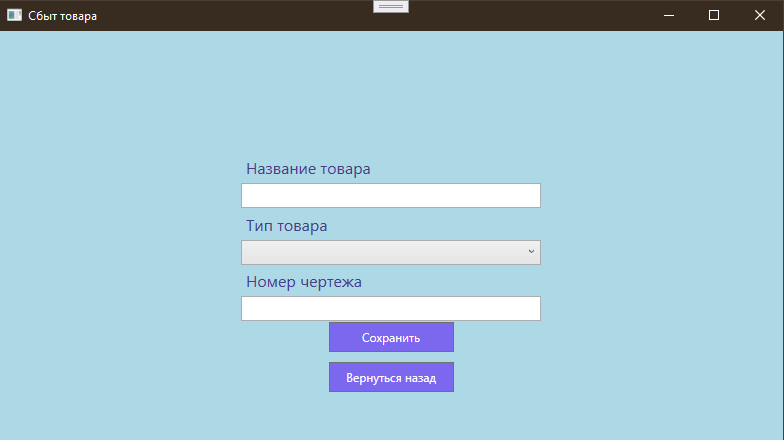


Рис. 69 «Страница добавления материалов»

После нажатия кнопки «Сохранить» пользователь возвращается на страницу материалов.

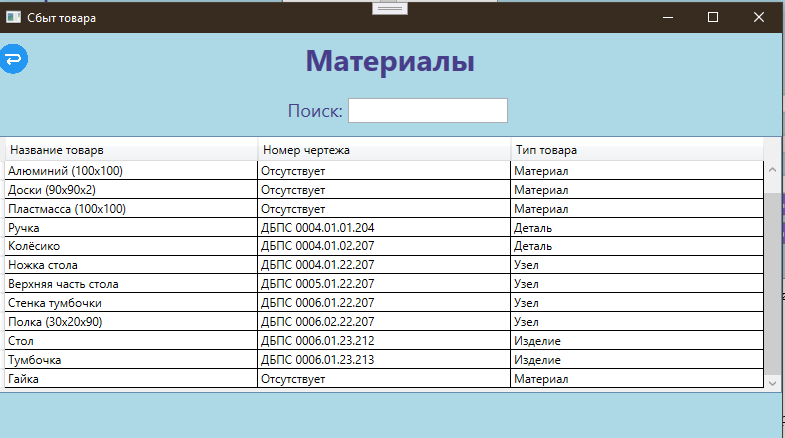


Рис. 70 «Страница материалов после добавления»

При изменении информации пользователю необходимо нажать ПКМ по таблице и выбрать из выпадающего списка «Изменить». Затем появится страница для изменения с заполненными значениями, которые потребуется изменить. Обязательные поля: название контрагента, ФИО сотрудника, Дата отправки.

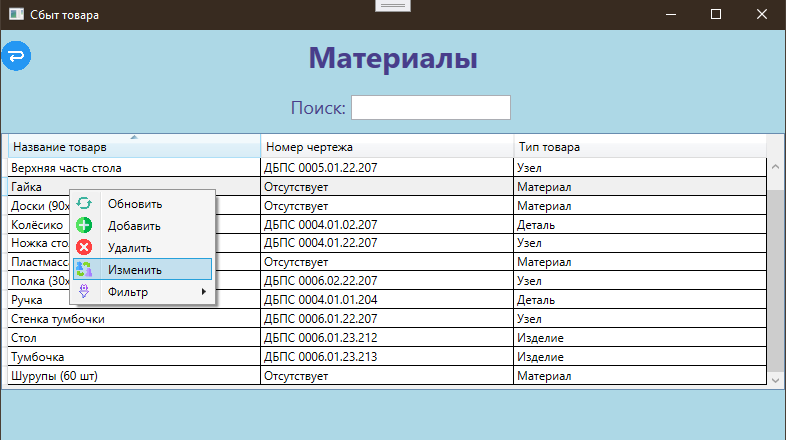


Рис. 71 «Страница материалов с вызовом изменения»

Изменим данные на странице редактирование.

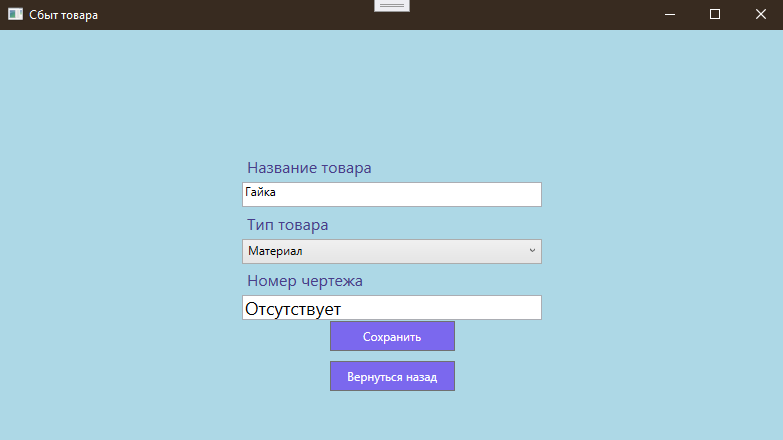


Рис. 72 «Страница редактирование материалов»

После нажатия кнопки «Сохранить» пользователь возвращается на страницу материалов.

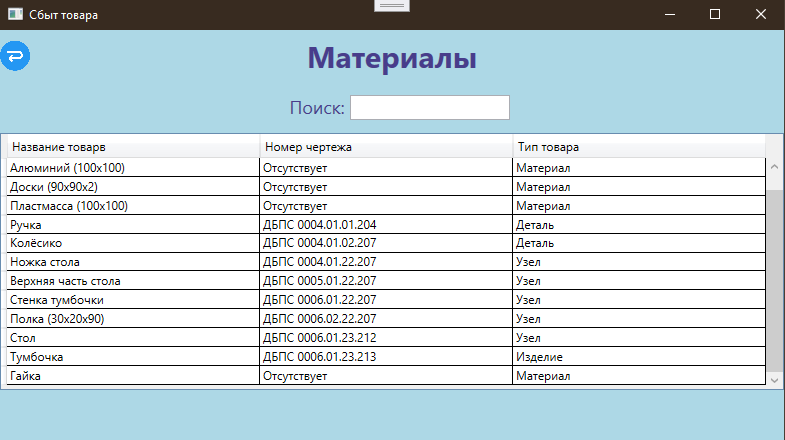


Рис. 73 «Страница материалов после редактирования»

В приложении присутствует поиск для нахождения данных по атрибутам таблицы. В него нужно ввести текст, которые присутствуют в названии материала, номер чертежа, тип товара.

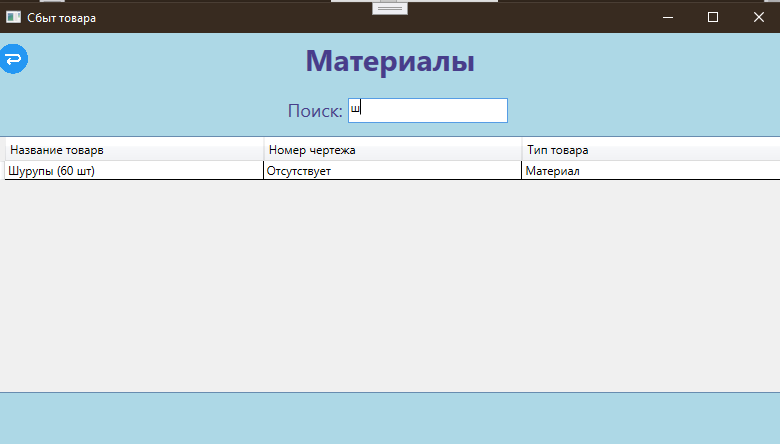


Рис. 74 «Страница материалов с поиском»

Чтобы убрать результат операций, пользователю нужно удалить данные из поиска.

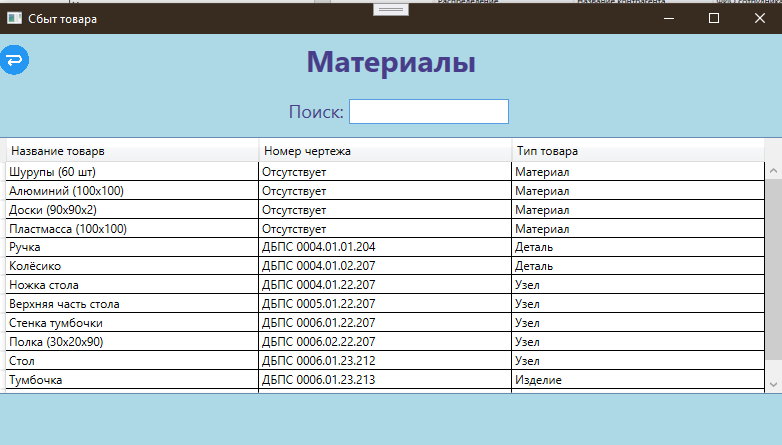


Рис. 75 «Страница материалов после сброса данных»

При удалении информации пользователю необходимо нажать ПКМ и из выпадающего списка выбрать «Удалить». Затем появится уведомление «Удалить заказ».

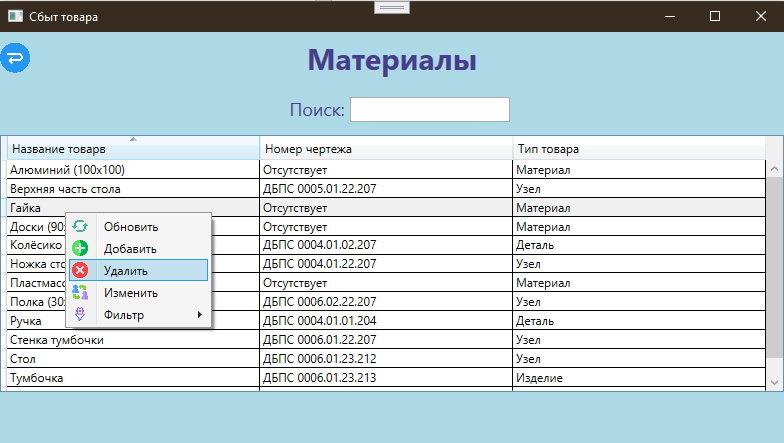


Рис. 76 «Страница материалов с удалением записей»

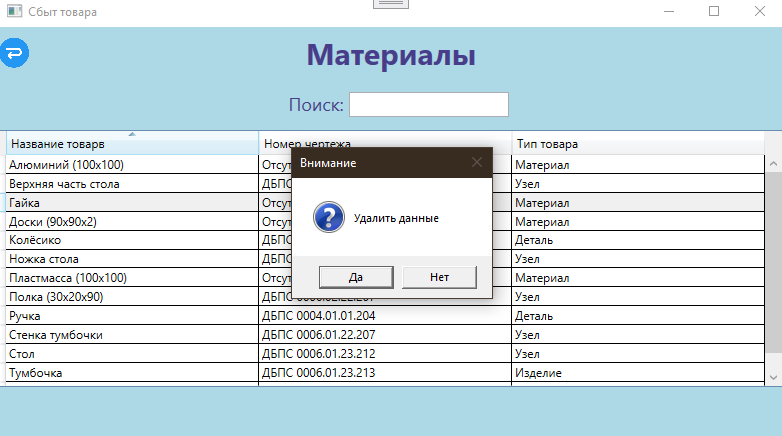


Рис. 77 «Страница материалов с уведомление на удаление»

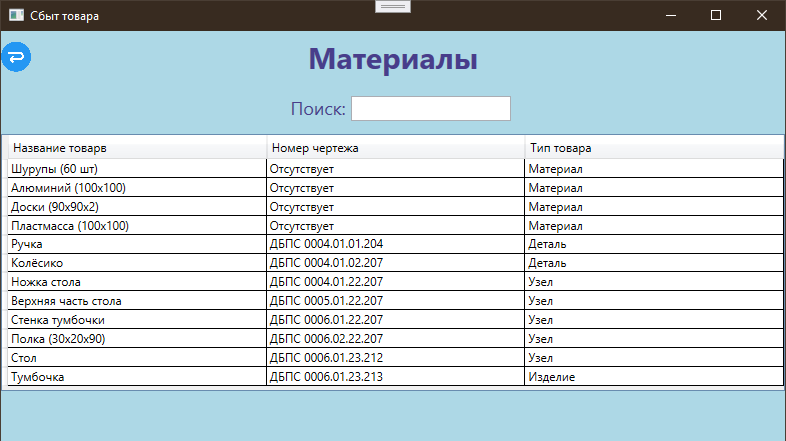


Рис. 78 «Страница материалов после удаления»

В приложении присутствует фильтрация для нахождения данных по атрибутам таблицы. С помощью ПКМ из списка выбираем из «Фильтрации» нужные нам данные. Если список с нужным фильтром больше не нужен, то нажимаем из этого же списка «Сбросить».

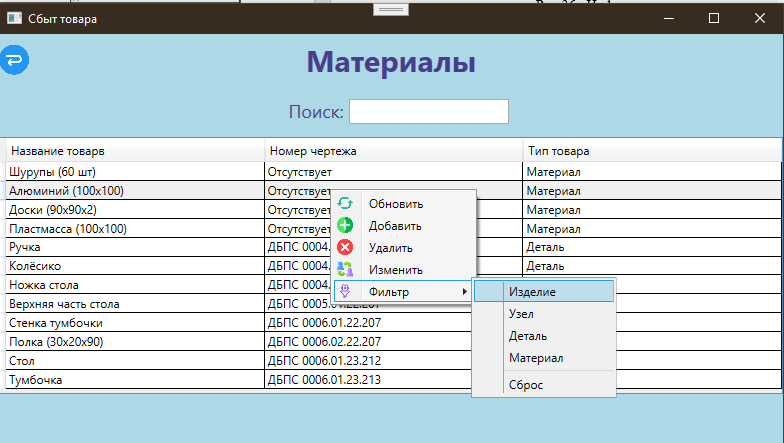


Рис. 79 «Страница материалов с фильтром»

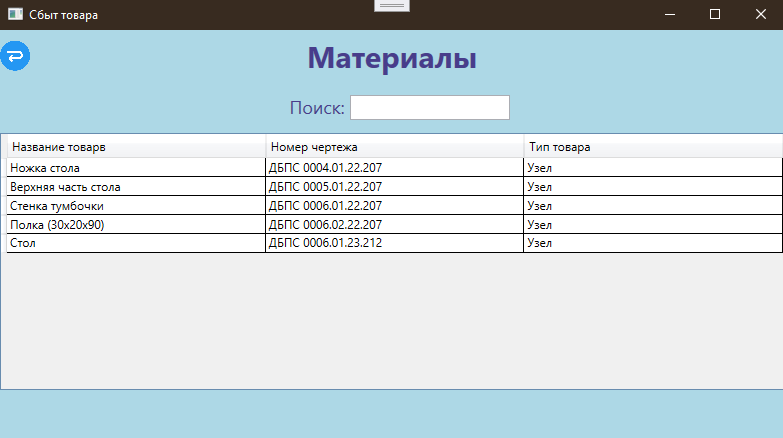


Рис. 80 «Страница материалов после фильтрации»

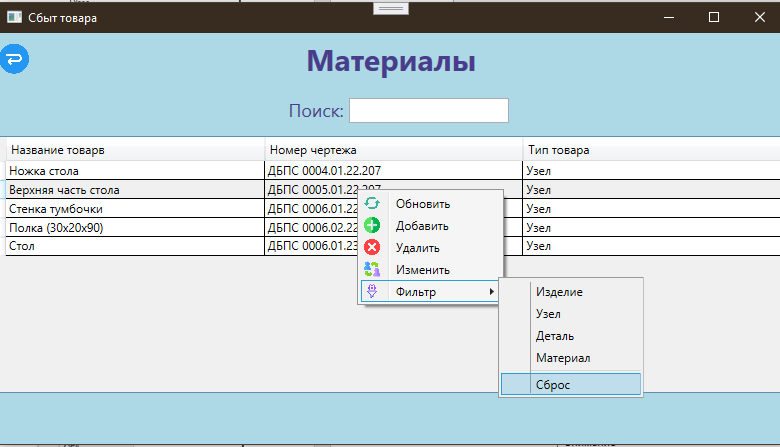


Рис. 81 «Страница материалов сброс фильтрации»

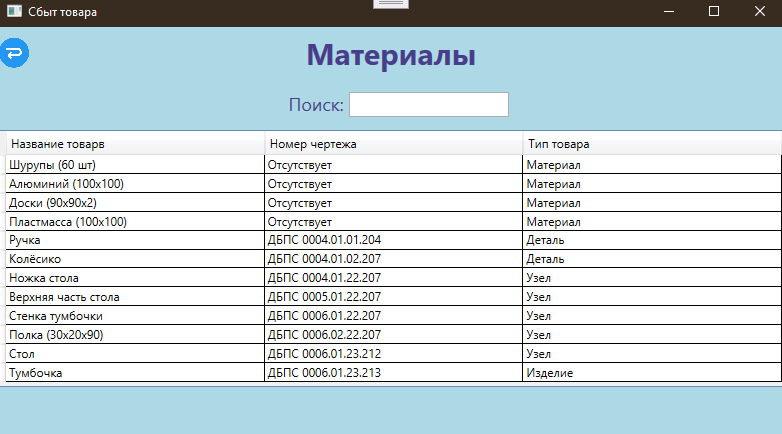


Рис. 82 «Страница материалов после сброса фильтрации»

При нажатии в навигационном меню на кнопку «Контрагент» откроется страница с соответствующим списком контрагентов.

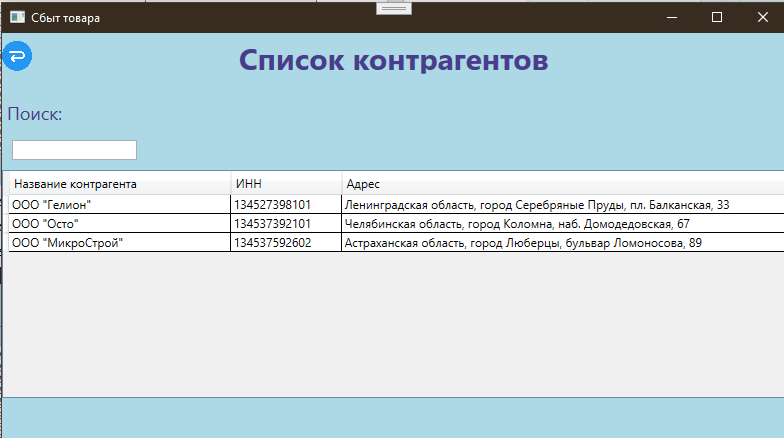


Рис. 83 «Страница контрагентов»

При добавлении информации пользователю необходимо нажать ПКМ по таблице и выбрать из выпадающего списка «Добавить». Затем появится страница для добавления с пустыми значениями, которые потребуется заполнить. Обязательные поля: Название контрагента, ИНН, Адрес.

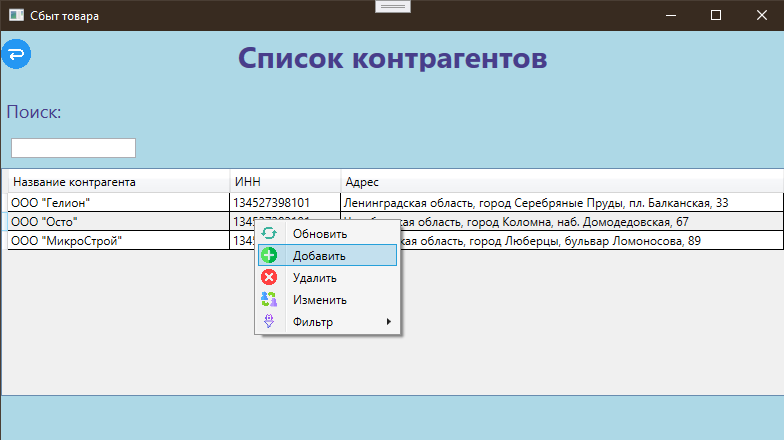


Рис. 84 «Страница контрагентов с вызовом добавления»

Заполним страницу добавления данными.

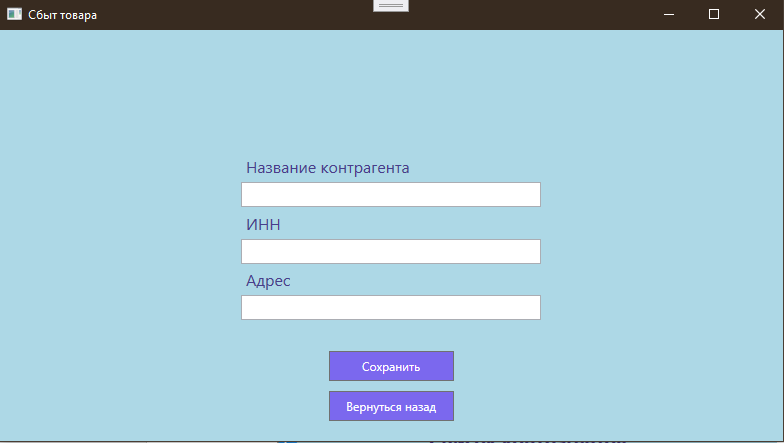


Рис. 85 «Страница добавления контрагентов»

После нажатия кнопки «Сохранить» пользователь возвращается на страницу контрагентов.



Рис. 86 «Страница контрагентов после добавления»

При изменении информации пользователю необходимо нажать ПКМ по таблице и выбрать из выпадающего списка «Изменить». Затем появится страница для изменения с заполненными значениями, которые потребуется изменить. Обязательные поля: Название контрагента, ИНН, Адрес.

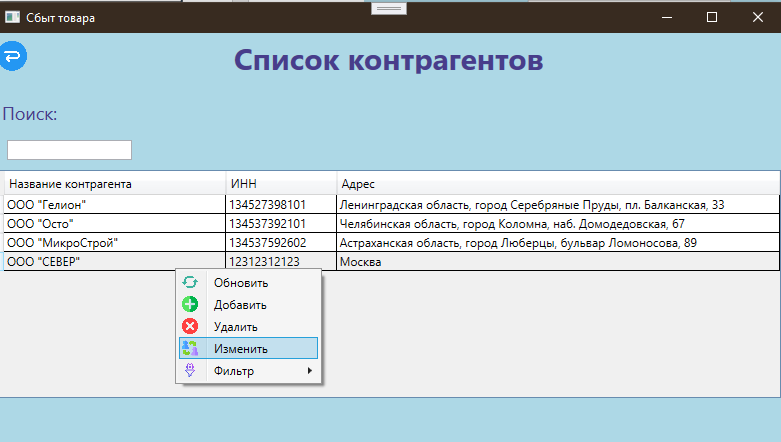
****

Рис. 87 «Страница контрагентов с вызовом изменения»

Изменим данные на странице редактирование.

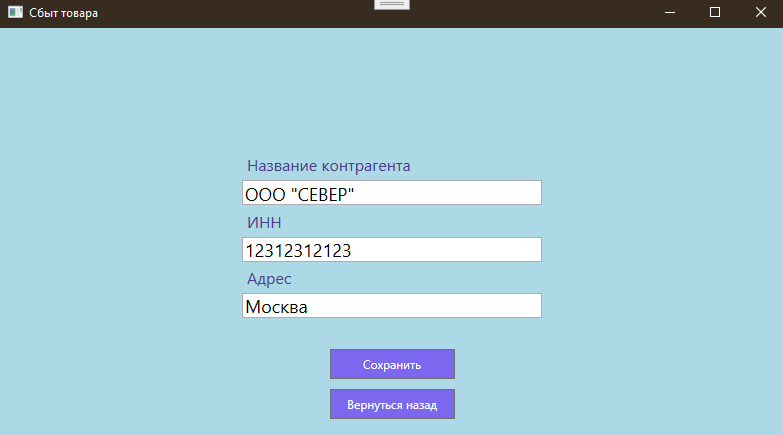


Рис. 88 «Страница редактирование контрагентов»

После нажатия кнопки «Сохранить» пользователь возвращается на страницу контрагентов.

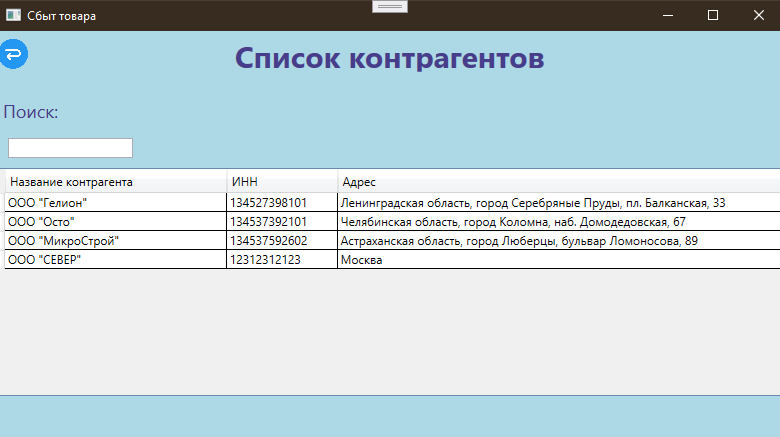


Рис. 89 «Страница контрагентов после редактирования»

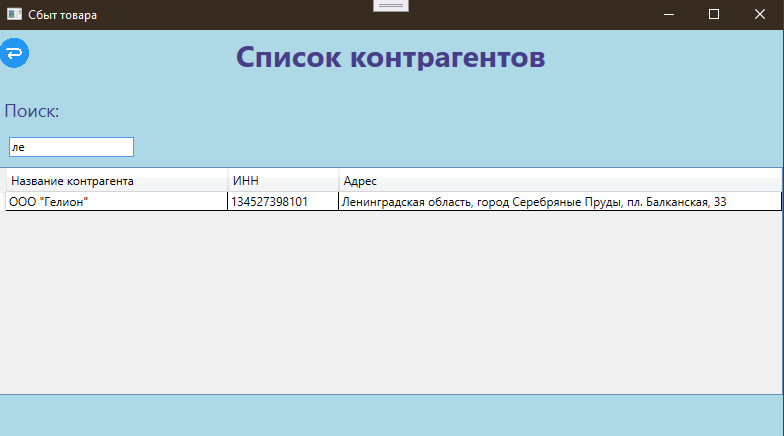
В приложении присутствует поиск для нахождения данных по атрибутам таблицы. В него нужно ввести текст, которые присутствуют в Название контрагента, ИНН, Адрес. 

Рис. 90 «Страница контрагентов с поиском»

Чтобы убрать результат операций, пользователю нужно удалить данные из поиска.

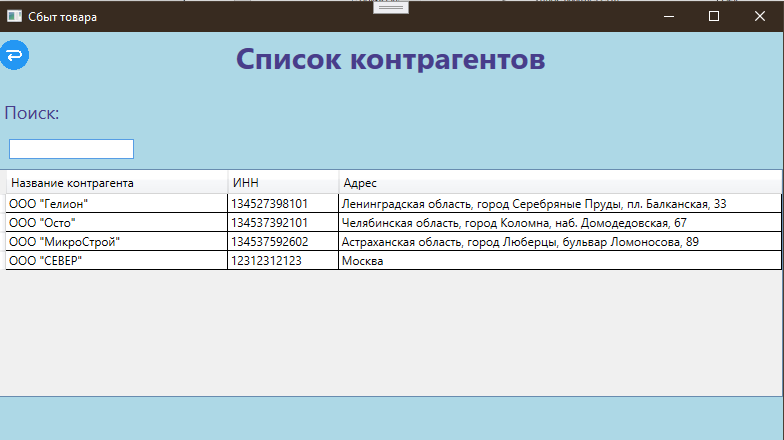


Рис. 91 «Страница контрагентов после сброса данных»

При удалении информации пользователю необходимо нажать ПКМ и из выпадающего списка выбрать «Удалить». Затем появится уведомление «Удалить».

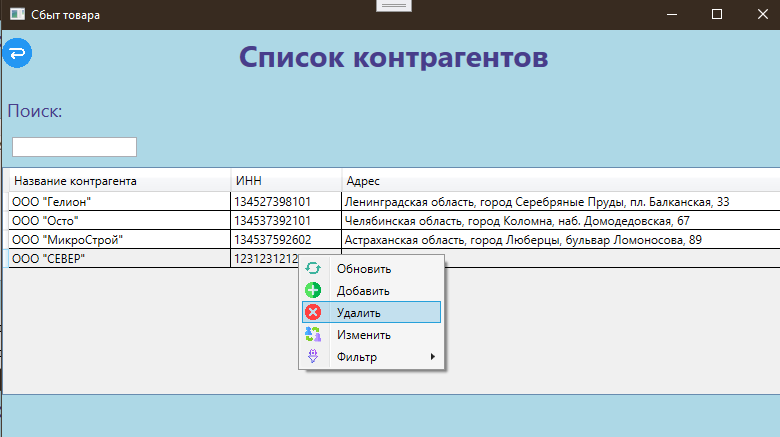


Рис. 92 «Страница контрагентов с удалением записей»

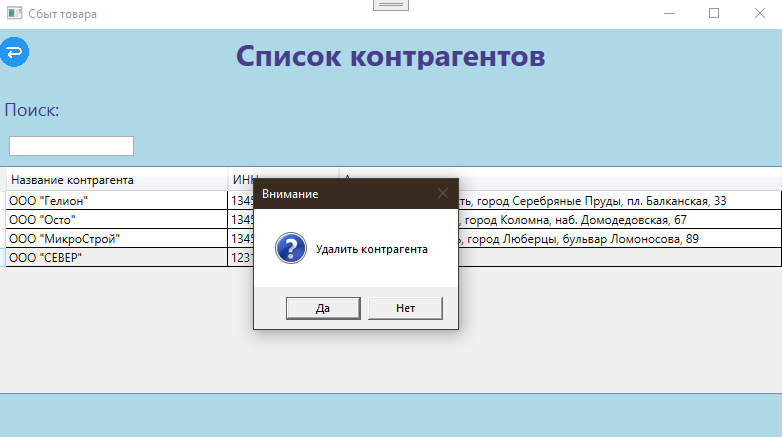


Рис. 93 «Страница контрагентов с уведомление на удаление»

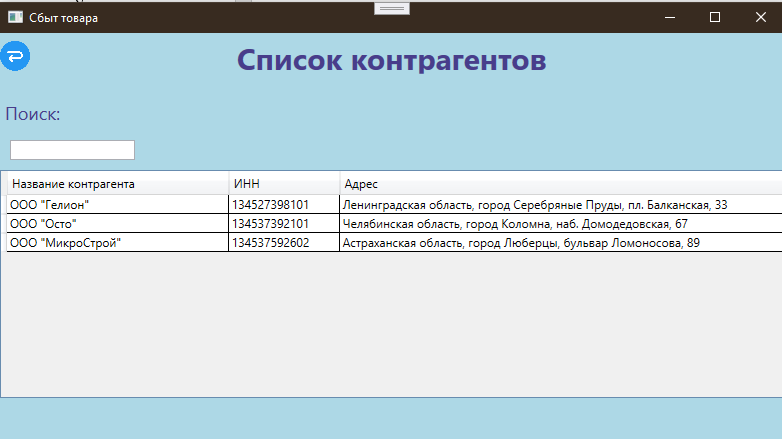


Рис. 94 «Страница контрагентов после удаления»

При нажатии в навигационном меню на кнопку «Товары хранящиеся на складе» откроется страница с соответствующим списком товаров.



Рис. 95 «Страница материалов хранящихся на складе»

Для просмотра из чего сделан товар необходимо выбрать товар из списка и нажать двойным щелчком по нему.



Рис. 96 «Страница материалов хранящихся на складе просмотр иерархии»

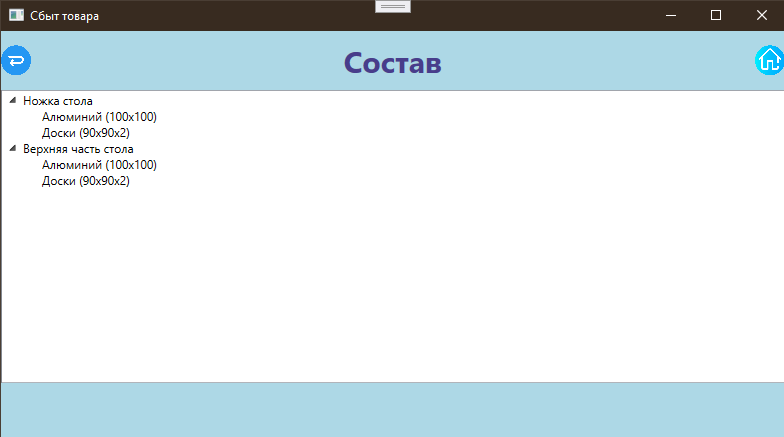


Рис. 97 «Страница материалов хранящихся на складе просмотр иерархии»

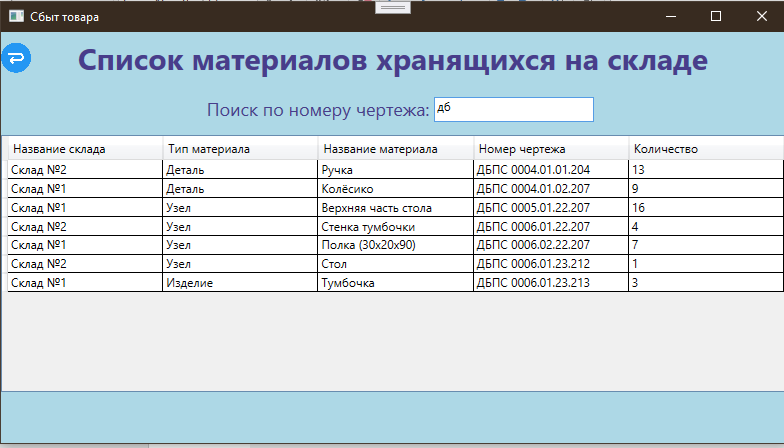
В приложении присутствует поиск для нахождения данных по атрибутам таблицы. В него нужно ввести текст, которые присутствуют в номере чертежа. 

Рис. 98 «Страница материалов хранящихся на складе с поиском»

Чтобы убрать результат операций, пользователю нужно удалить данные из поиска.



Рис. 99 «Страница материалов хранящихся на складе после сброса данных»

В приложении присутствует фильтрация для нахождения данных по атрибутам таблицы. С помощью ПКМ из списка выбираем из «Фильтрации» нужные нам данные. Если список с нужным фильтром больше не нужен, то нажимаем из этого же списка «Сбросить».

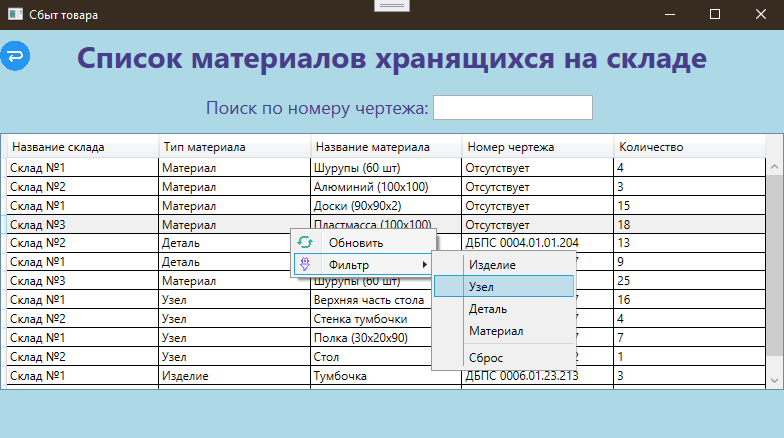


Рис. 100 «Страница материалов хранящихся на складе с фильтром»

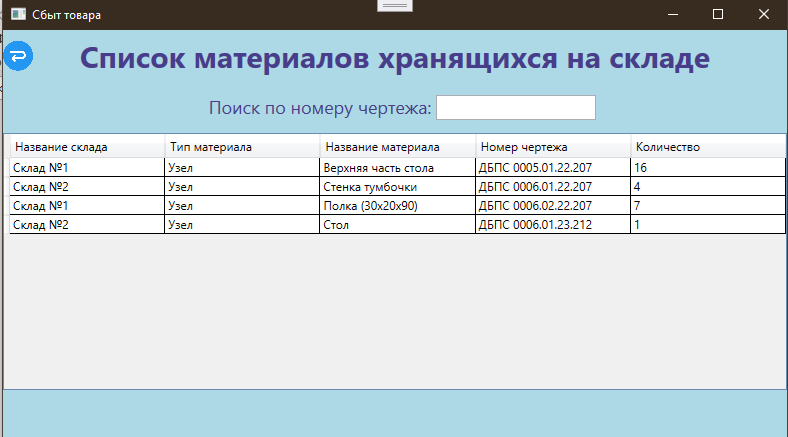


Рис. 101 «Страница материалов хранящихся на складе после фильтрации»

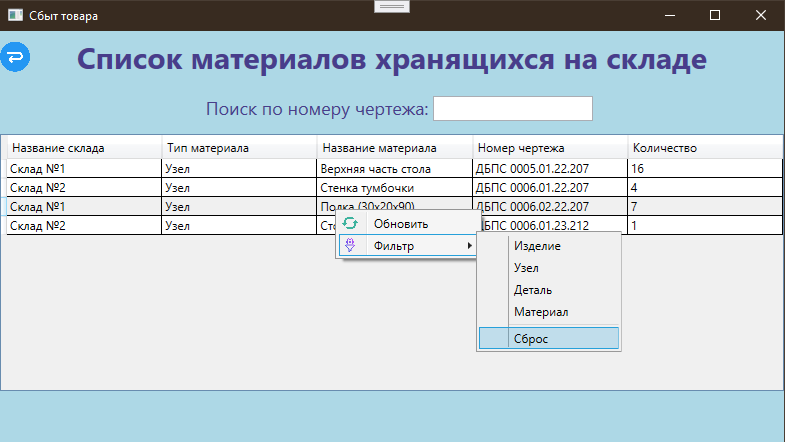


Рис. 102 «Страница материалов хранящихся на складе сброс фильтрации»



Рис. 103 «Страница материалов хранящихся на складе после сброса фильтрации»

При нажатии в навигационном меню на кнопку «Отдаваемые материалы» откроется страница с соответствующим списком материалов.



Рис. 104 «Страница отдачи товаров»

Для распределения товара нужно выбрать товар из списка и с помощью двойного щелчка позволить ему ввести количество с определённого склада потом нажать из выпадающего меню «Распределить». Когда весь товар из конкретного товара будет распределён то нужно нажать из выпадающего меню «Проверить распределение» для того чтобы закрыть заказ.

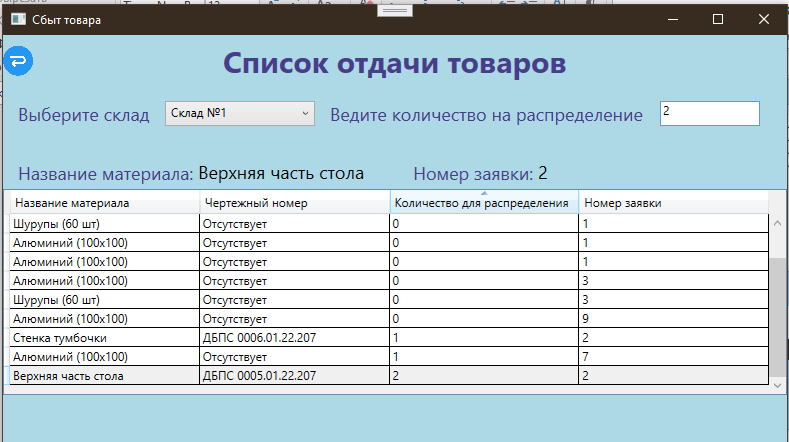


Рис. 105 «Страница отдачи товаров выбор товара на распределение»

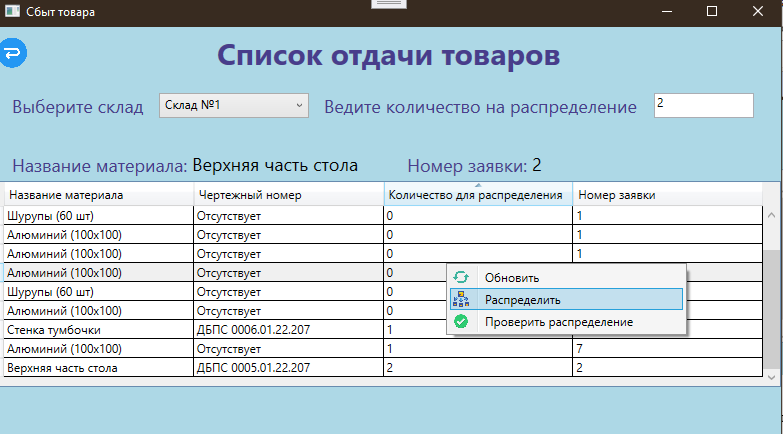


Рис. 106 «Страница отдачи товаров распределение товара»

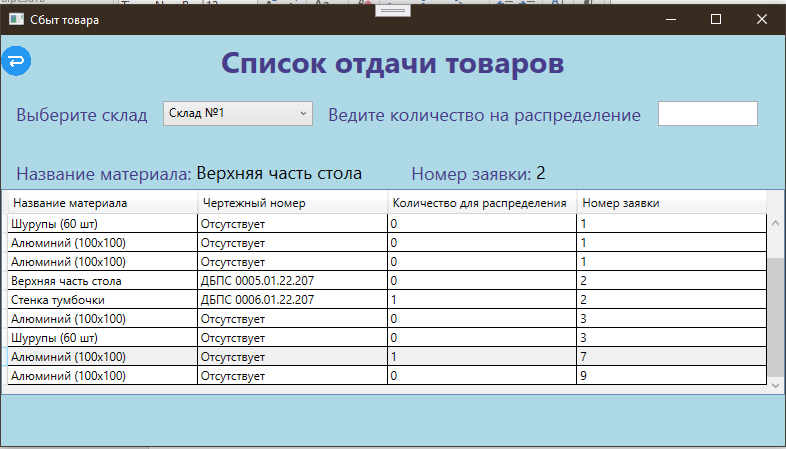


Рис. 107 «Страница отдачи товаров после распределения»

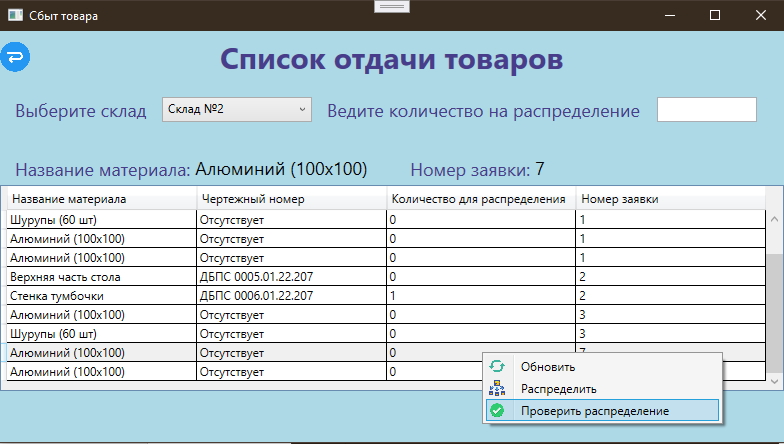


Рис. 108 «Страница отдачи товаров проверка распределения»

При нажатии в навигационном меню на кнопку «Пользователи» откроется страница с соответствующим списком пользователей.

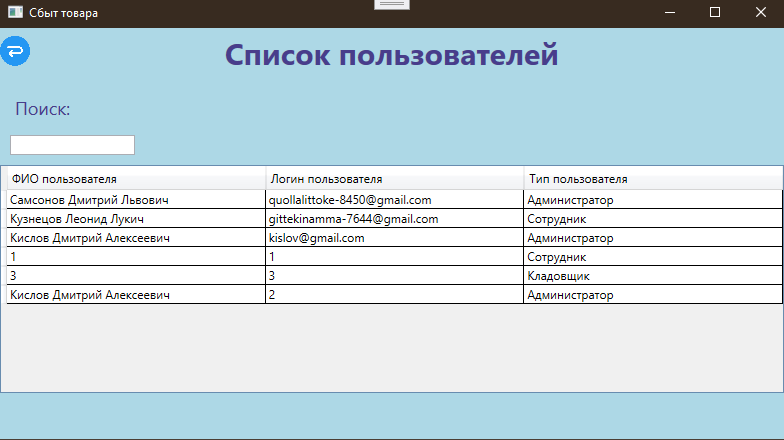


Рис. 109 «Страница пользователей»

При добавлении информации пользователю необходимо нажать ПКМ по таблице и выбрать из выпадающего списка «Добавить». Затем появится страница для добавления с пустыми значениями, которые потребуется заполнить. Обязательные поля: ФИО, Логин пользователя, Пароль пользователя и Тип пользователя.

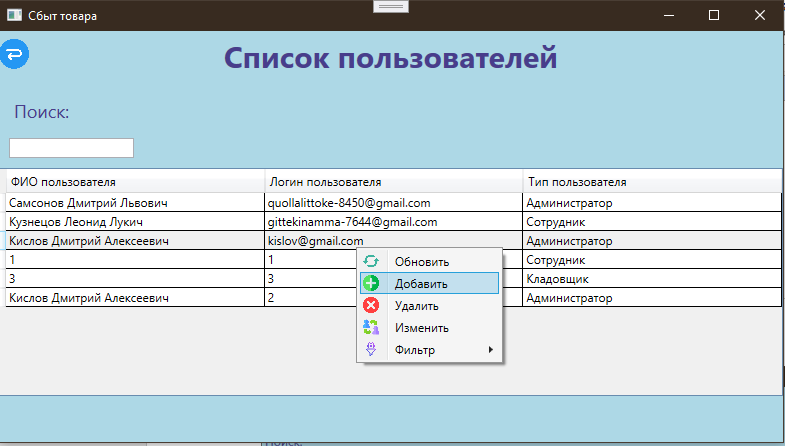


Рис. 110 «Страница пользователей с вызовом добавления»

Заполним страницу добавления данными.

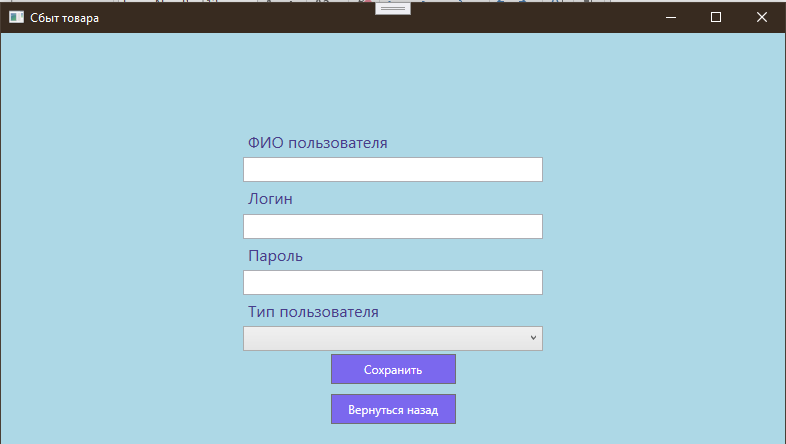


Рис. 111 «Страница добавления пользователей»

После нажатия кнопки «Сохранить» пользователь возвращается на страницу пользователей.

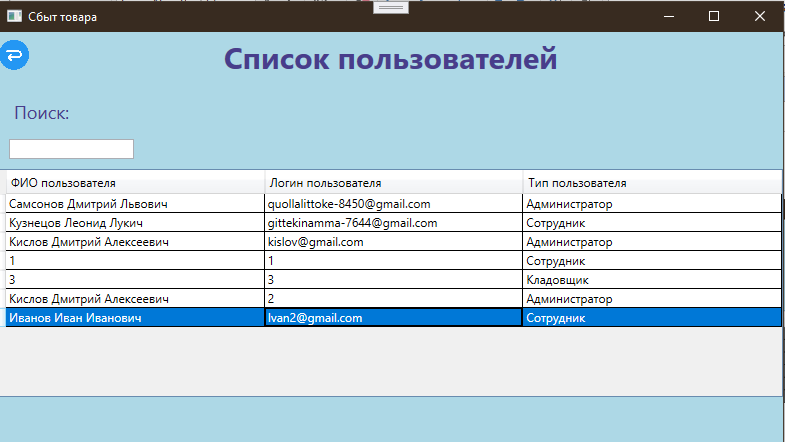


Рис. 112 «Страница пользователей после добавления»

При изменении информации пользователю необходимо нажать ПКМ по таблице и выбрать из выпадающего списка «Изменить». Затем появится страница для изменения с заполненными значениями, которые потребуется изменить. Обязательные поля: ФИО, Логин пользователя, Пароль пользователя и Тип пользователя.

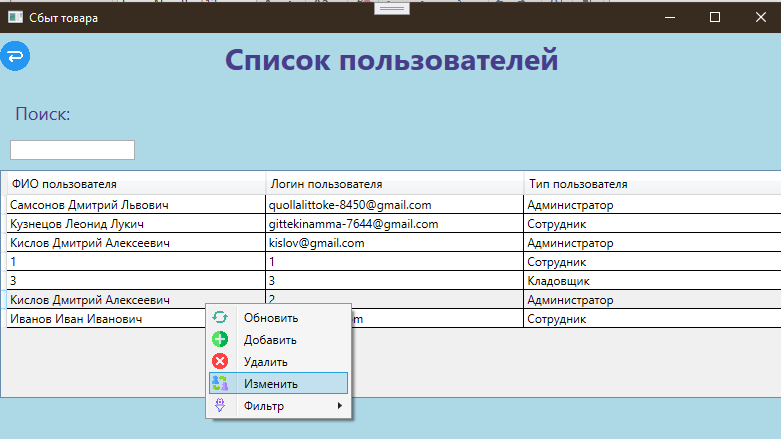
****

Рис. 113 «Страница пользователей с вызовом изменения»

Изменим данные на странице редактирование.

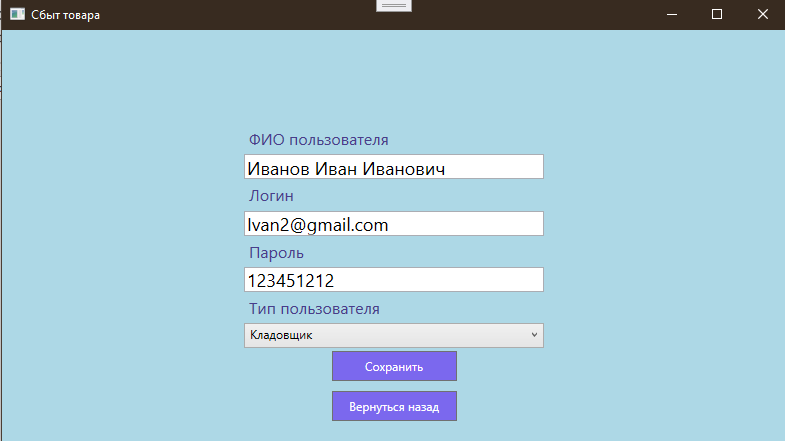


Рис. 114 «Страница редактирование пользователей»

После нажатия кнопки «Сохранить» пользователь возвращается на страницу пользователей.

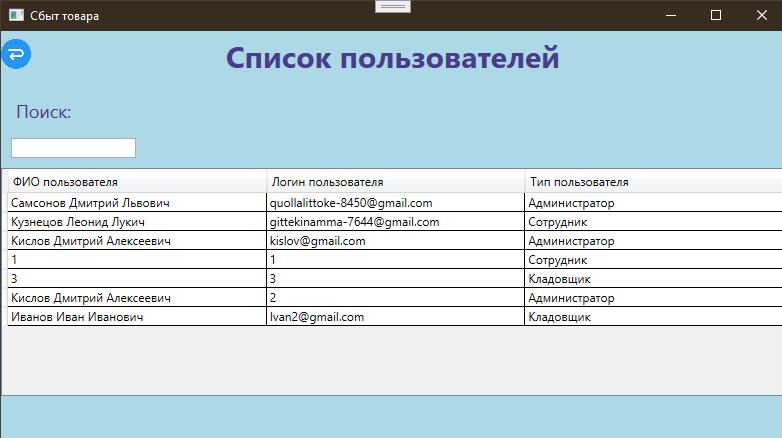


Рис. 115 «Страница пользователей после редактирования»

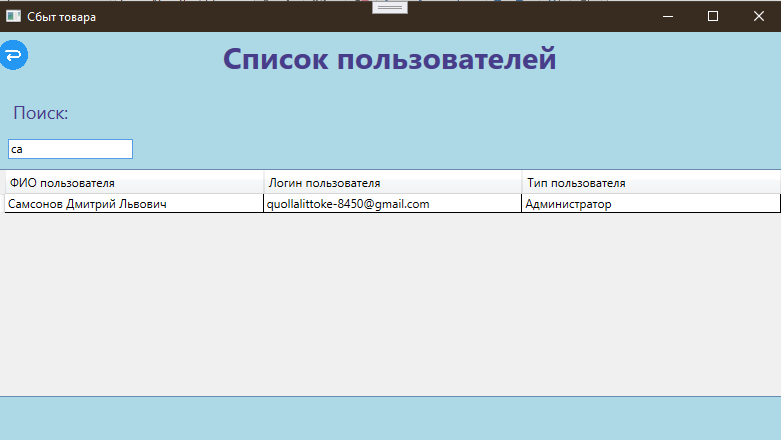
В приложении присутствует поиск для нахождения данных по атрибутам таблицы. В него нужно ввести текст, которые присутствуют в ФИО, Логин пользователя, и Тип пользователя. 

Рис. 116 «Страница пользователей с поиском»

Чтобы убрать результат операций, пользователю нужно удалить данные из поиска.

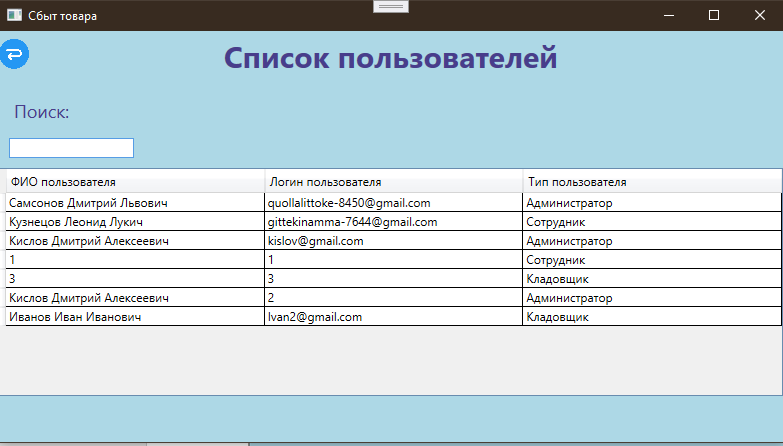


Рис. 117 «Страница пользователей после сброса данных»

При удалении информации пользователю необходимо нажать ПКМ и из выпадающего списка выбрать «Удалить». Затем появится уведомление «Удалить».

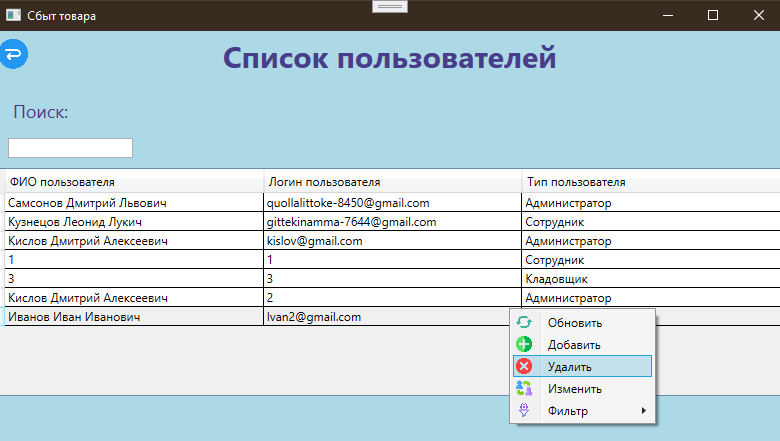


Рис. 118 «Страница пользователей с удалением записей»

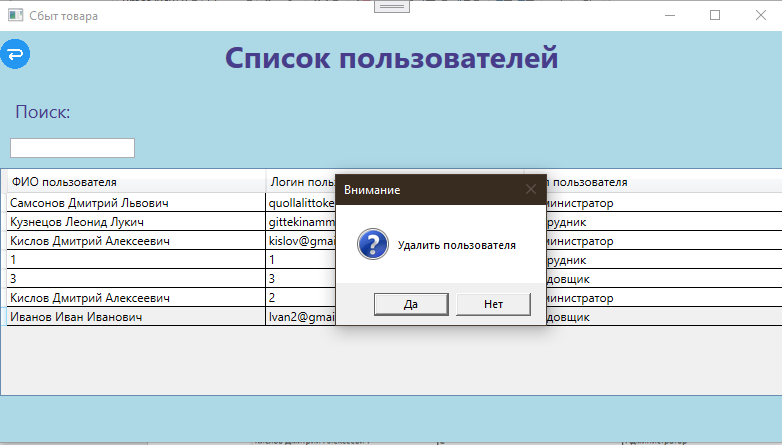


Рис. 119 «Страница пользователей с уведомление на удаление»

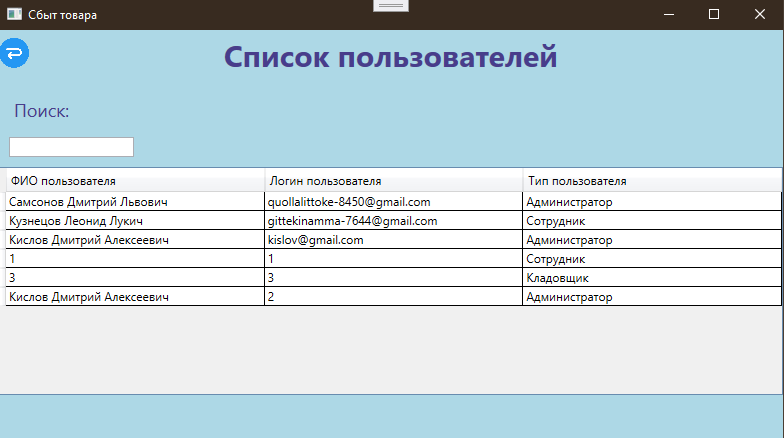


Рис. 120 «Страница пользователей после удаления»

В приложении присутствует фильтрация для нахождения данных по атрибутам таблицы. С помощью ПКМ из списка выбираем из «Фильтрации» нужные нам данные. Если список с нужным фильтром больше не нужен, то нажимаем из этого же списка «Сбросить».

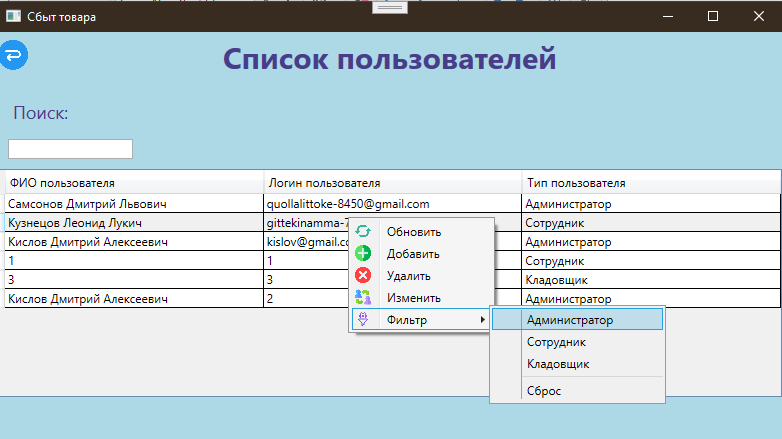


Рис. 121 «Страница пользователей с фильтром»

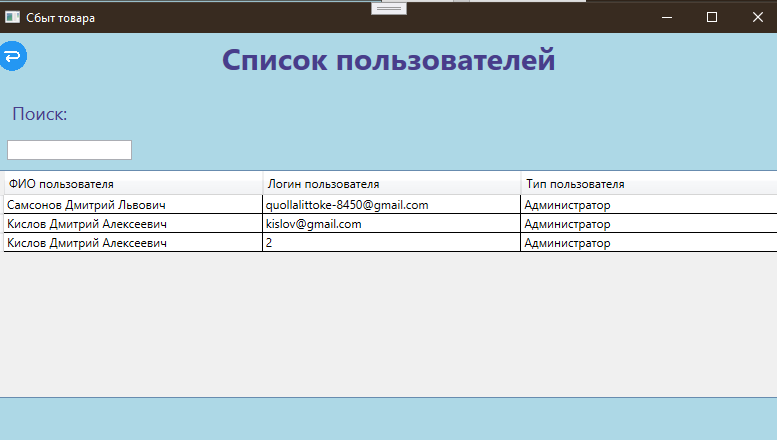


Рис. 122 «Страница пользователей после фильтрации»

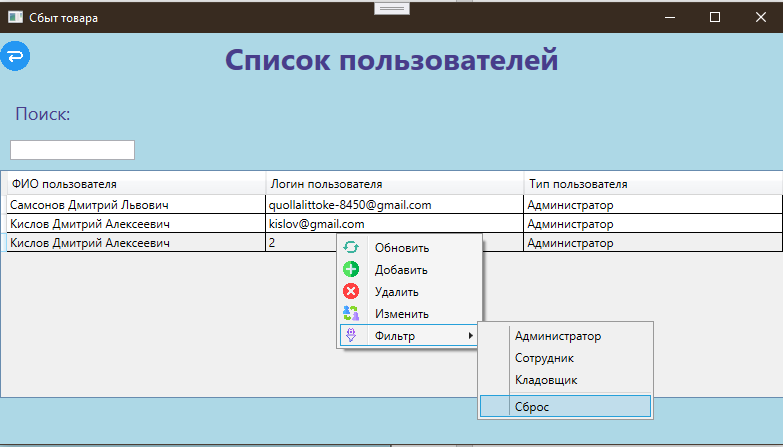


Рис. 123 «Страница пользователей сброс фильтрации»

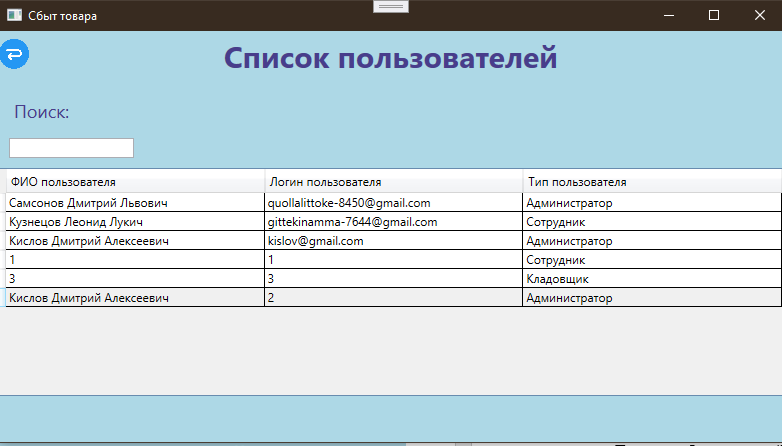


Рис. 124 «Страница пользователей после сброса фильтрации»

При нажатии в навигационном меню на кнопку «Отданные материалы» откроется страница с соответствующим списком материалов.



Рис. 125 «Страница отданных материалов»

В приложении присутствует поиск для нахождения данных по атрибутам таблицы. В него нужно ввести текст, которые присутствуют в Складе. 

Рис. 116 «Страница пользователей с поиском»

Чтобы убрать результат операций, пользователю нужно удалить данные из поиска.



Рис. 117 «Страница пользователей после сброса данных»

**Сообщения пользователю:**

Попытка добавления нового заказа в список с неуказанными данными.

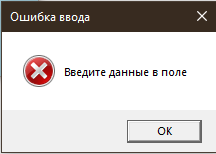


Рис.118 «Попытка добавления заказа»

Подтверждение удаления нескольких заказов.

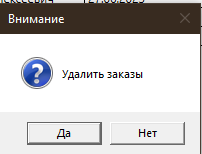


Рис.119 «Подтверждение удаления нескольких заказов»

Информация после удаления нескольких пунктов.

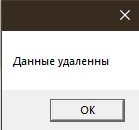


Рис.120 «Информация при удалении нескольких пунктов»

Информация после редактирования проекта.

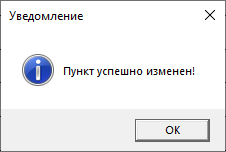


Рис.121 «Уведомление после изменения»

Информация после распределения товара.

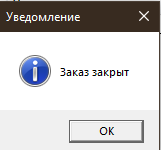


Рис.122 «Уведомление после распределения»

## 3.6. Отладка ПО

Участок кода с добавление материала в таблицу, работающий некорректно:

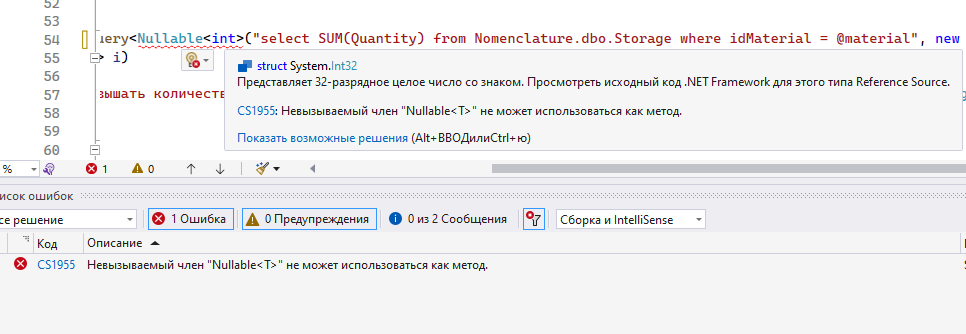


Рис.123 «Неявное преобразование типов»

Исправленный участок кода с добавление материала в таблицу, работающий корректно:



Рис.124 «Переделанный код для метода»

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

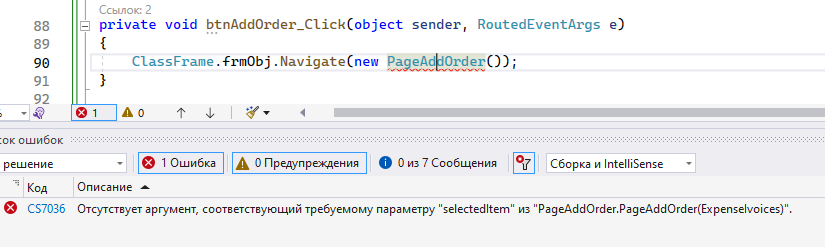


Рис.125 «Отсутствие аргумента, требуемой страницы»

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

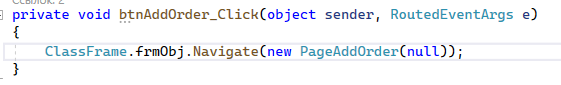


Рис.126 «Добавление пустого аргумента, требуемой странице»

## 3.7. Методика тестирования и испытания задачи

Таблица№5 «Аннотация теста»

|  |  |
| --- | --- |
| **Поле** | **Описание** |
| Название проекта | SalesOfMaterials |
| Рабочая версия | 8.3 |
| Имя тестирующего | Стародубцев Максим Сергеевич |
| Дата(ы) теста | 02.06.2023 |

Таблица№6 «Тест№1»

|  |  |
| --- | --- |
| **Поле** | **Описание** |
| Тестовый пример # | Test1 |
| Приоритет тестирования  (Низкий/Средний/Высокий) | Средний |
| Заголовок/название теста | Добавить запись, не указав одну из характеристик. |
| Краткое изложение теста | Попытка добавления записи в таблицу «Контрагентов», если не указать одну из характеристик. |
| Этапы теста | Нажать на таблицу с помощью ПКМ и выбрать из списка пункт «Добавить» и заполнить следующие данные: Название контрагенты, ИНН. Но оставить значение поля Адрес пустым. |
| Тестовые данные | Название контрагенты: ООО «Вера», ИНН: 134537592602, Адрес: пустое. |
| Ожидаемый результат | Предупреждающее сообщение о незаполненном поле «Адрес». |
| Фактический результат | Сообщение о том, что поле Адрес незаполненное. |
| Предварительное условие | На главном меню пользовательского интерфейса должна быть подсистема в которой хранится нужный нам справочник для добавления данных. |
| Постусловие | В таблицу не добавляются данные |
| Статус(Зачет/Незачет) | Зачет |

Таблица№7 «Тест№2»

|  |  |
| --- | --- |
| **Поле** | **Описание** |
| Тестовый пример # | Test2 |
| Приоритет тестирования  (Низкий/Средний/Высокий) | Средний |
| Заголовок/название теста | Отрицательная сумма. |
| Краткое изложение теста | Не должно осуществляться сохранение записи, в которой указана отрицательное количество товара. |
| Этапы теста | Открыть контекстное меню добавление и заполнить следующие данные: Код, Контрагент, Склад, Магазин, Номенклатура, Единица измерения, Количество, Цена, Виды цен, Сумма. Но поля Количество сделать отрицательным |
| Тестовые данные | Код:1, Контрагент: Царство Золота, Склад: Дельта, Магазин: Золотой рай, Номенклатура: Цепочка, Единица измерения: Штука, Количество: -5, Цена: 5000, Виды цен: Оптовая, Сумма: -25 000. |
| Ожидаемый результат | Предупреждающее сообщение об отрицательном значении полей «Количество» и «Сумма». |
| Фактический результат | Запись проведена! |
| Предварительное условие | На главном меню пользовательского интерфейса должна быть подсистема в которой хранится нужный нам документ для добавления данных. |
| Постусловие | Запись была добавлена. |
| Статус (Зачет/Незачет) | Незачет |

Таблица№8 «Тест№3»

|  |  |
| --- | --- |
| **Поле** | **Описание** |
| Тестовый пример # | Test3 |
| Приоритет тестирования  (Низкий/Средний/Высокий) | Низкий |
| Заголовок/название теста | Ввод в поисковую строку цифр с буквами. |
| Краткое изложение теста | Если при вводе в поисковую строку мы вводим два символа, один из которых есть в записях, а другого нет, тогда записи не должны отображаться. |
| Этапы теста | В поисковую строку вводим символ, который точно есть в записи, затем, которого нет ни в одной записи (например: цифру, если поиск происходит по атрибуту Номенклатуры). |
| Тестовые данные | В поисковую строку введём значение «Ц6» |
| Ожидаемый результат | Записи не должны отображаться. |
| Фактический результат | Не отобразилось ни одной записи. |
| Предварительное условие | В таблице должна быть информация, чтобы осуществить поиск. |
| Постусловие | Не отобразилось ни одной записи. |
| Статус (Зачет/Незачет) | Зачет |

## 3.8. Текст программы

Представлен в [Приложении](#_Приложение)

## 3.9 Предложения по улучшению

# Спектр функционала и удобства использования приложением можно расширить. К примеру, можно добавить фотку в личном кабинете. Можно добавить создание изделия из других материалов. Добавить новые обязанности сотрудников и кладовщиков. Изменение интерфейса программы

# 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 10 и Windows 11, 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, удаление и редактирование материалов, заказов, складов, движение, хранения, состав материала, сбыт.
* Поиск материалов, склада, сбыта, хранящихся материалов, пользователей по всем атрибутам
* Фильтрация материалов, склада, сбыта, хранящихся материалов, по всем
* Вывод на печать в Excel состава заказа
* Авторизация с помощью логина и пароля
* Регистрация пользователя

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; }

}