**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

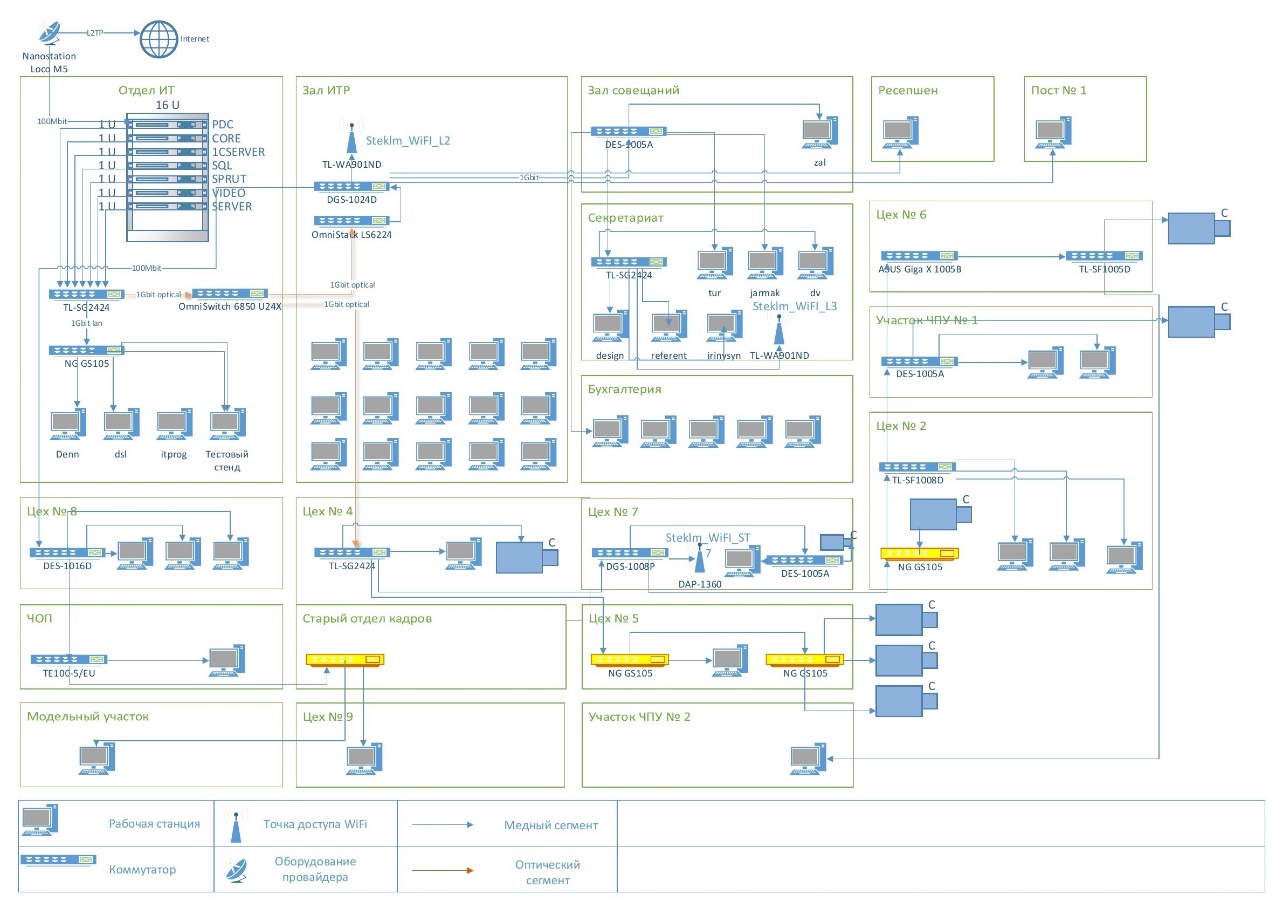


Рис. 2 «Структура локальной сети»

# 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

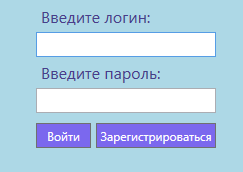


Рис. 3 «Страница авторизации»

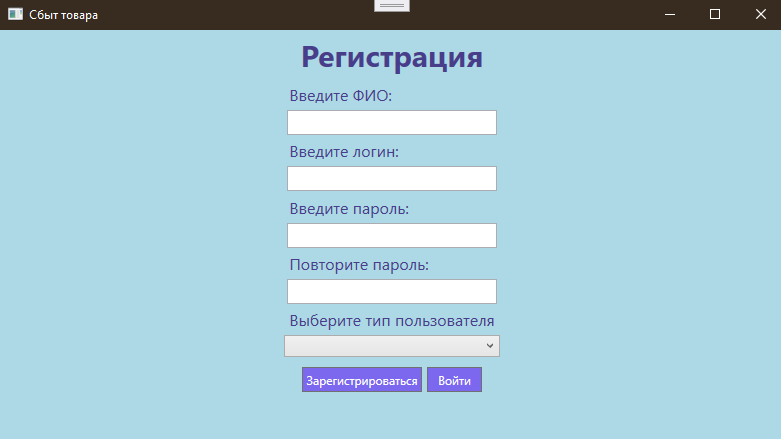


Рис. 4 «Страница регистрации»



Рис. 5 «Главная страница»

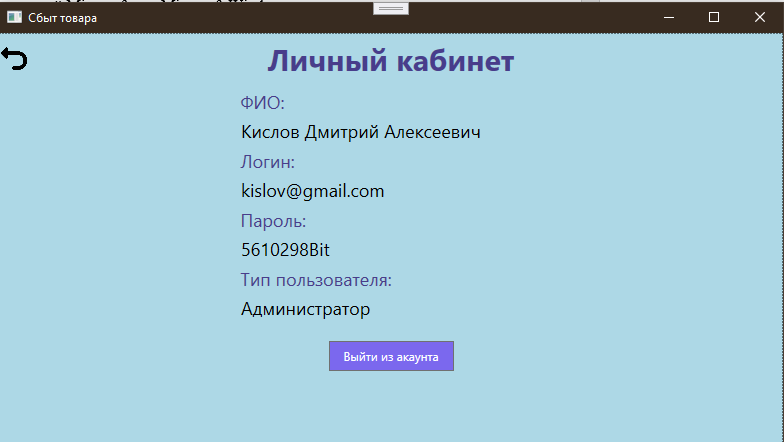


Рис. 6 «Страница c личным кабинетом»

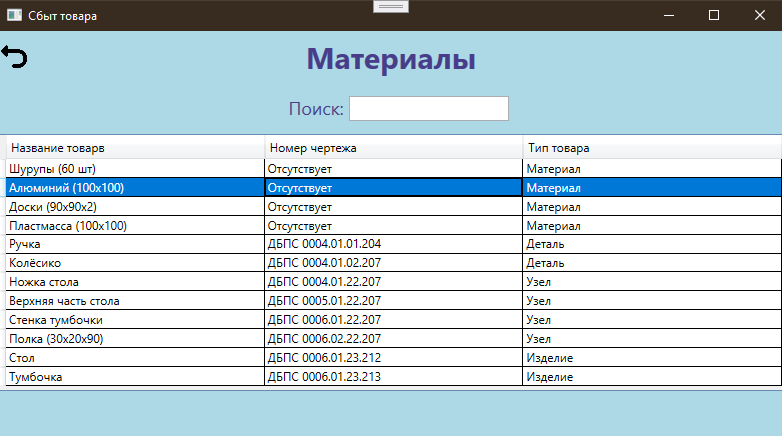


Рис. 7 «Страница материалов»

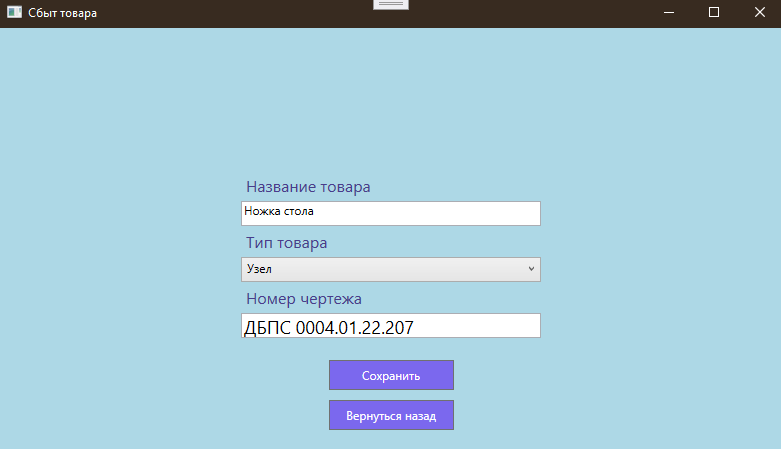


Рис. 8 «Страница добавления и редактирования материалов»

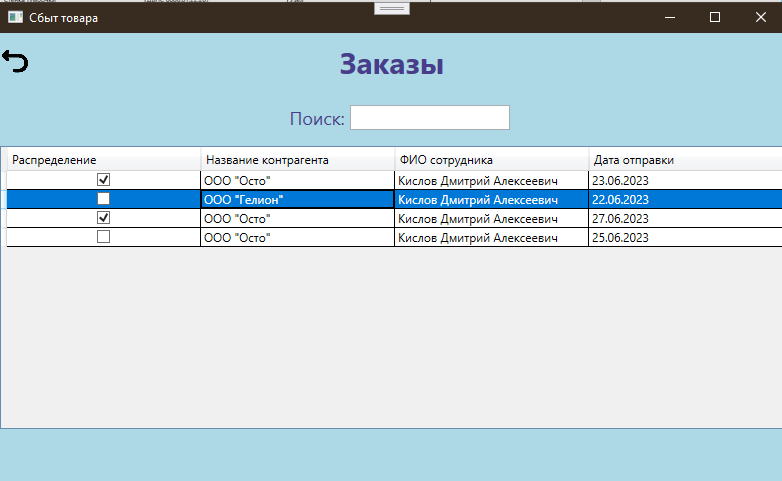


Рис. 9 «Страница заказов»

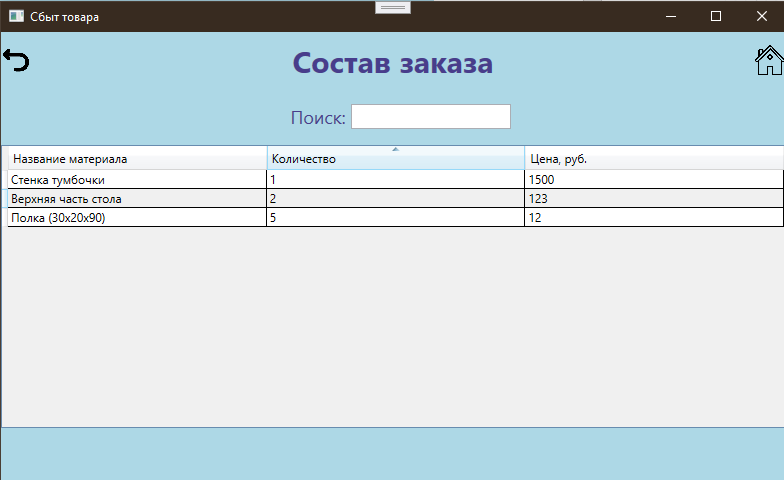


Рис. 10 «Страница состава заказа»

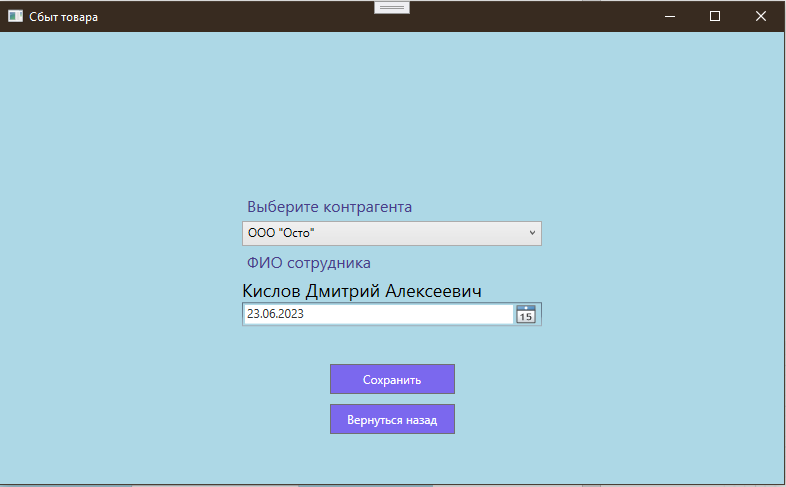


Рис. 11 «Страница добавления и редактирования заказов»

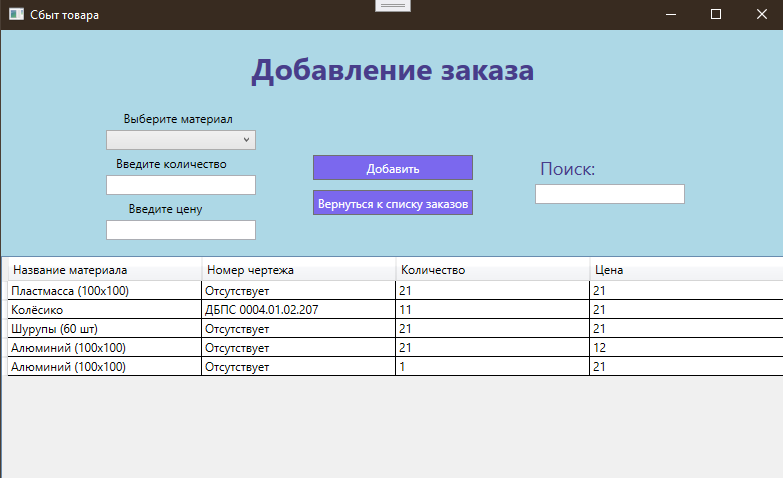


Рис. 12 «Страница добавления и редактирования состава заказов»

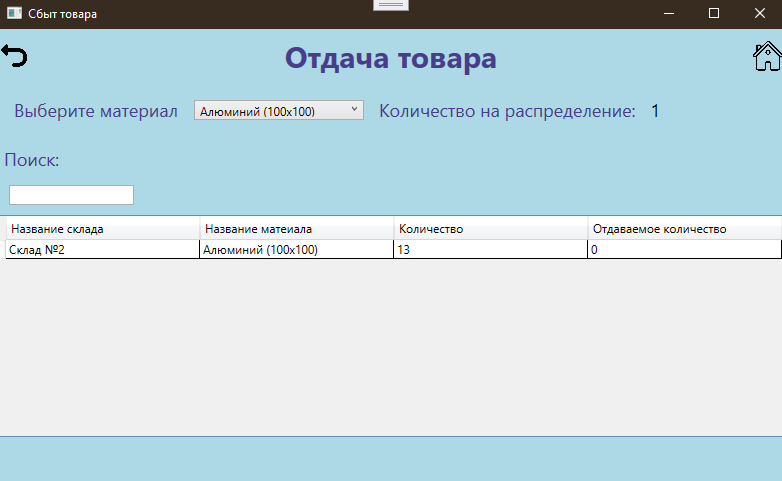


Рис. 13 «Страница отдачи товара»

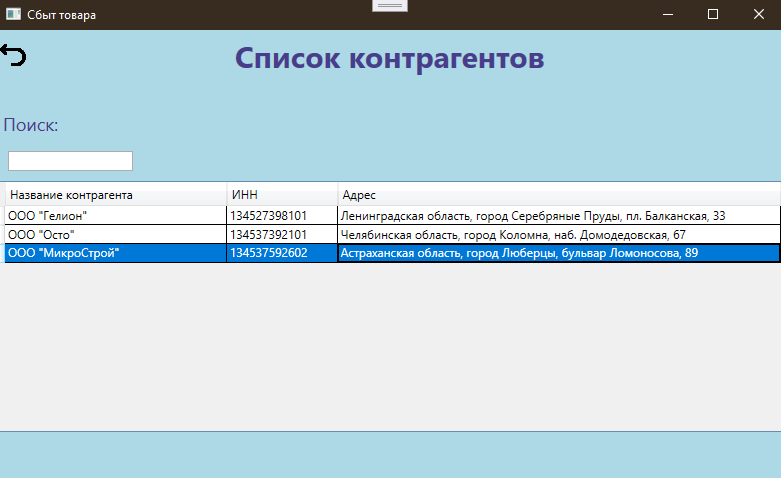


Рис. 14 «Страница контрагентов»

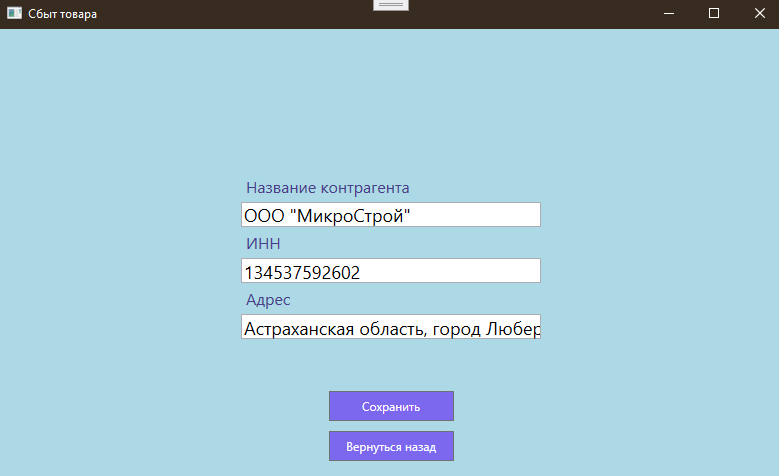


Рис. 15 «Страница добавления и редактирования контрагентов»

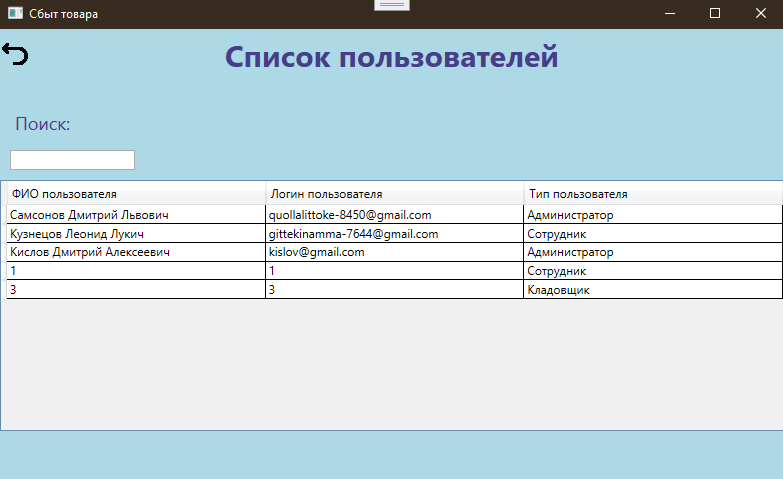


Рис. 16 «Страница пользователей»

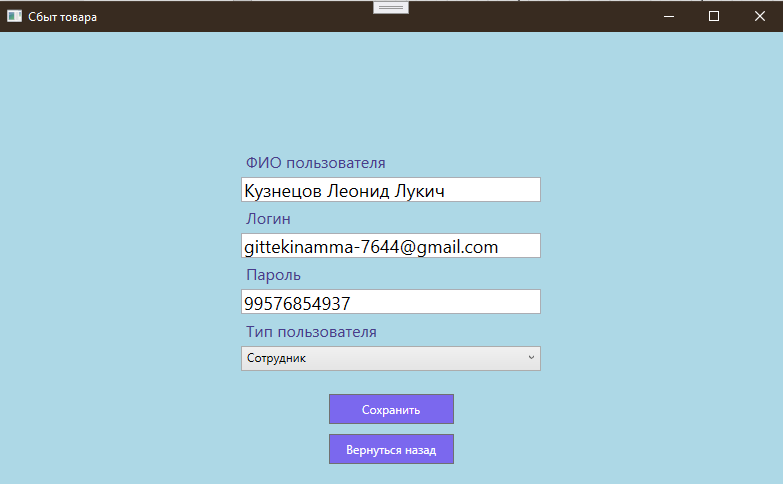


Рис. 17 «Страница добавления и редактирования пользователей»

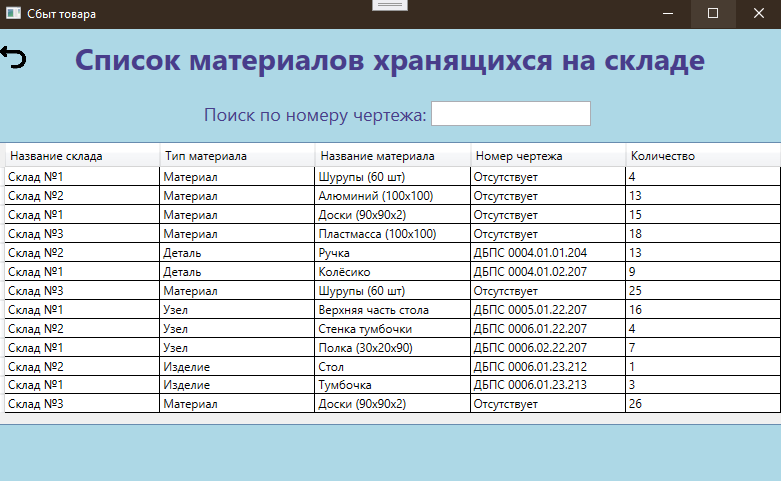


Рис. 18 «Страница материалов со склада»



Рис. 19 «Страница сбыта товара»

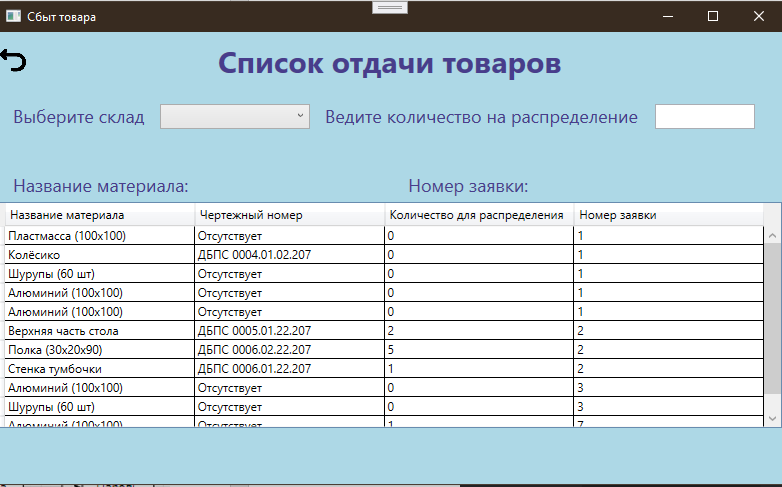


Рис. 20 «Страница отдачи товара»

## 3.3. Блок-схема программы

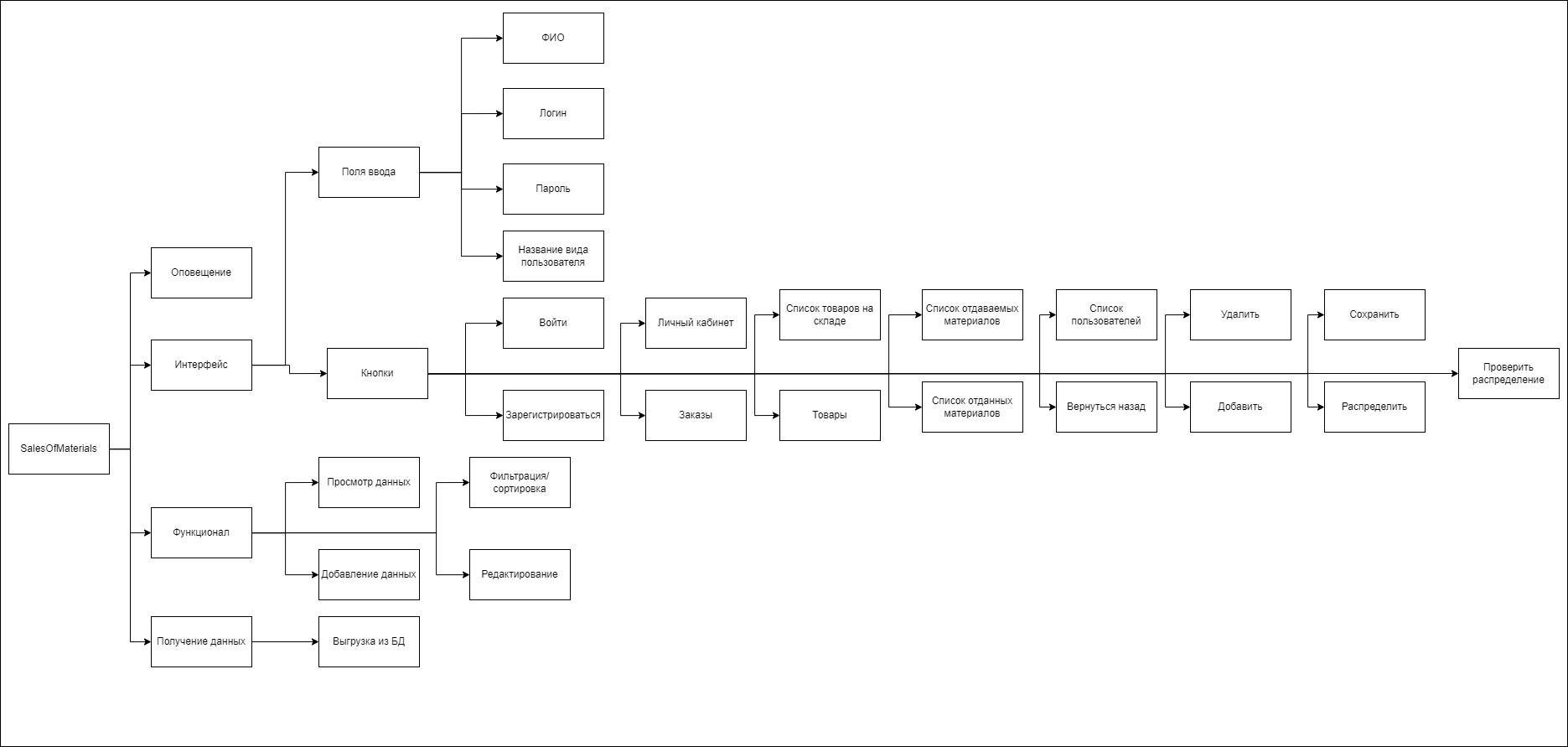


Рис. 21 «Функциональная схема программы»

## 3.4. Руководство программиста

Приложение разработано в Visual Studio 2019

Создание проекта:

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

Рис. 22 «Создание проекта»

Для корректной работы приложения необходимо подключить БД:

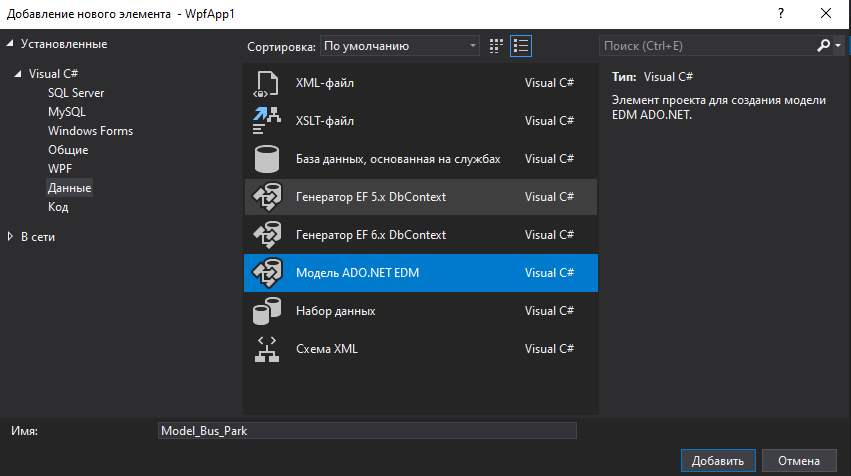


Рис. 23 «Создание модели ADO.NET»

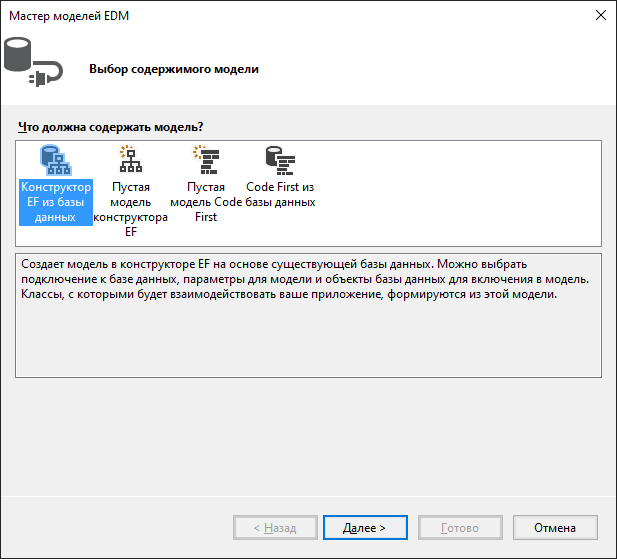


Рис. 24 «Выбор содержимого модели»

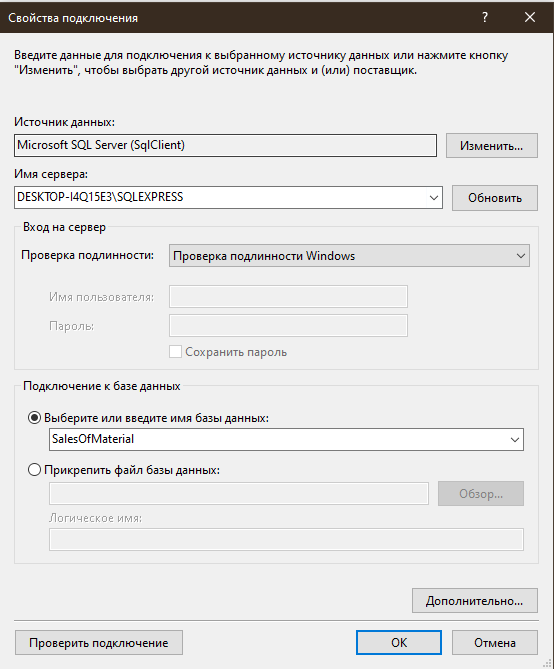


Рис. 25 «Выбор сервера и имени БД»

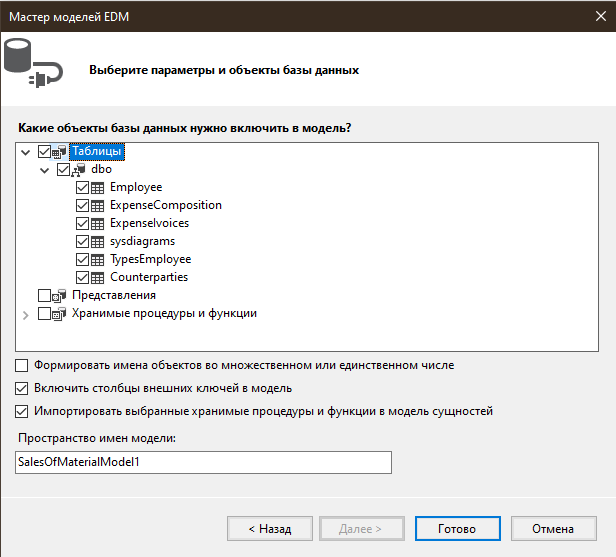


Рис. 26 «Выбор объектов БД»

Диаграмма БД в Visual Studio:

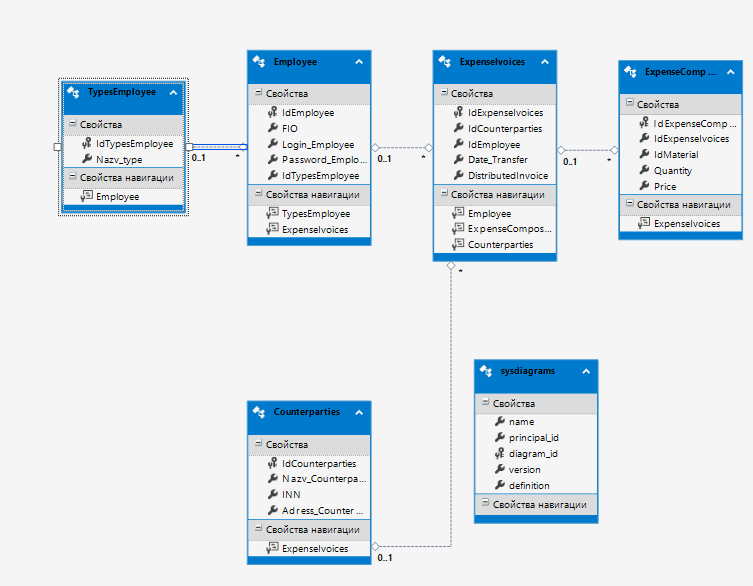


Рис. 33 «Модель данных»

Структура приложения в обозревателе решений:

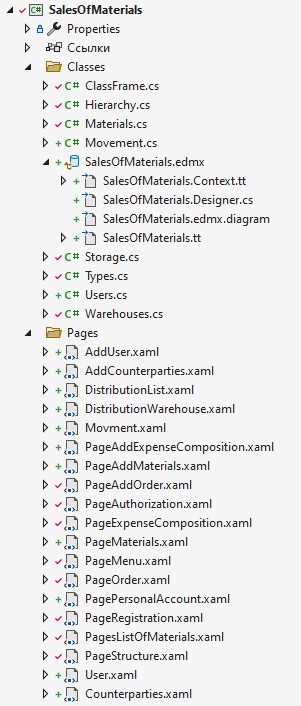


Рис. 34 «Структура приложения»

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».

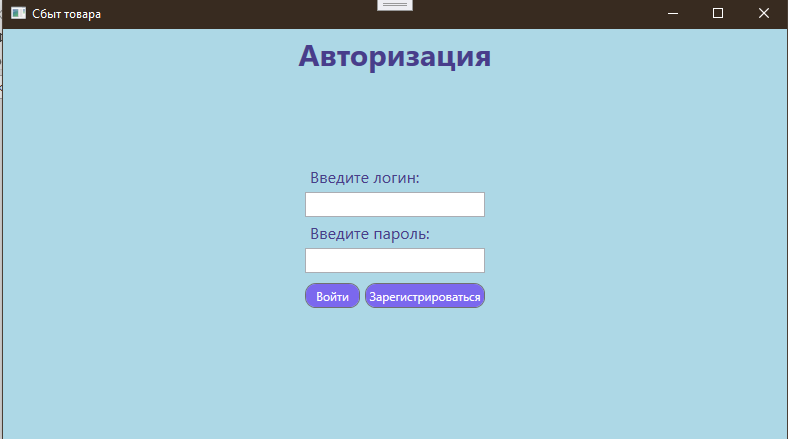


Рис. 35 «Страница авторизации»

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».

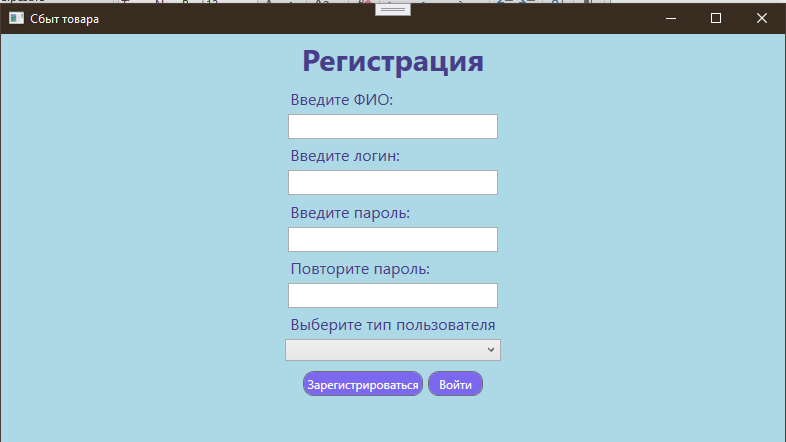


Рис. 36 «Страница регистрации»

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.

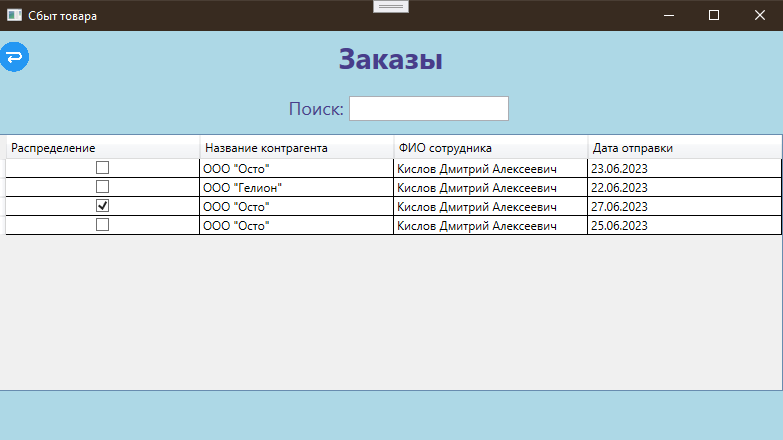


Рис. 38 «Страница заказов»

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.

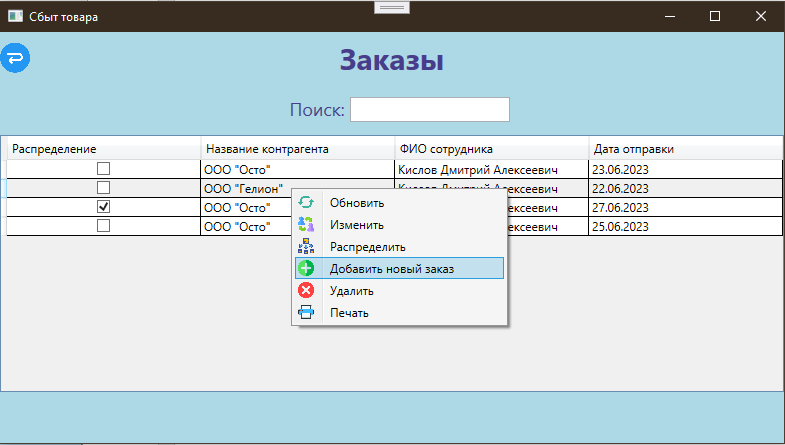


Рис. 39 «Страница заказов с вызовом добавления»

Заполним страницу добавления данными.

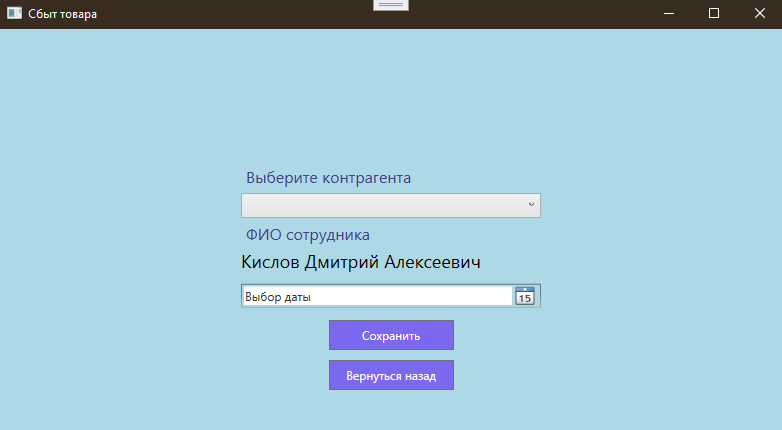


Рис. 40 «Страница добавления заказов»

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

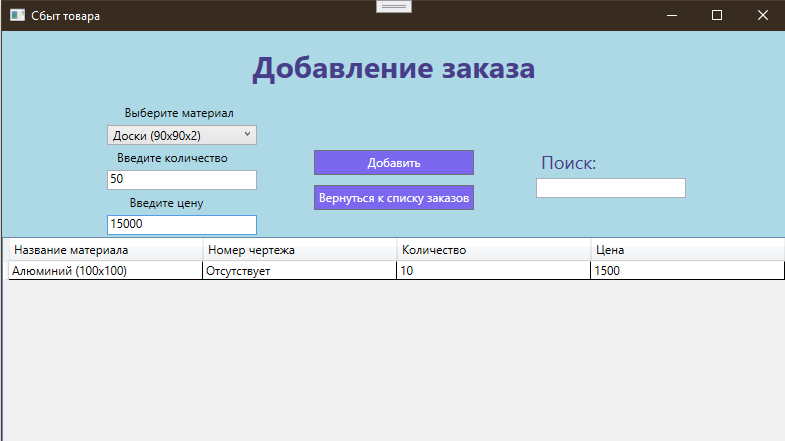


Рис. 41 «Страница добавления состава товара»

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»

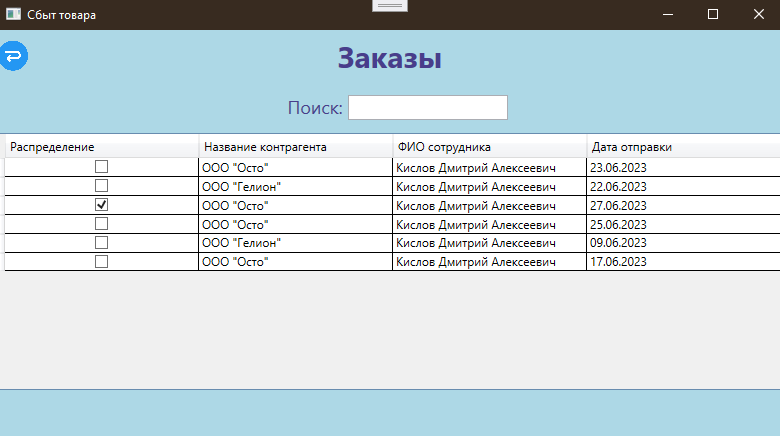


Рис. 42 «Страница заказы с добавленной записью»

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.

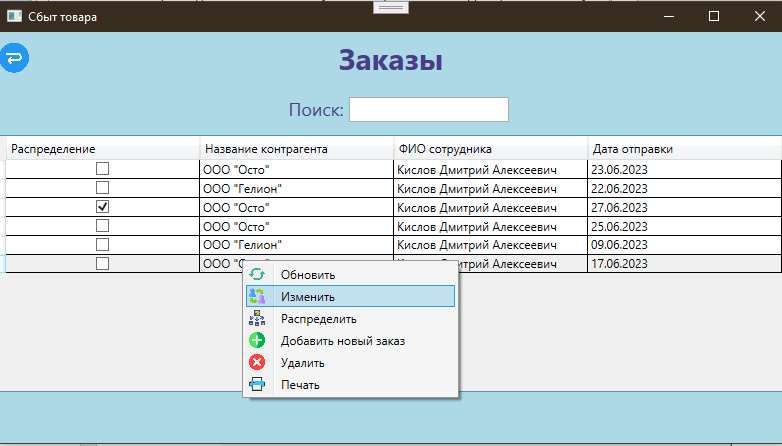


Рис. 43 «Страница заказов с вызовом изменения»

Изменим данные на странице редактирование.

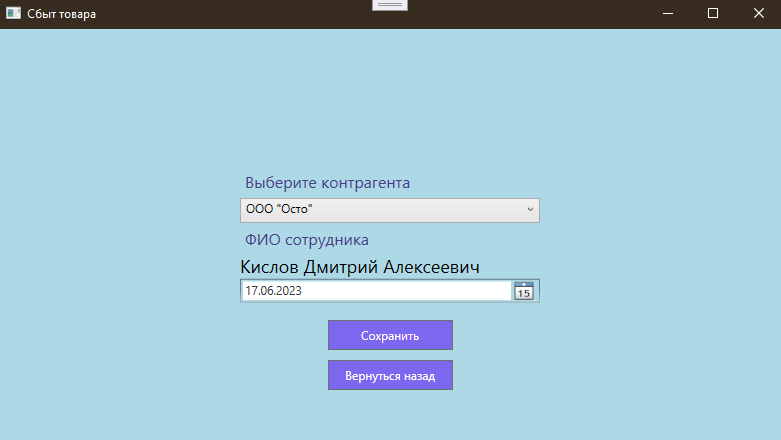


Рис. 44 «Страница редактирование заказов»

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

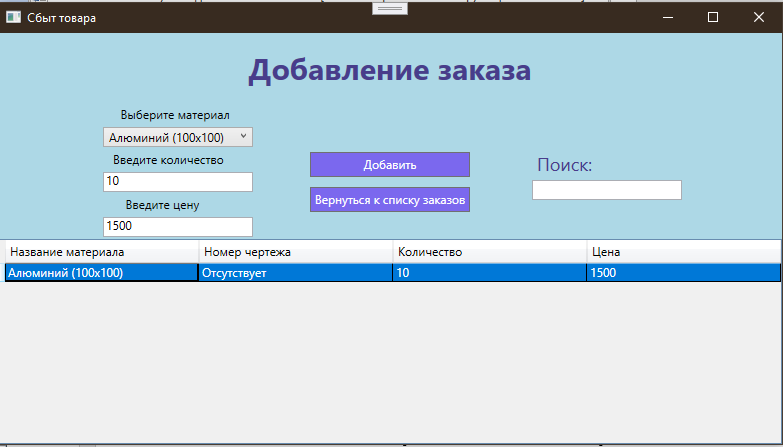


Рис. 45 «Страница редактирование состава заказа»

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»

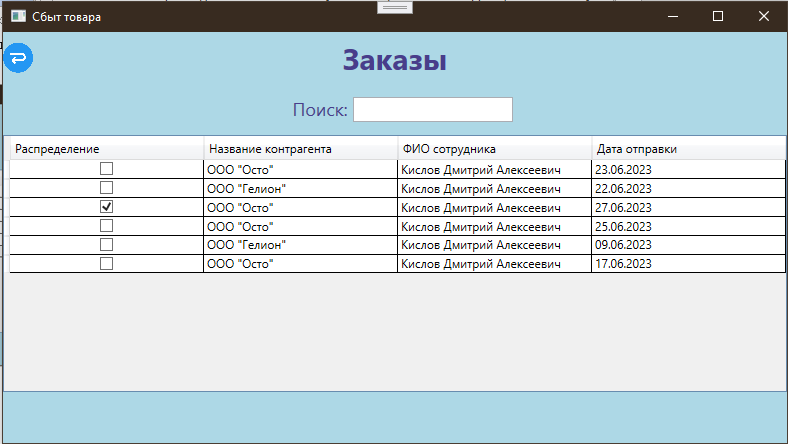


Рис. 46 «Страница заказы после редактирования»

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

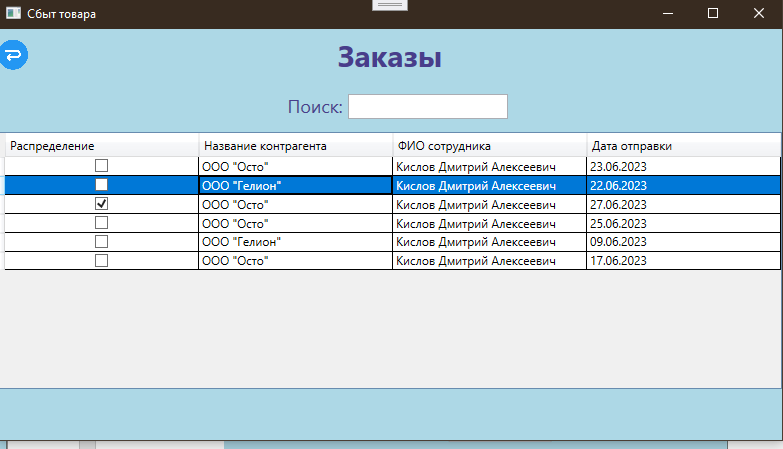


Рис. 47 «Страница заказы переход на состав заказа»

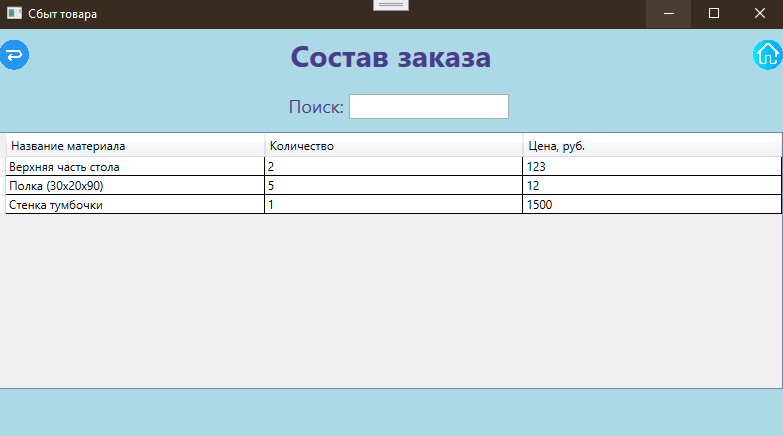


Рис. 48 «Страница состав товара»

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

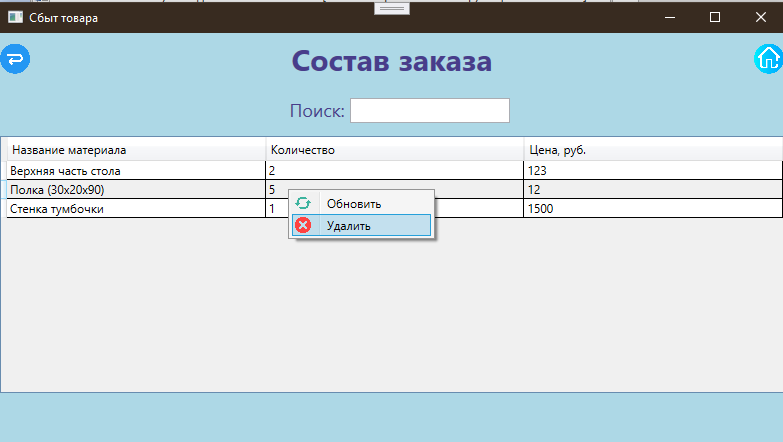


Рис. 49 «Страница состава заказов с удалением записей»

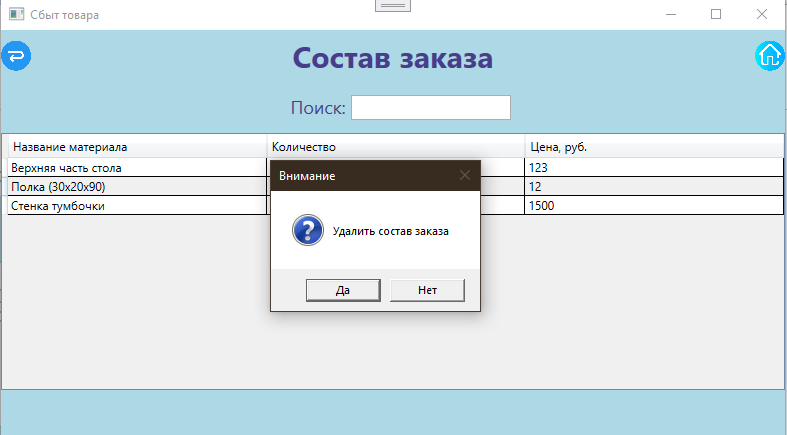


Рис. 50 «Страница состава заказов с уведомление на удаление»

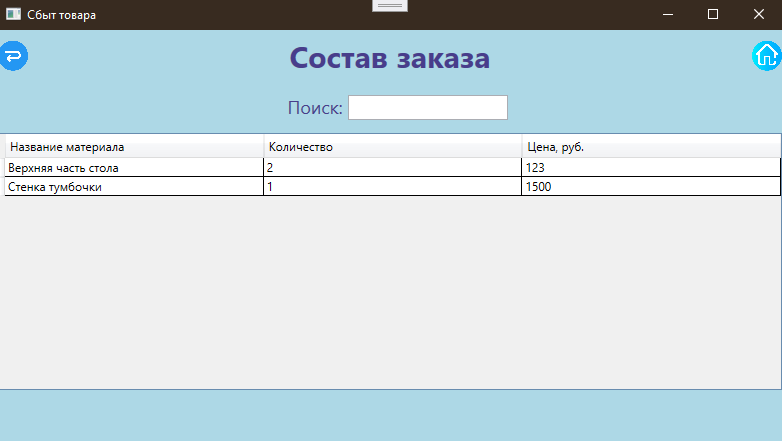


Рис. 51 «Страница состава после удаления»

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.

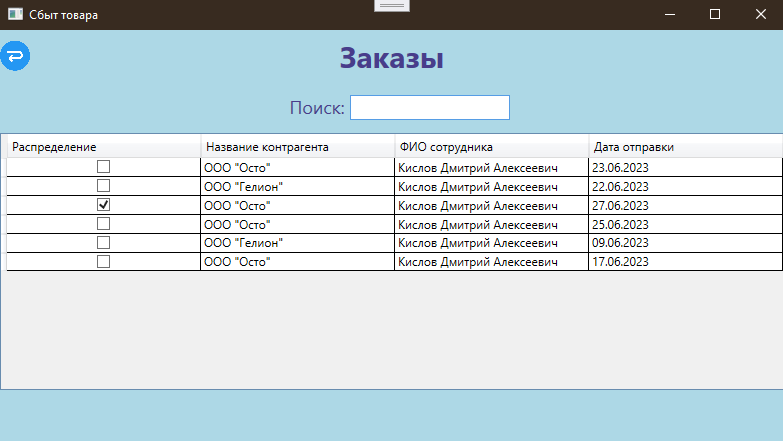


Рис. 53 «Страница заказов после сброса данных»

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

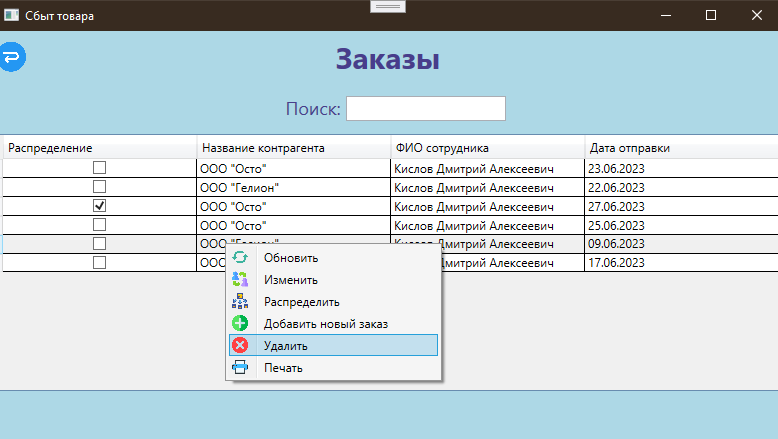


Рис. 54 «Страница заказов с удалением записей»

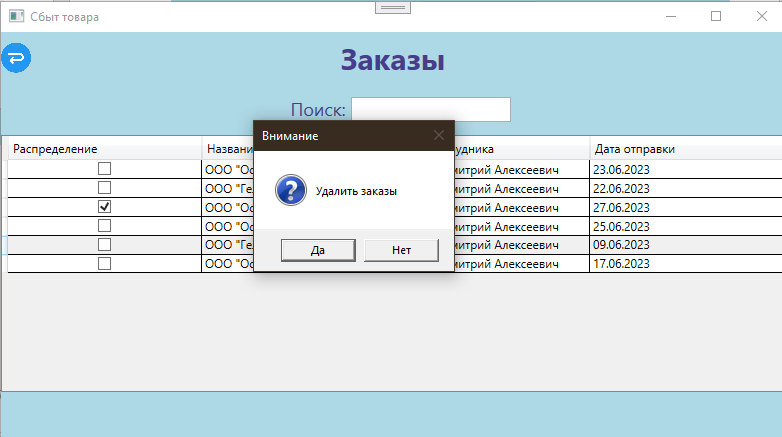


Рис. 55 «Страница заказов с уведомление на удаление»

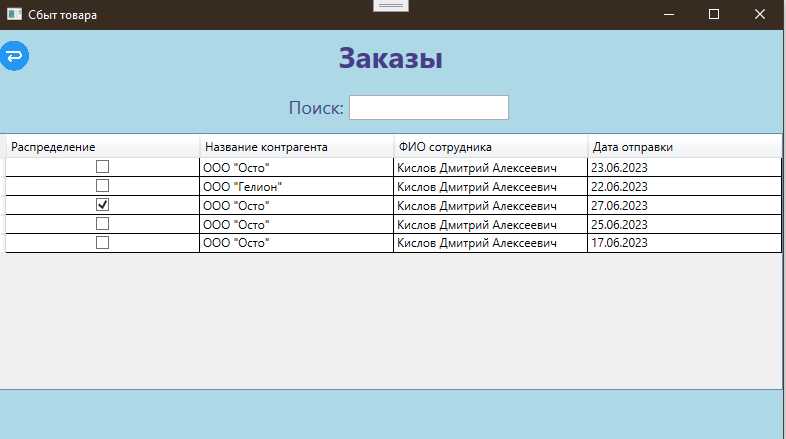


Рис. 56 «Страница заказов после удаления»

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.

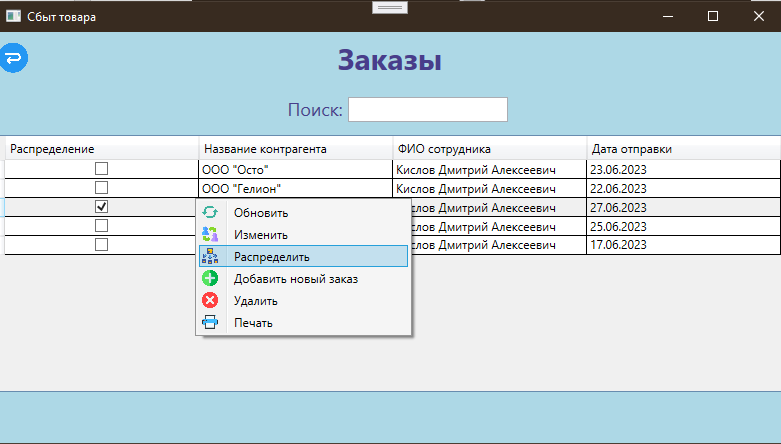


Рис. 57 «Страница заказов с распределением товара»

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».

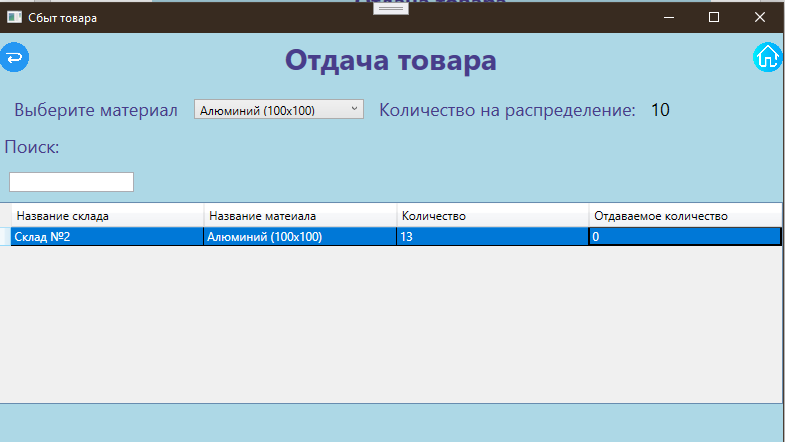


Рис. 58 «Страница отдача товара»

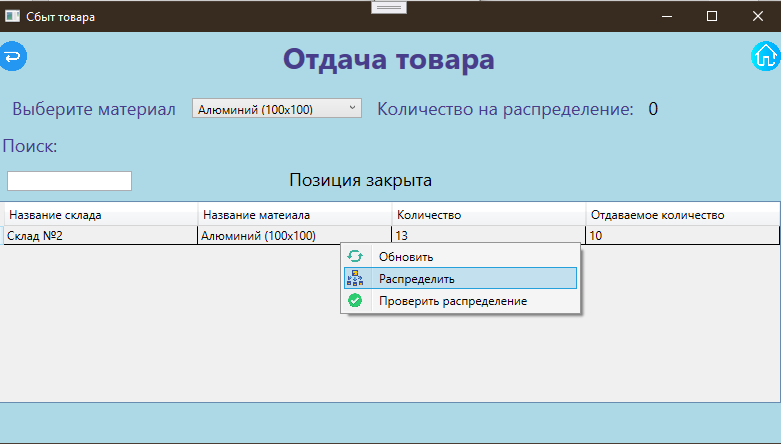


Рис. 59 «Страница отдача товаров распределение»



Рис. 60 «Страница отдача товаров уведомление на распределение»

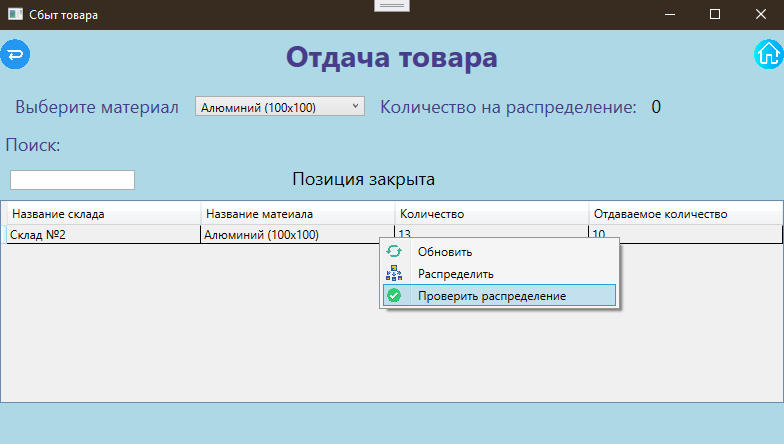


Рис. 61 «Страница отдача товаров проверка распределение»

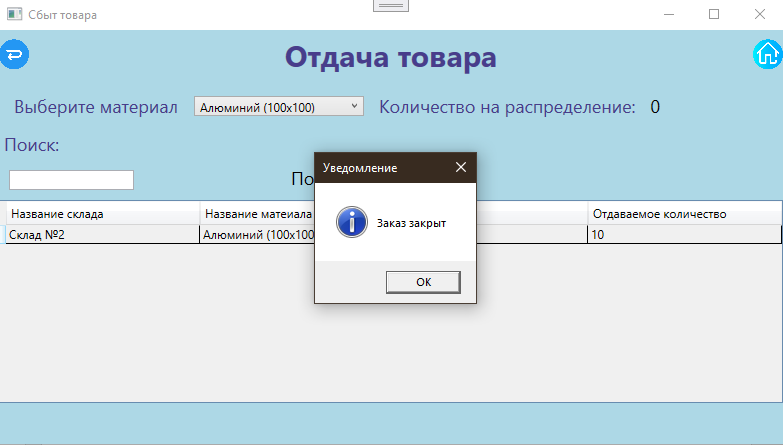


Рис. 62 «Страница отдача товаров уведомление на проверку распределение»

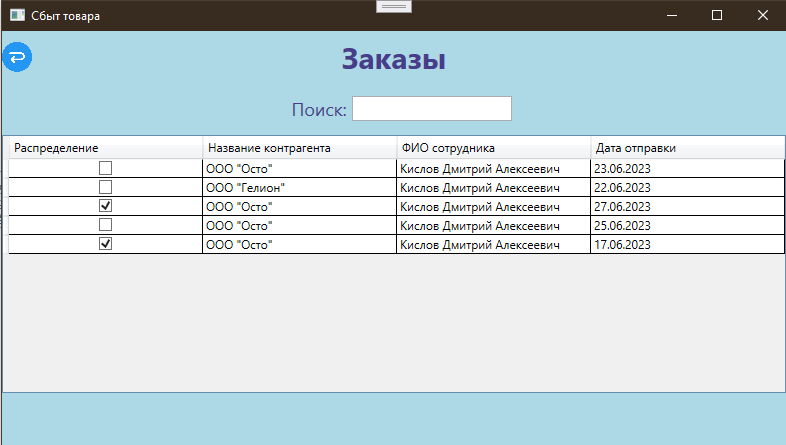


Рис. 63 «Страница заказов после распределения»

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

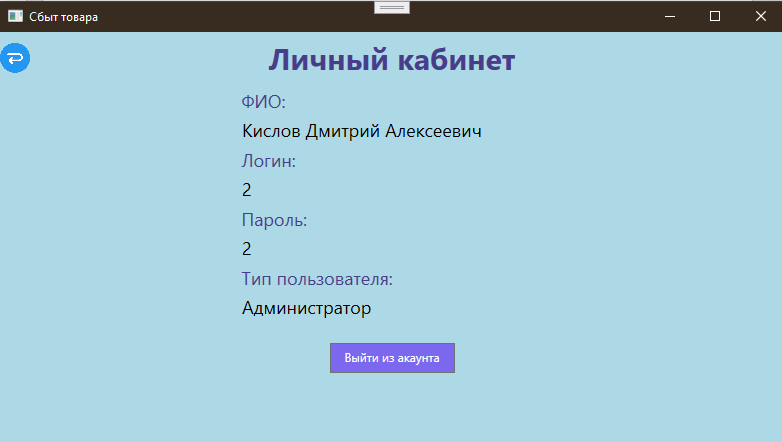


Рис. 64 «Страница с личным кабинетом»

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.

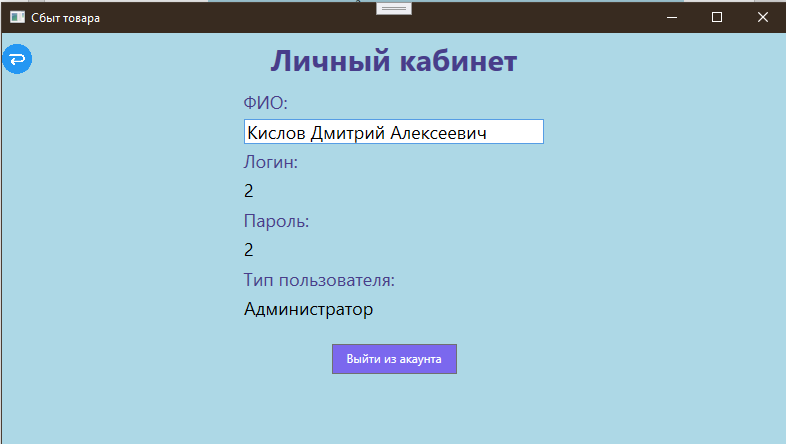


Рис. 65 «Страница редактирование личного кабинета»

На данной странице так же можно выйти из аккаунта.

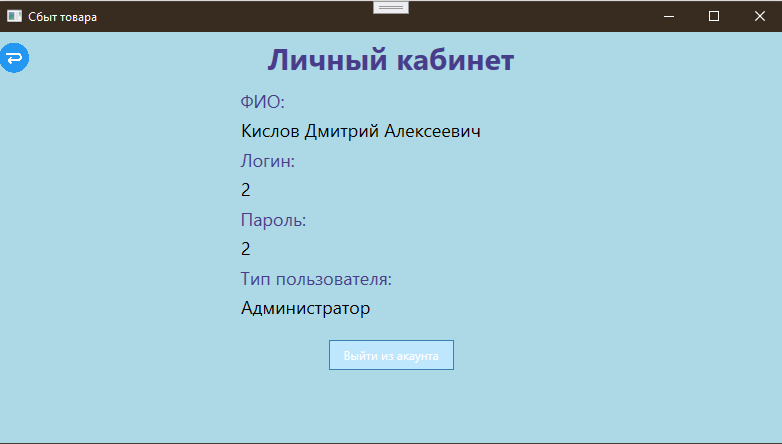


Рис. 66 «Выход из личного кабинета»

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

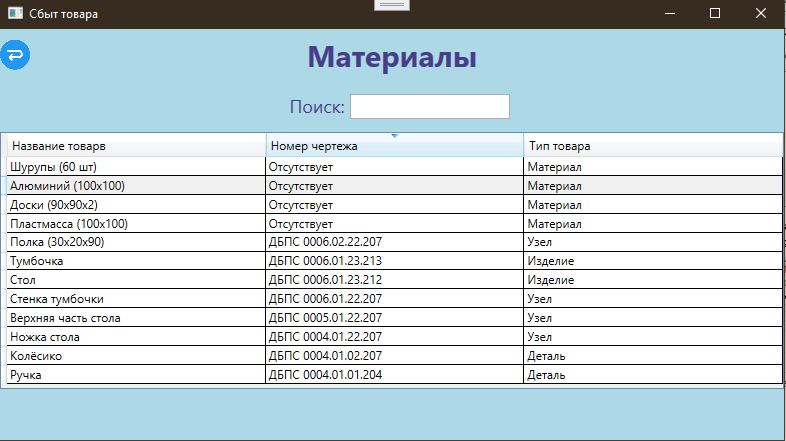


Рис. 67 «Страница материалов»

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.

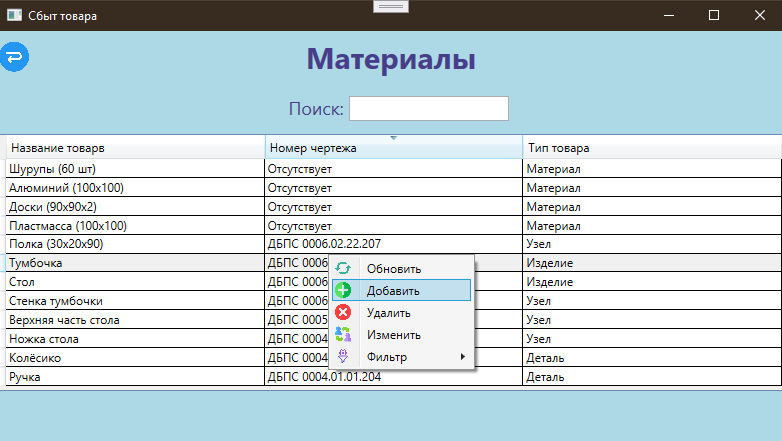


Рис. 68 «Страница материалов с вызовом добавления»

Заполним страницу добавления данными.

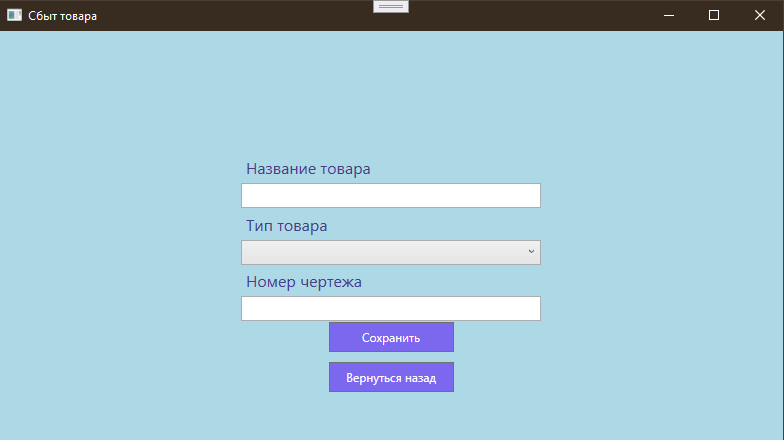


Рис. 69 «Страница добавления материалов»

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

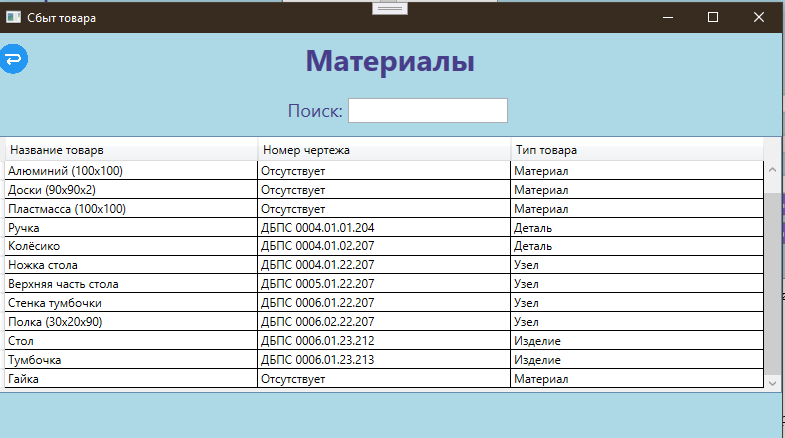


Рис. 70 «Страница материалов после добавления»

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.

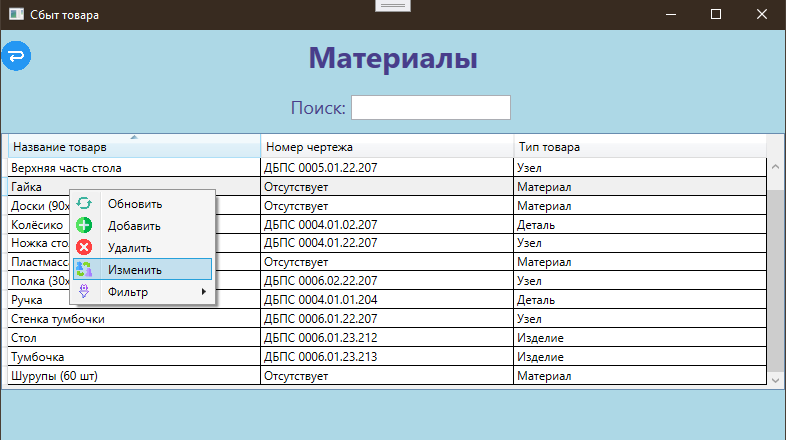


Рис. 71 «Страница материалов с вызовом изменения»

Изменим данные на странице редактирование.

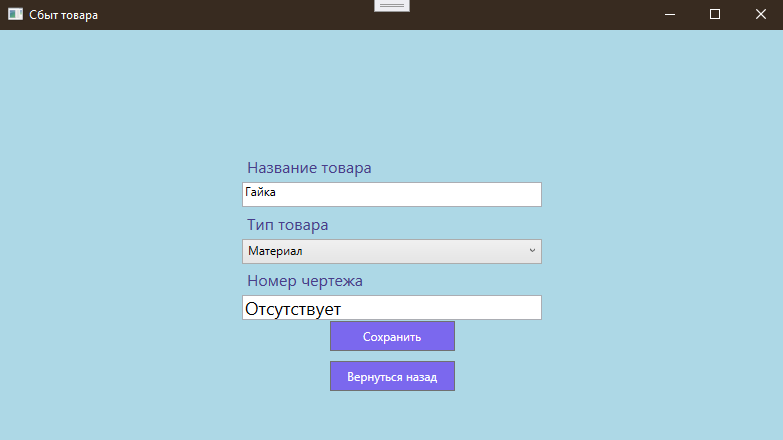


Рис. 72 «Страница редактирование материалов»

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

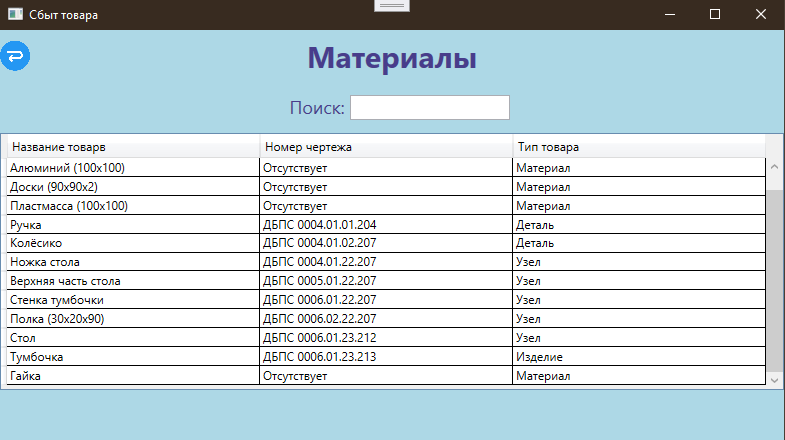


Рис. 73 «Страница материалов после редактирования»

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

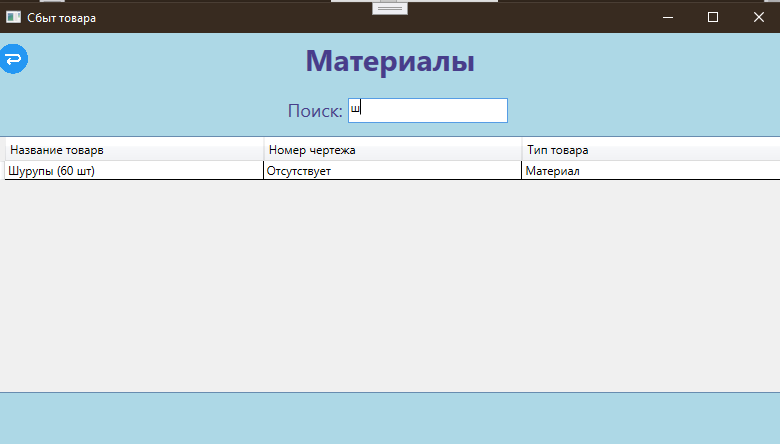


Рис. 74 «Страница материалов с поиском»

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

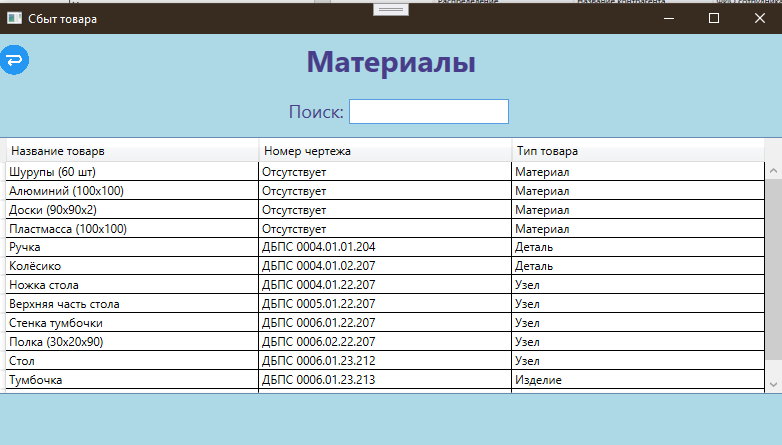


Рис. 75 «Страница материалов после сброса данных»

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

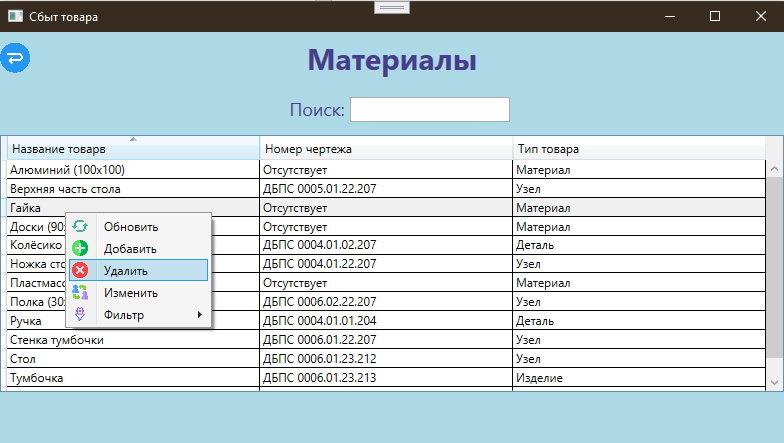


Рис. 76 «Страница материалов с удалением записей»

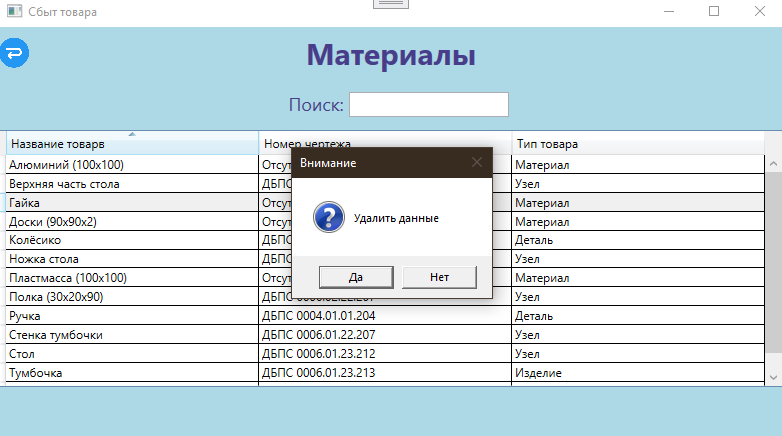


Рис. 77 «Страница материалов с уведомление на удаление»

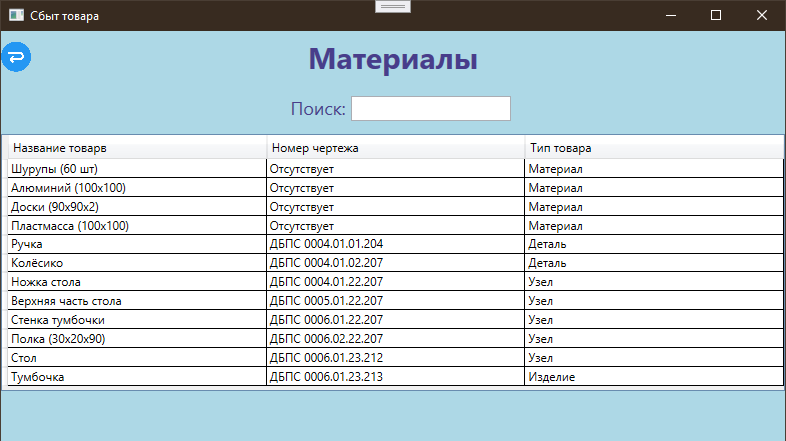


Рис. 78 «Страница материалов после удаления»

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».

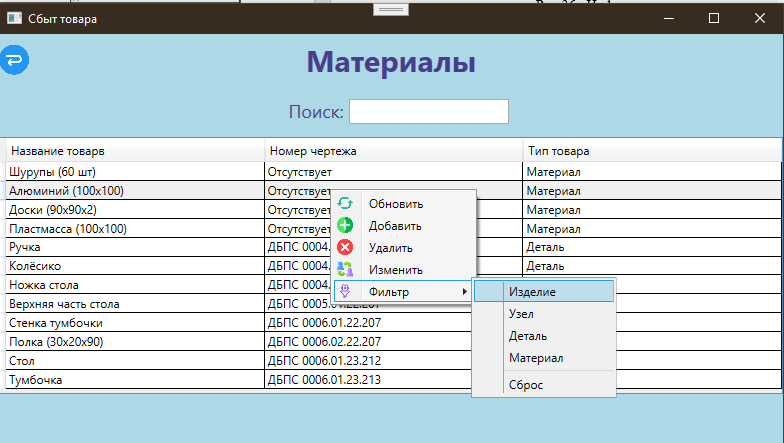


Рис. 79 «Страница материалов с фильтром»

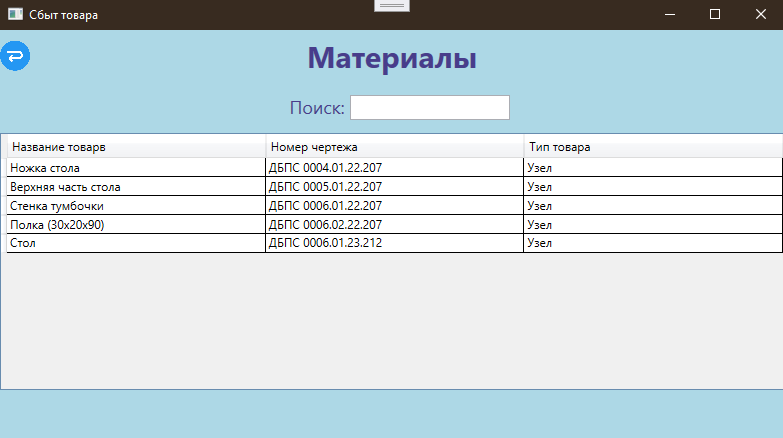


Рис. 80 «Страница материалов после фильтрации»

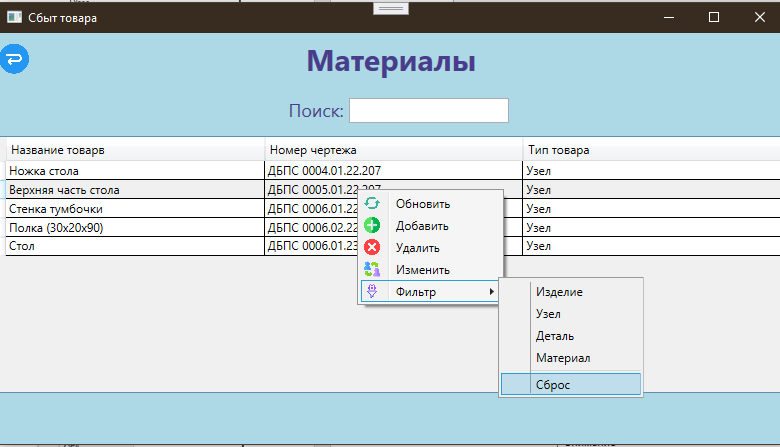


Рис. 81 «Страница материалов сброс фильтрации»

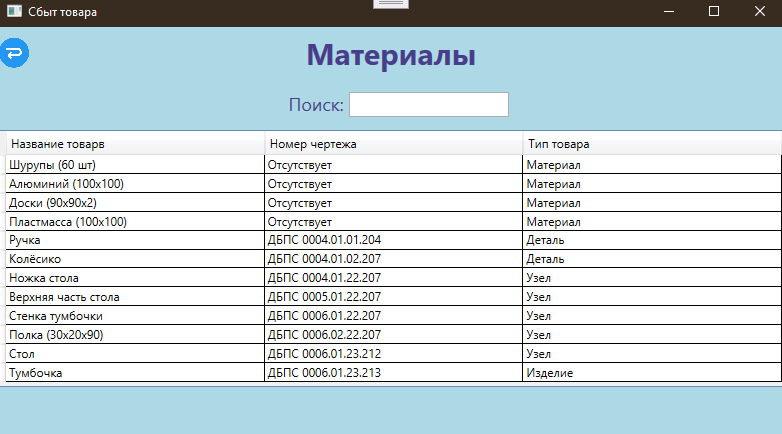


Рис. 82 «Страница материалов после сброса фильтрации»

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

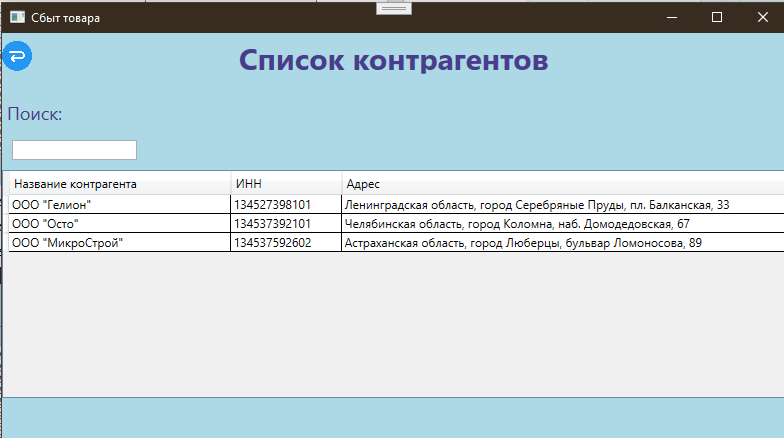


Рис. 83 «Страница контрагентов»

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.

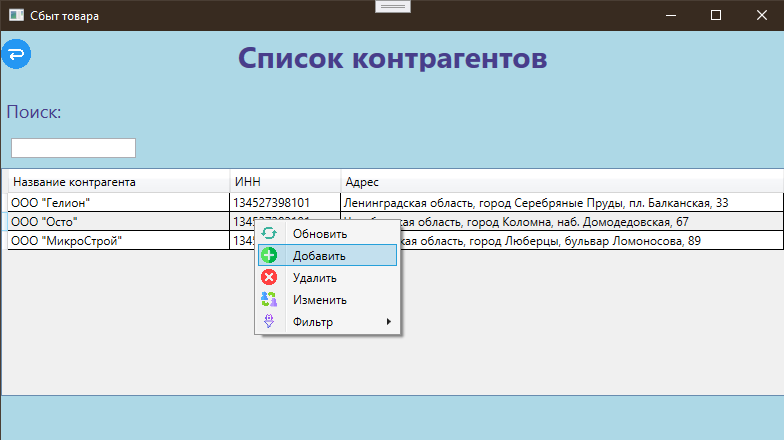


Рис. 84 «Страница контрагентов с вызовом добавления»

Заполним страницу добавления данными.

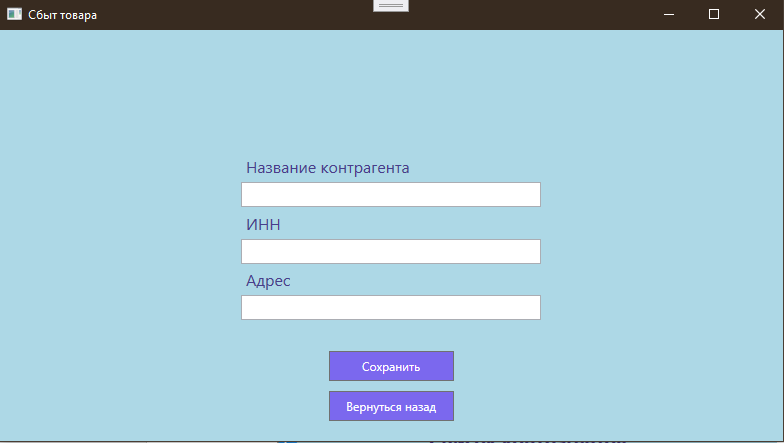


Рис. 85 «Страница добавления контрагентов»

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.

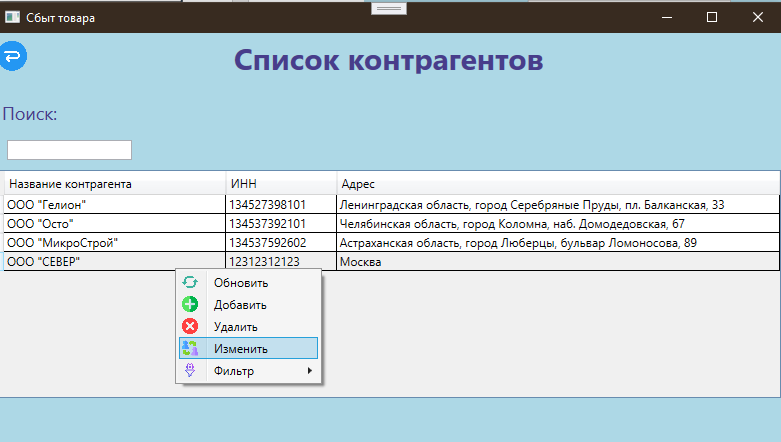
****

Рис. 87 «Страница контрагентов с вызовом изменения»

Изменим данные на странице редактирование.

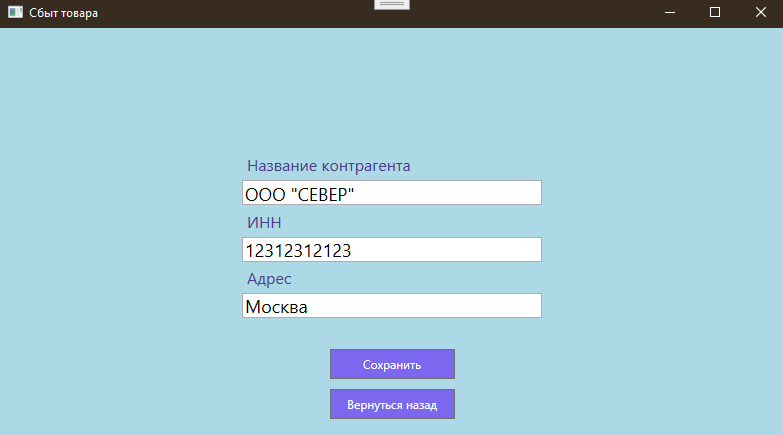


Рис. 88 «Страница редактирование контрагентов»

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

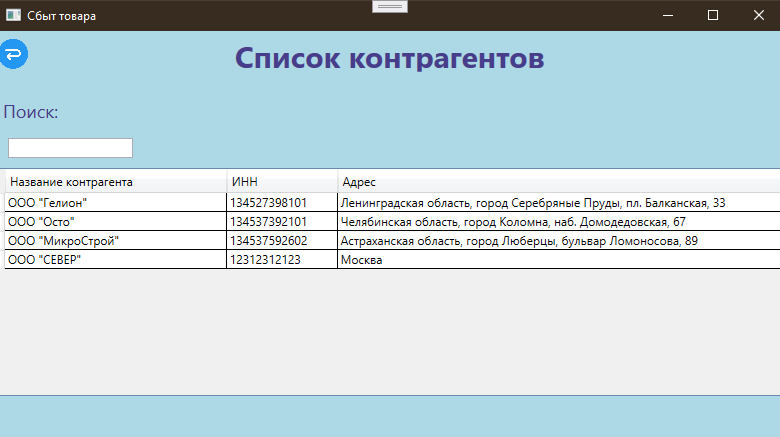


Рис. 89 «Страница контрагентов после редактирования»

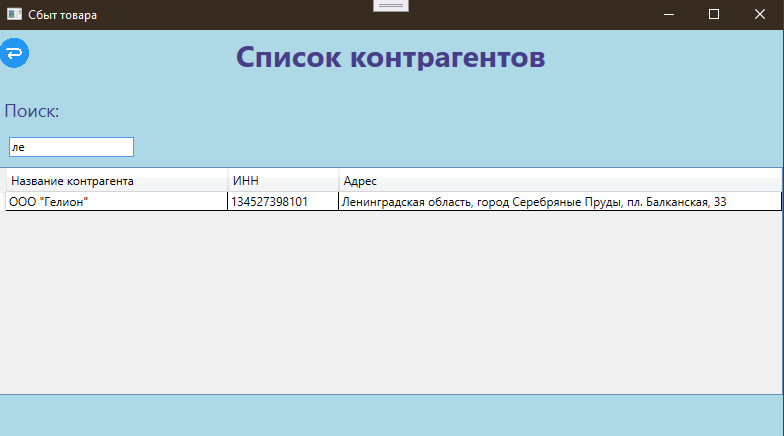
В приложении присутствует поиск для нахождения данных по атрибутам таблицы. В него нужно ввести текст, которые присутствуют в Название контрагента, ИНН, Адрес. 

Рис. 90 «Страница контрагентов с поиском»

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

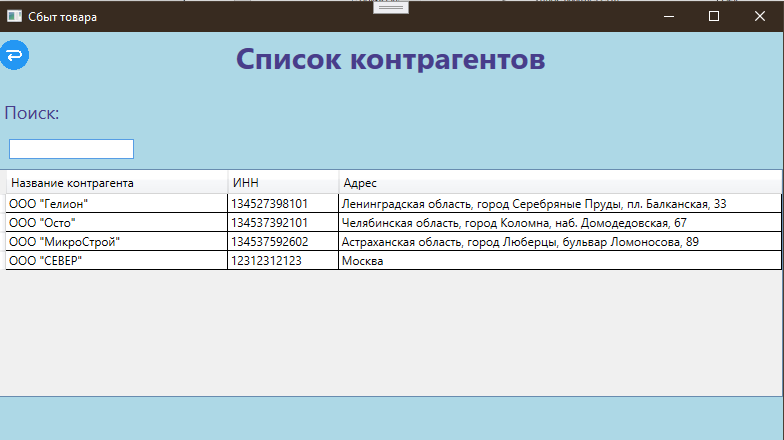


Рис. 91 «Страница контрагентов после сброса данных»

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

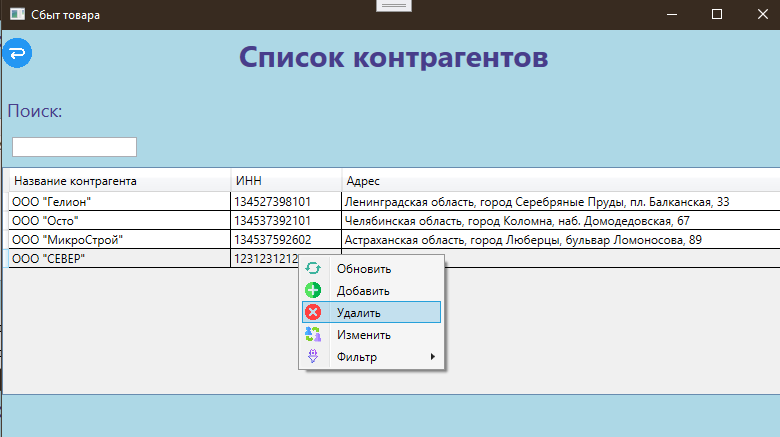


Рис. 92 «Страница контрагентов с удалением записей»

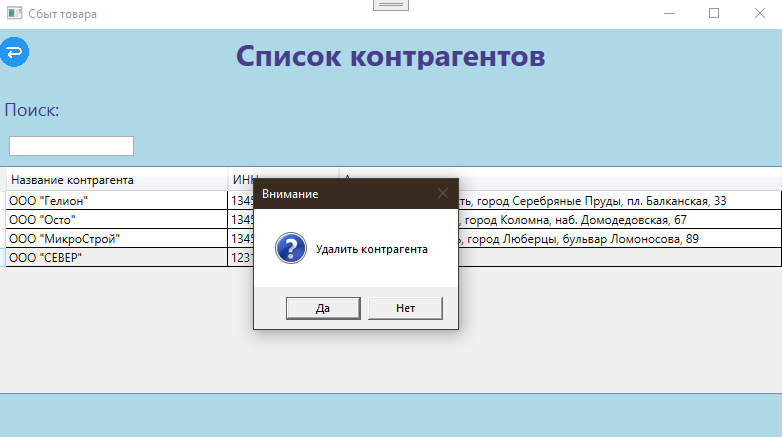


Рис. 93 «Страница контрагентов с уведомление на удаление»

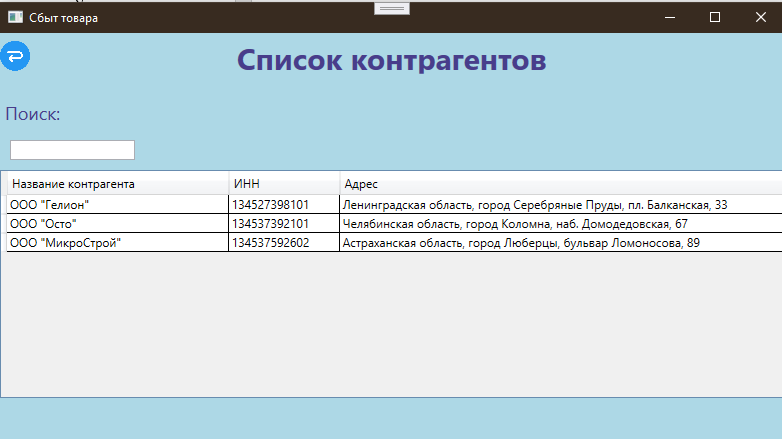


Рис. 94 «Страница контрагентов после удаления»

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»

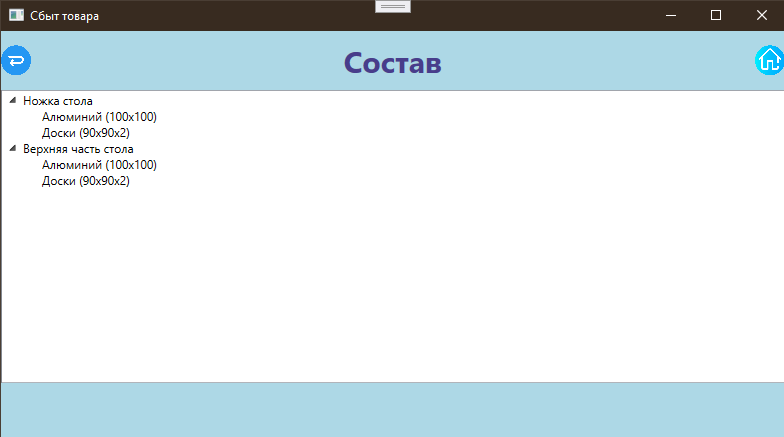


Рис. 97 «Страница материалов хранящихся на складе просмотр иерархии»

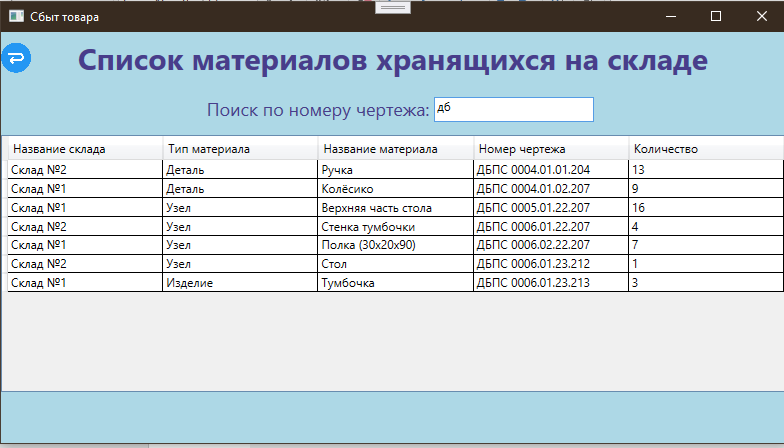
В приложении присутствует поиск для нахождения данных по атрибутам таблицы. В него нужно ввести текст, которые присутствуют в номере чертежа. 

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».

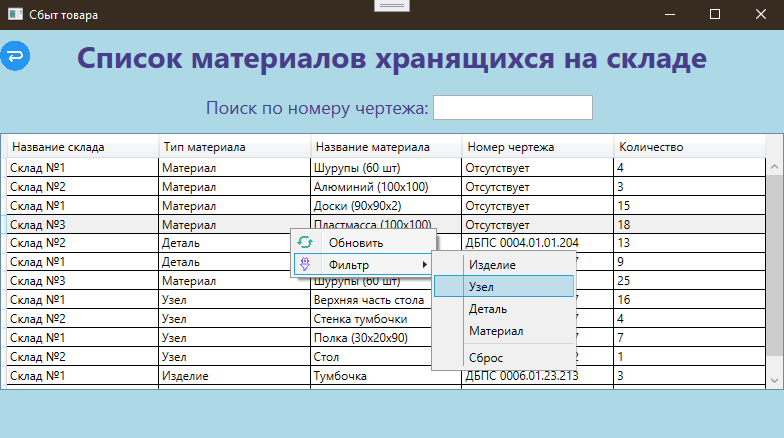


Рис. 100 «Страница материалов хранящихся на складе с фильтром»

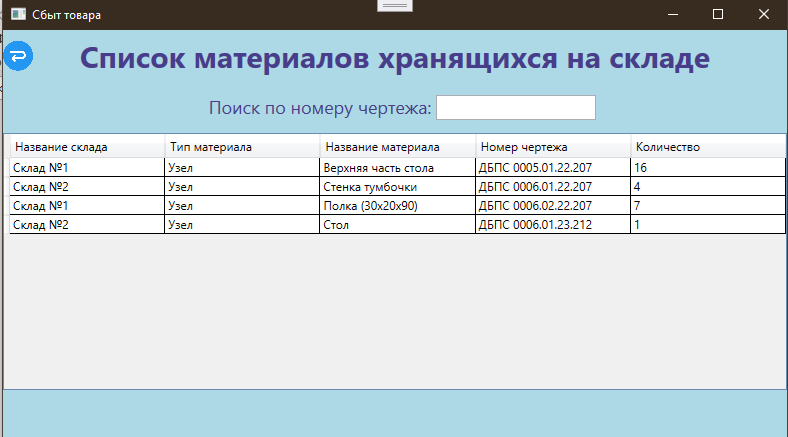


Рис. 101 «Страница материалов хранящихся на складе после фильтрации»

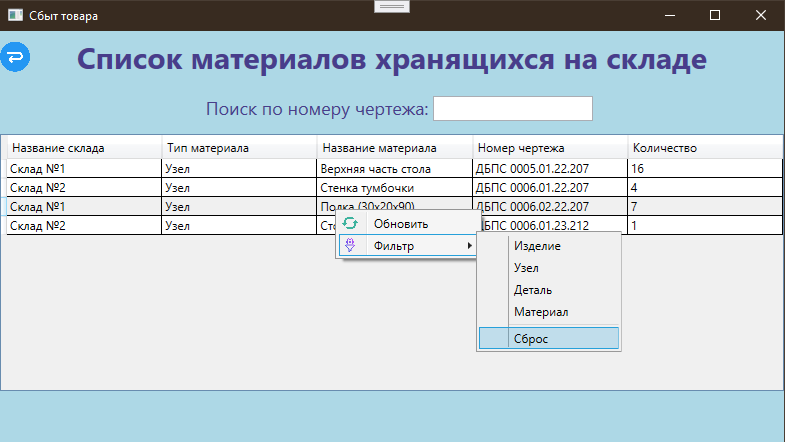


Рис. 102 «Страница материалов хранящихся на складе сброс фильтрации»



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.

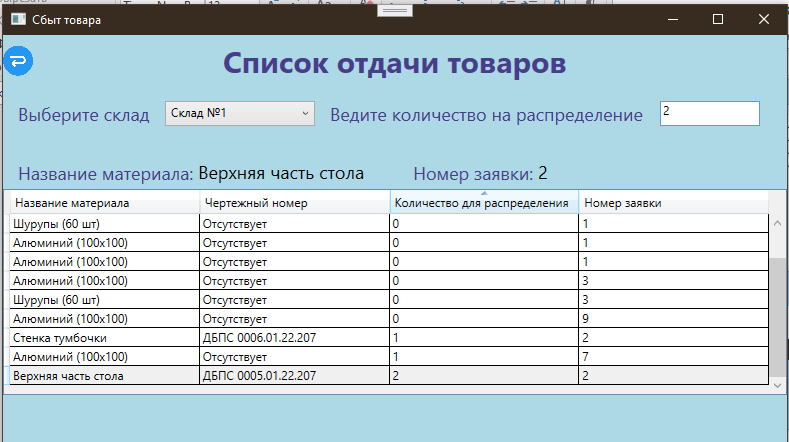


Рис. 105 «Страница отдачи товаров выбор товара на распределение»

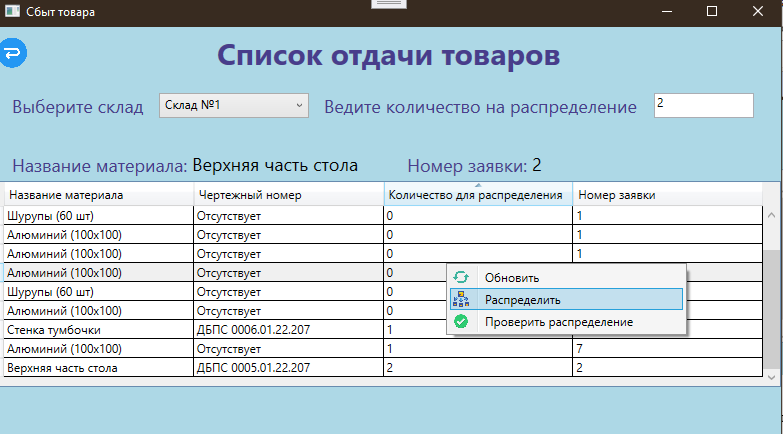


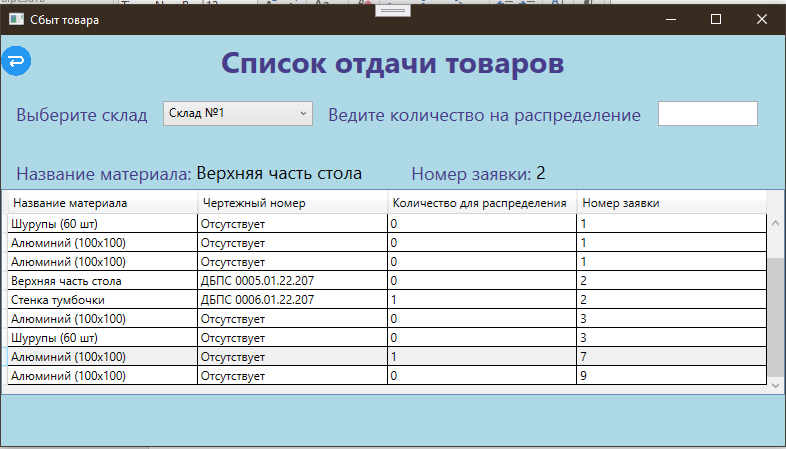
Рис. 106 «Страница отдачи товаров распределение товара»

Рис. 107 «Страница отдачи товаров после распределения»

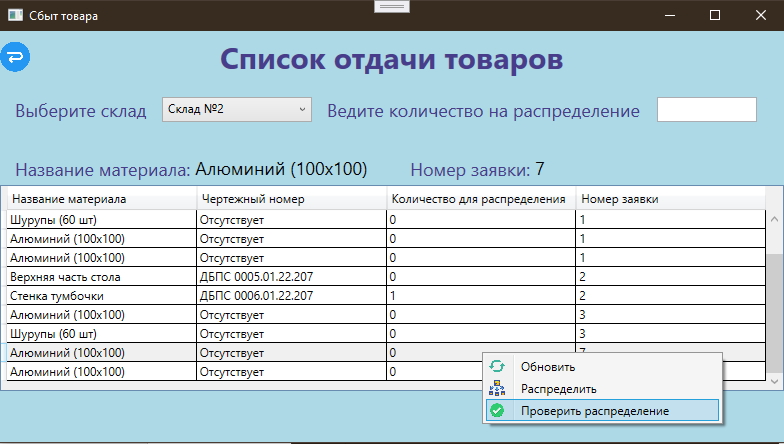


Рис. 108 «Страница отдачи товаров проверка распределения»

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

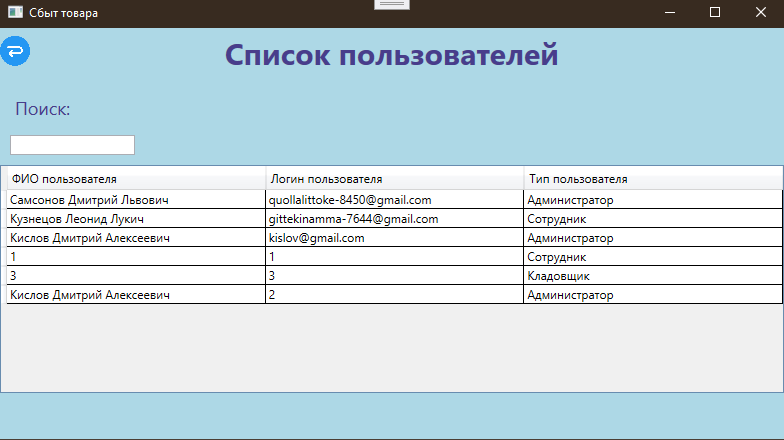


Рис. 109 «Страница пользователей»

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.

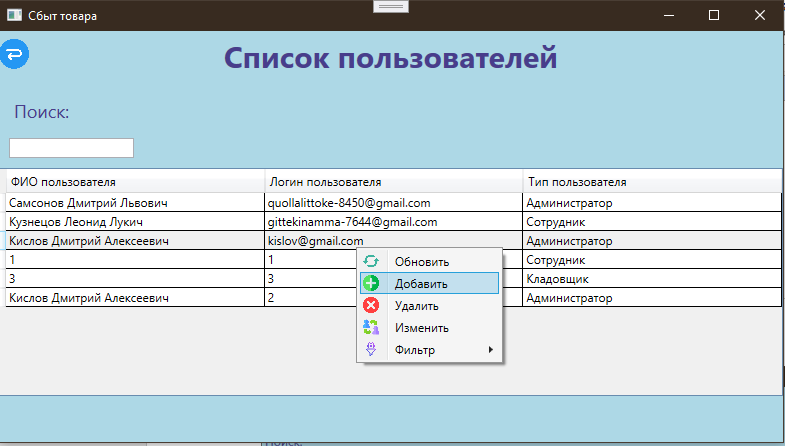


Рис. 110 «Страница пользователей с вызовом добавления»

Заполним страницу добавления данными.

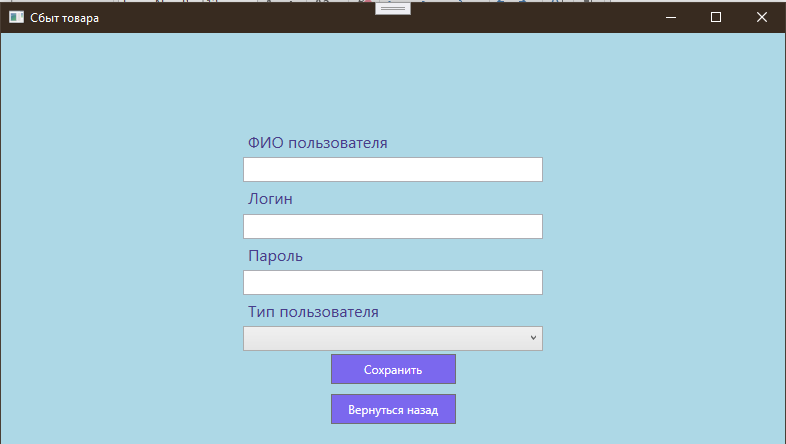


Рис. 111 «Страница добавления пользователей»

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

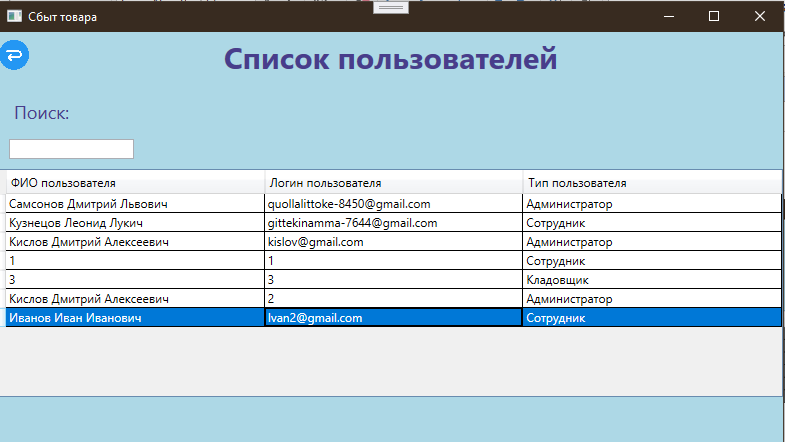


Рис. 112 «Страница пользователей после добавления»

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.

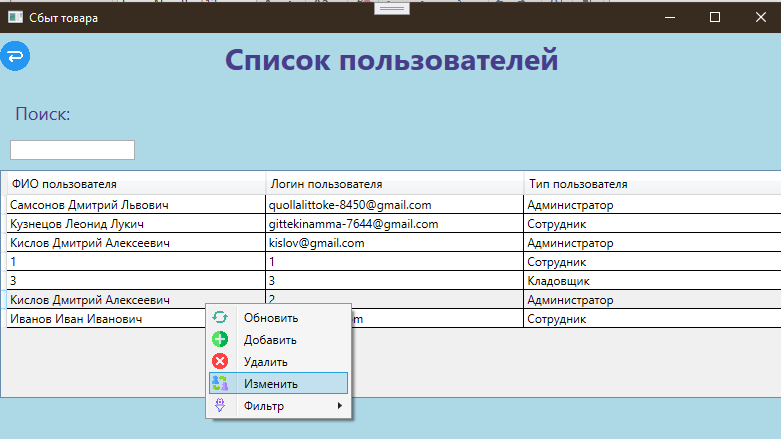
****

Рис. 113 «Страница пользователей с вызовом изменения»

Изменим данные на странице редактирование.

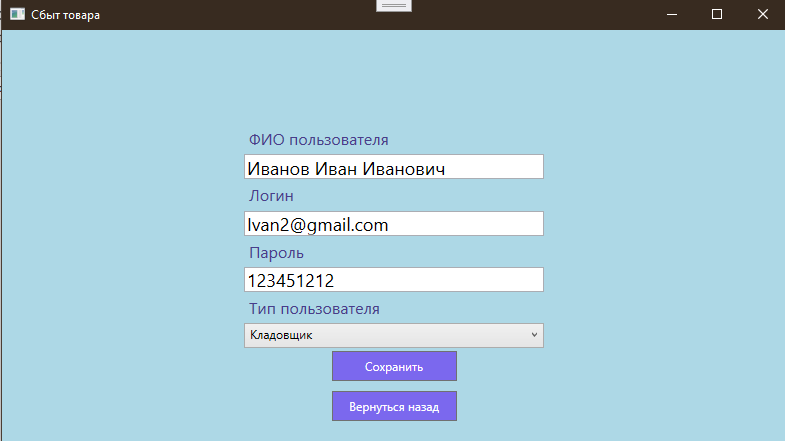


Рис. 114 «Страница редактирование пользователей»

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

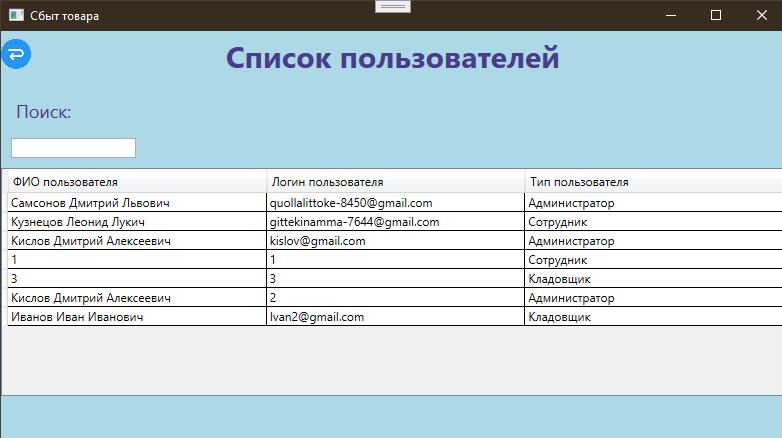


Рис. 115 «Страница пользователей после редактирования»

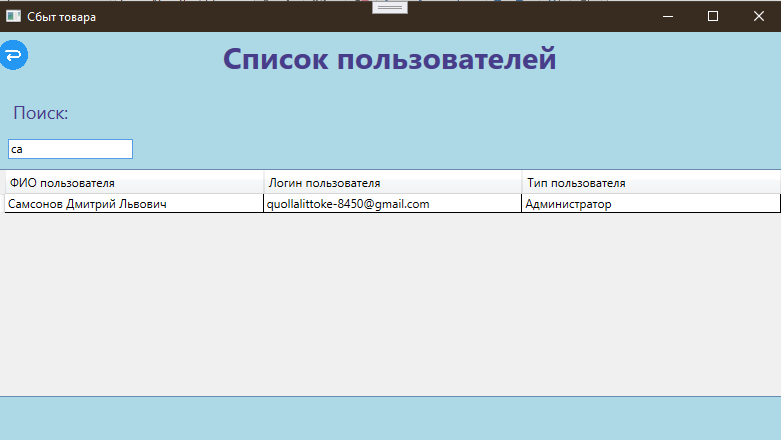
В приложении присутствует поиск для нахождения данных по атрибутам таблицы. В него нужно ввести текст, которые присутствуют в ФИО, Логин пользователя, и Тип пользователя. 

Рис. 116 «Страница пользователей с поиском»

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

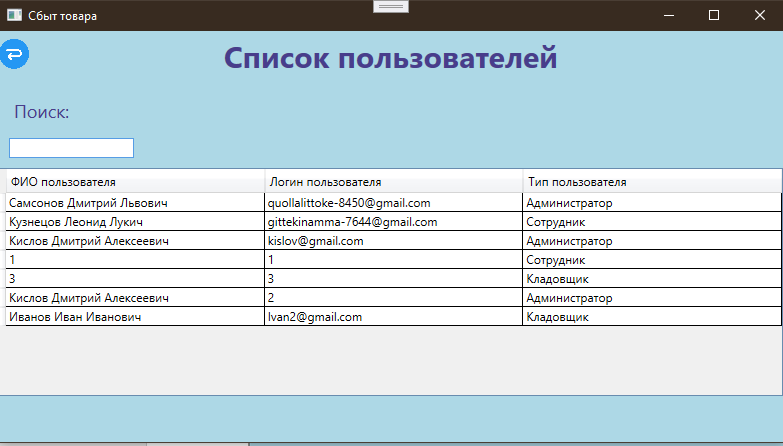


Рис. 117 «Страница пользователей после сброса данных»

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

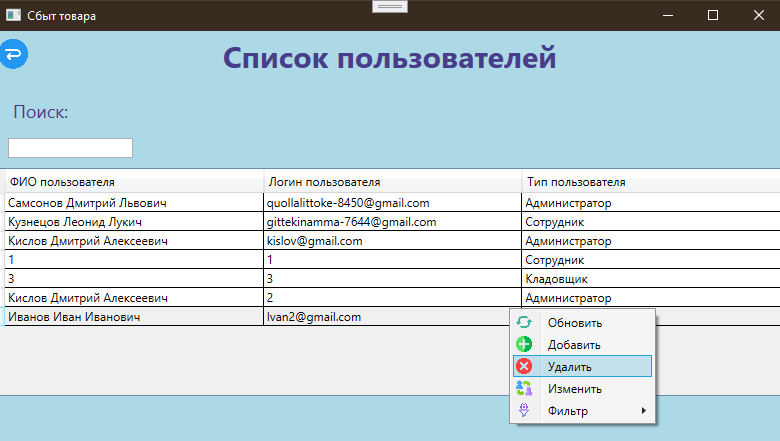


Рис. 118 «Страница пользователей с удалением записей»

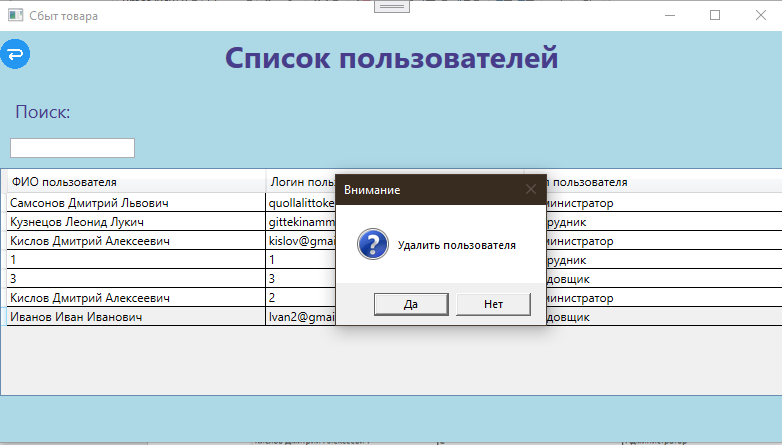


Рис. 119 «Страница пользователей с уведомление на удаление»

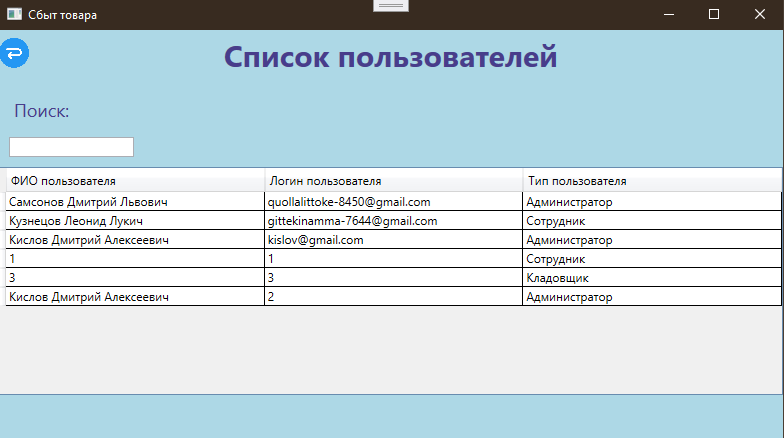


Рис. 120 «Страница пользователей после удаления»

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».

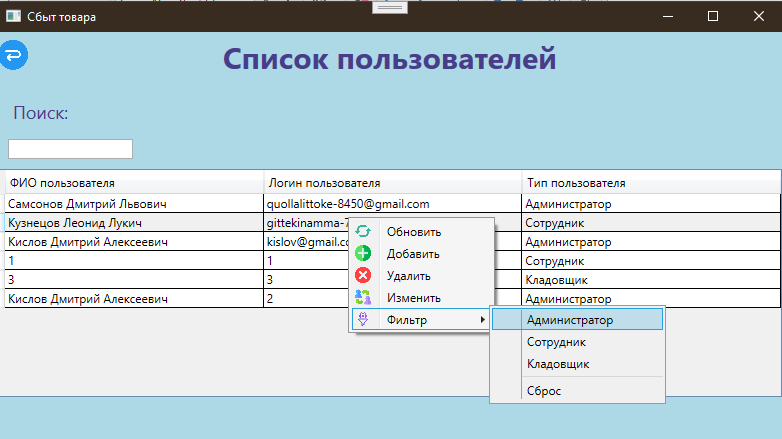


Рис. 121 «Страница пользователей с фильтром»

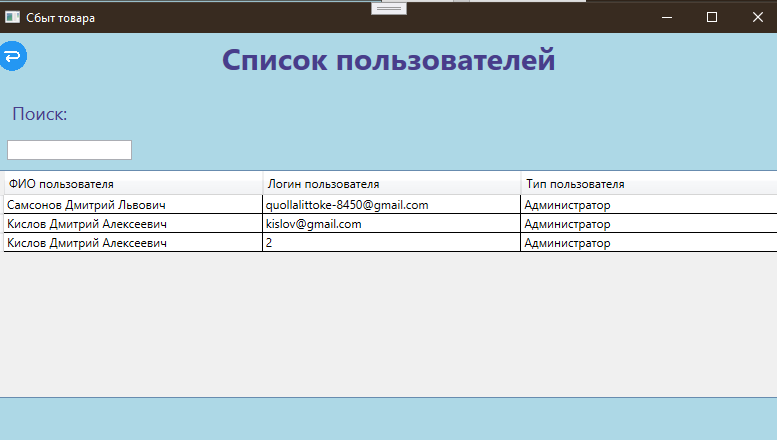


Рис. 122 «Страница пользователей после фильтрации»

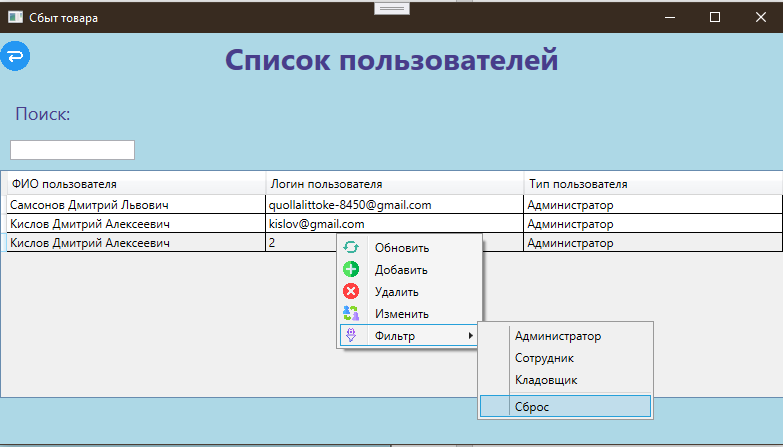


Рис. 123 «Страница пользователей сброс фильтрации»

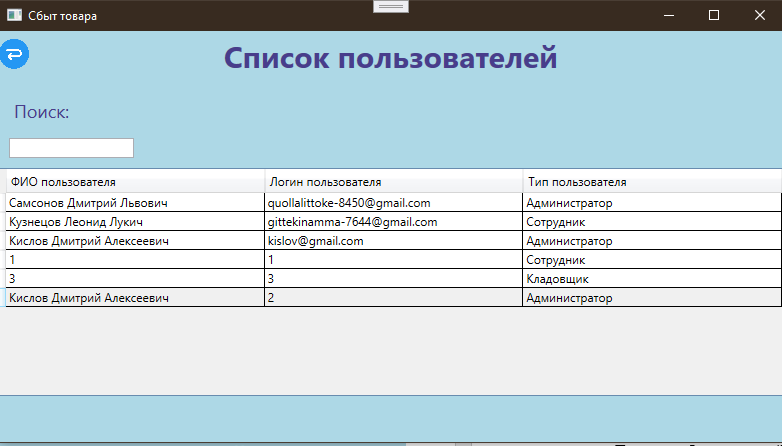


Рис. 124 «Страница пользователей после сброса фильтрации»

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.

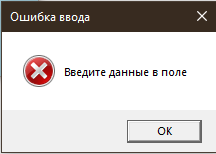


Рис.118 «Попытка добавления заказа»

Подтверждение удаления нескольких заказов.

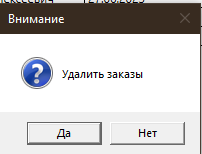


Рис.119 «Подтверждение удаления нескольких заказов»

Информация после удаления нескольких пунктов.

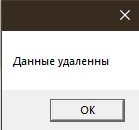


Рис.120 «Информация при удалении нескольких пунктов»

Информация после редактирования проекта.

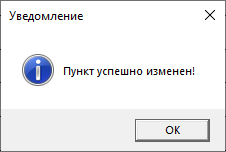


Рис.121 «Уведомление после изменения»

Информация после распределения товара.

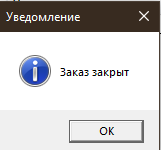


Рис.122 «Уведомление после распределения»

## 3.6. Отладка ПО

Участок кода с добавление материала в таблицу, работающий некорректно:

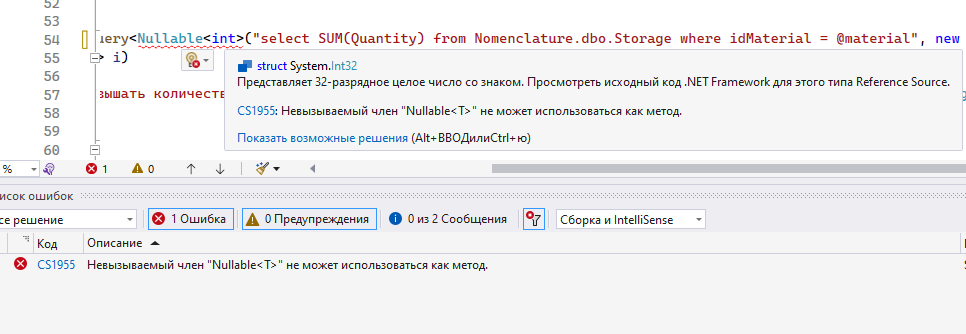


Рис.123 «Неявное преобразование типов»

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:

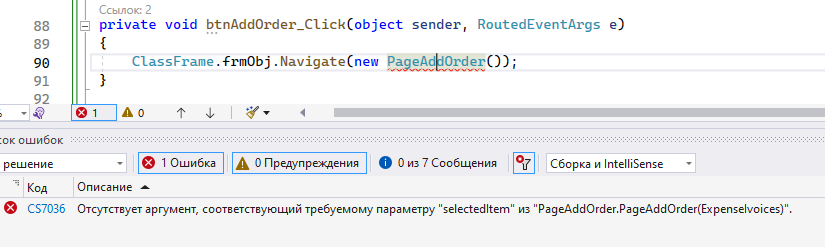


Рис.125 «Отсутствие аргумента, требуемой страницы»

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

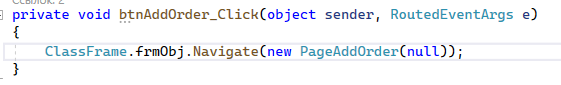


Рис.126 «Добавление пустого аргумента, требуемой странице»

## 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 кнопка, в которой хранится нужная нам страница для добавления данных в базу 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 | Удаление запись. |
| Краткое изложение теста | Попытка удаление записи из таблице «Контрагентов». |
| Этапы теста | Нажать на таблицу с помощью ПКМ и выбрать из списка пункт «Удалить» |
| Тестовые данные | Название контрагенты: ООО «Вера», ИНН: 134537592602, Адрес: Москва. |
| Ожидаемый результат | Предупреждающее сообщение об удалении данных в таблице «Контрагентов». |
| Фактический результат | Запись удалена! |
| Предварительное условие | На главном меню пользовательского интерфейса должна быть кнопка, в которой хранится нужная нам страница для удаления данных из базы данных. |
| Постусловие | Запись была удалена. |
| Статус (Зачет/Незачет) | Зачет |

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 Контрагент). |
| Тестовые данные | В поисковую строку введём значение «О6» |
| 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.

# Приложение

***MainWindow*.xaml:**

<Grid>

<Grid.ColumnDefinitions>

<ColumnDefinition/>

</Grid.ColumnDefinitions>

<Frame Name="FrameSales" NavigationUIVisibility="Hidden" />

</Grid>

***MainWindow*.xaml.cs:**

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

ClassFrame.db = new SalesOfMaterialEntities();

ClassFrame.frmObj = FrameSales;

ClassFrame.frmObj.Navigate(new PageAuthorization());

}

}

***AddUser.xaml:***

<Grid>

<StackPanel VerticalAlignment="Center" HorizontalAlignment="Center">

<Label Content="ФИО пользователя" Style="{StaticResource lblText2}"/>

<TextBox Name="txtFIO" Width="300" Height="25" FontSize="18" Text="{Binding FIO}"/>

<Label Content="Логин" Style="{StaticResource lblText2}"/>

<TextBox Name="txtLogin" Width="300" Height="25" FontSize="18" Text="{Binding Login\_Employee}"/>

<Label Content="Пароль" Style="{StaticResource lblText2}"/>

<TextBox Name="txtPassword" Width="300" Height="25" FontSize="18" Text="{Binding Password\_Employee}"/>

<Label Content="Тип пользователя" Style="{StaticResource lblText2}"/>

<ComboBox Name="cmbType"

Height="25"

DisplayMemberPath="Nazv\_type"

SelectedValue="{Binding IdTypesEmployee}"

SelectedValuePath="IdTypesEmployee"/>

</StackPanel>

<StackPanel VerticalAlignment="Bottom" HorizontalAlignment="Center" Margin="0,0,0,10">

<Button Name="btnAdd" Content="Сохранить" Click="btnAdd\_Click" Style="{StaticResource btn1}"/>

<Button Name="btnBack" Content="Вернуться назад" Click="btnBack\_Click" Margin="10" Style="{StaticResource btn1}"/>

</StackPanel>

</Grid>

***AddUser.xaml.cs:***

public partial class AddUser : Page

{

public Employee \_currentItem = new Employee();

public AddUser(Employee selectedItem)

{

InitializeComponent();

if (selectedItem != null)

{

\_currentItem = selectedItem;

}

DataContext = \_currentItem;

cmbType.ItemsSource = ClassFrame.db.TypesEmployee.ToList();

}

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

{

if (txtFIO.Text == "" && txtLogin.Text == "" && txtPassword.Text == "" && cmbType.Text == "")

{

MessageBox.Show("Введите данные в поле", "Ошибка ввода", MessageBoxButton.OK, MessageBoxImage.Error);

return;

}

if (\_currentItem.IdEmployee == 0) ClassFrame.db.Employee.Add(\_currentItem);

try

{

ClassFrame.db.SaveChanges();

ClassFrame.frmObj.Navigate(new User());

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

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

{

ClassFrame.frmObj.Navigate(new User());

}

} }

***AddСounterparties.xaml:***

<Grid>

<StackPanel VerticalAlignment="Center" HorizontalAlignment="Center">

<Label Content="Название контрагента" Style="{StaticResource lblText2}"/>

<TextBox Name="txtNazv" Width="300" Height="25" FontSize="18" Text="{Binding Nazv\_Сounterparties}"/>

<Label Content="ИНН" Style="{StaticResource lblText2}"/>

<TextBox Name="txtINN" Width="300" Height="25" FontSize="18" Text="{Binding INN}"/>

<Label Content="Адрес" Style="{StaticResource lblText2}"/>

<TextBox Name="txtAdress" Width="300" Height="25" FontSize="18" Text="{Binding Adress\_Сounterparties}"/>

</StackPanel>

<StackPanel VerticalAlignment="Bottom" HorizontalAlignment="Center" Margin="0,0,0,10">

<Button Name="btnAdd" Content="Сохранить" Click="btnAdd\_Click" Style="{StaticResource btn1}"/>

<Button Name="btnBack" Content="Вернуться назад" Click="btnBack\_Click" Margin="10" Style="{StaticResource btn1}"/>

</StackPanel>

</Grid>

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();

}

***AddСounterparties.xaml.cs:***

public partial class AddСounterparties : Page

{

public Classes.Сounterparties \_currentItem = new Classes.Сounterparties();

public AddСounterparties(Classes.Сounterparties selectedItem)

{

InitializeComponent();

if (selectedItem != null)

{

\_currentItem = selectedItem;

}

DataContext = \_currentItem;

}

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

{

if (txtNazv.Text == "" && txtINN.Text == "" && txtAdress.Text == "")

{

MessageBox.Show("Введите данные в поле", "Ошибка ввода", MessageBoxButton.OK, MessageBoxImage.Error);

return;

}

if (\_currentItem.IdСounterparties == 0) ClassFrame.db.Сounterparties.Add(\_currentItem);

try

{

ClassFrame.db.SaveChanges();

ClassFrame.frmObj.Navigate(new Сounterparties());

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

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

{

}

}

***DistributionList.xaml:***

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="1\*"/>

<RowDefinition Height="2\*"/>

<RowDefinition Height="4\*"/>

<RowDefinition Height="1\*"/>

</Grid.RowDefinitions>

<StackPanel Grid.Row="0" VerticalAlignment="Center" HorizontalAlignment="Left">

<Button x:Name="Back1" Click="Back1\_Click" Width="30" Height="30" BorderThickness="0">

<Button.Background>

<ImageBrush ImageSource="/Photo/iconback.png"/>

</Button.Background>

</Button>

</StackPanel>

<Label Content="Список отдачи товаров" Style="{StaticResource lblTitle}"/>

<StackPanel Grid.Row="1" Orientation="Horizontal" VerticalAlignment="Top">

<Label Content="Выберите склад" Width="142" Height="35" Margin="10" Style="{StaticResource lblText}"/>

<ComboBox Name="cmbWarehouse"

Width="150"

Height="25"

DisplayMemberPath="Name" SelectedValue="{Binding idWarehouse}" SelectedValuePath="idWarehouse"/>

<Label Content="Ведите количество на распределение" Width="325" Height="35" Margin="10" Style="{StaticResource lblText}"/>

<TextBox Name="txtQuantity" Width="100" Height="25"/>

</StackPanel>

<StackPanel Grid.Row="1" Orientation="Horizontal" VerticalAlignment="Bottom">

<Label Content="Название материала:" Style="{StaticResource lblText1}"/>

<TextBlock Name="txtMaterials" Width="200" Height="25" FontSize="18" Style="{StaticResource ddlStyle}" />

<Label Content="Номер заявки:" Style="{StaticResource lblText1}"/>

<TextBlock Name="txtApplication" Width="200" Height="25" FontSize="18" Style="{StaticResource ddlStyle}" />

</StackPanel>

<DataGrid Name="dgMovement" Grid.Row="2" AutoGenerateColumns="False" IsEnabled="True" CanUserAddRows="False" IsReadOnly="True" MouseDoubleClick="dgMovement\_MouseDoubleClick">

<DataGrid.ContextMenu>

<ContextMenu>

<MenuItem Header="Обновить" Click="MenuUpdate\_Click">

<MenuItem.Icon>

<Image Source="/Photo/iconupdate.png"/>

</MenuItem.Icon>

</MenuItem>

<MenuItem Header="Распределить" Click="MenuDistribute\_Click">

<MenuItem.Icon>

<Image Source="/Photo/icondistribution.png"/>

</MenuItem.Icon>

</MenuItem>

<MenuItem Header="Проверить распределение" Click="MenuСheck\_Click">

<MenuItem.Icon>

<Image Source="/Photo/iconcheck.png"/>

</MenuItem.Icon>

</MenuItem>

</ContextMenu>

</DataGrid.ContextMenu>

<DataGrid.Columns>

<DataGridTextColumn Header="Название материала" Width="\*" Binding="{Binding Material.Name}"/>

<DataGridTextColumn Header="Чертежный номер" Width="\*" Binding="{Binding Material.DrawingNumber}"/>

<DataGridTextColumn Header="Количество для распределения" Width="\*" Binding ="{Binding DifferenceQuantity}"/>

<DataGridTextColumn Header="Номер заявки" Width="\*" Binding="{Binding IdExpenseIvoices}"/>

</DataGrid.Columns>

</DataGrid>

</Grid>

***DistributionList.xaml.cs:***

public partial class DistributionList : Page

{

public ExpenseComposition \_currentItem = new ExpenseComposition();

public DistributionList()

{

InitializeComponent();

dgMovement.ItemsSource = ClassFrame.db.ExpenseComposition.ToList();

cmbWarehouse.ItemsSource = ClassFrame.db.Database.SqlQuery<Warehouses>("select \* from Nomenclature.dbo.Warehouses").ToList();

}

private void dgMovement\_MouseDoubleClick(object sender, MouseButtonEventArgs e)

{

\_currentItem = dgMovement.SelectedItem as ExpenseComposition;

if (\_currentItem != null)

{

foreach (Warehouses w in cmbWarehouse.ItemsSource)

{

int i = ClassFrame.db.Database.SqlQuery<int?>("select Quantity from Nomenclature.dbo.Storage where idWarehouse = @idW and idMaterial = @idM", new SqlParameter("@idW", w.idWarehouse), new SqlParameter("@idM", \_currentItem.IdMaterial)).FirstOrDefault() == null

? 0 : ClassFrame.db.Database.SqlQuery<int>("select Quantity from Nomenclature.dbo.Storage where idWarehouse = @idW and idMaterial = @idM", new SqlParameter("@idW", w.idWarehouse), new SqlParameter("@idM", \_currentItem.IdMaterial)).FirstOrDefault();

if (i > 0)

{

txtMaterials.Text = \_currentItem.Material.Name;

txtApplication.Text = Convert.ToString(\_currentItem.IdExpenseIvoices);

cmbWarehouse.SelectedItem = w;

MessageBox.Show($"Количество материала на складе: {i}", "Уведомление", MessageBoxButton.OK, MessageBoxImage.Information);

return;

}

}

MessageBox.Show("Такого материала нет на складах.", "Уведомление", MessageBoxButton.OK, MessageBoxImage.Information);

cmbWarehouse.SelectedIndex = -1;

}

}

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

{

ClassFrame.frmObj.Navigate(new PageMenu());

}

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

{

if (string.IsNullOrEmpty(txtQuantity.Text))

{

MessageBox.Show("Поле количество на раcпределение пустое", "Ошибка ввода", MessageBoxButton.OK, MessageBoxImage.Error);

return;

}

if (Convert.ToInt32(txtQuantity.Text) > \_currentItem.DifferenceQuantity)

{

MessageBox.Show("Количество на раcпределение не может быть больше количества для распределения", "Ошибка ввода", MessageBoxButton.OK, MessageBoxImage.Error);

return;

}

Warehouses w = cmbWarehouse.SelectedItem as Warehouses;

try

{

ClassFrame.db.Database.ExecuteSqlCommand("insert into Nomenclature.dbo.Movement values (@warehouse,@composition,@user,@quantity,1)", new SqlParameter("@warehouse", w.idWarehouse), new SqlParameter("@composition", \_currentItem.IdExpenseComposition), new SqlParameter("@user", Users.GetUsers.ID), new SqlParameter("@quantity", Convert.ToInt32(txtQuantity.Text)));

ClassFrame.db.Database.ExecuteSqlCommand("update Nomenclature.dbo.Storage set Quantity = Storage.Quantity - @quantity where idMaterial = @material and idWarehouse = @warehouse", new SqlParameter("@quantity", Convert.ToInt32(txtQuantity.Text)), new SqlParameter("@material", \_currentItem.IdMaterial), new SqlParameter("@warehouse", w.idWarehouse));

ClassFrame.db.Database.ExecuteSqlCommand("delete Nomenclature.dbo.Storage where idMaterial = @material and Quantity = 0 and idWarehouse = @warehouse", new SqlParameter("@material", \_currentItem.IdMaterial), new SqlParameter("@warehouse", w.idWarehouse));

dgMovement.ItemsSource = ClassFrame.db.Database.ExecuteSqlCommand("d");

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

if (\_currentItem.DifferenceQuantity != 0)

{

cmbWarehouse.SelectedIndex++;

w = cmbWarehouse.SelectedItem as Warehouses;

int i = ClassFrame.db.Database.SqlQuery<int?>("select Quantity from Nomenclature.dbo.Storage where idWarehouse = @idW and idMaterial = @idM", new SqlParameter("@idW", w.idWarehouse), new SqlParameter("@idM", \_currentItem.IdMaterial)).FirstOrDefault() == null

? 0 : ClassFrame.db.Database.SqlQuery<int>("select Quantity from Nomenclature.dbo.Storage where idWarehouse = @idW and idMaterial = @idM", new SqlParameter("@idW", w.idWarehouse), new SqlParameter("@idM", \_currentItem.IdMaterial)).FirstOrDefault();

MessageBox.Show($"Количество материала на складе: {i}", "Уведомление", MessageBoxButton.OK, MessageBoxImage.Information);

}

else

{

MessageBox.Show("Успешно распределенно");

\_currentItem = null;

}

dgMovement.ItemsSource = ClassFrame.db.ExpenseComposition.ToList();

txtQuantity.Text = "";

}

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

{

if (\_currentItem == null)

{

MessageBox.Show($"Позиция не выбрана.", "Уведомление", MessageBoxButton.OK);

return;

}

List<ExpenseComposition> list = ClassFrame.db.ExpenseComposition.Where(x => x.IdExpenseIvoices == \_currentItem.IdExpenseIvoices).ToList();

foreach (ExpenseComposition expense in list)

{

int? d = ClassFrame.db.Database.SqlQuery<int?>($"select sum(PartOfQuantity) from Nomenclature.dbo.Movement where idComposition = {expense.IdExpenseComposition} and ArrivalOrExpenditure = 1").Single();

if (d == null) d = 0;

if (d != expense.Quantity)

{

MessageBox.Show($"Товар \"{expense.Material.Name}\" не закрыт.", "Уведомление", MessageBoxButton.OK);

return;

}

}

try

{

ClassFrame.db.ExpenseIvoices.FirstOrDefault(x => x.IdExpenseIvoices == \_currentItem.IdExpenseIvoices).DistributedInvoice = true;

ClassFrame.db.SaveChanges();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

MessageBox.Show("Заказ закрыт", "Уведомление", MessageBoxButton.OK, MessageBoxImage.Information);

ClassFrame.frmObj.Navigate(new PageOrder());

}

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

{

dgMovement.ItemsSource = ClassFrame.db.ExpenseComposition.ToList();

cmbWarehouse.ItemsSource = ClassFrame.db.Database.SqlQuery<Warehouses>("select \* from Nomenclature.dbo.Warehouses").ToList();

}

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

{

ClassFrame.frmObj.Navigate(new PageMenu());

}

private void Back1\_Click(object sender, RoutedEventArgs e)

{

ClassFrame.frmObj.Navigate(new PageMenu());

}

}

***DistributionWarehouse.xaml:***

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="1.2\*"/>

<RowDefinition Height="3\*"/>

<RowDefinition Height="5\*"/>

<RowDefinition Height="1\*"/>

</Grid.RowDefinitions>

<StackPanel Grid.Row="0" VerticalAlignment="Center" HorizontalAlignment="Right">

<Button x:Name="Back" Click="Back\_Click" Width="30" Height="30" BorderThickness="0">

<Button.Background>

<ImageBrush ImageSource="/Photo/iconhome.png"/>

</Button.Background>

</Button>

</StackPanel>

<StackPanel Grid.Row="0" VerticalAlignment="Center" HorizontalAlignment="Left">

<Button x:Name="Back1" Click="Back1\_Click" Width="30" Height="30" BorderThickness="0" >

<Button.Background>

<ImageBrush ImageSource="/Photo/iconback.png"/>

</Button.Background>

</Button>

</StackPanel>

<Label Content="Отдача товара" Style="{StaticResource lblTitle}"/>

<StackPanel Grid.Row="1" Orientation="Horizontal" VerticalAlignment="Top">

<Label Content="Выберите материал" Width="175" Height="35" Margin="10" Style="{StaticResource lblText}"/>

<ComboBox Name="cmbMaterial" Width="170" Height="20" SelectedIndex="0" DisplayMemberPath="Material.Name" SelectionChanged="cmbMaterial\_SelectionChanged"/>

<Label Content="Количество на распределение:" Style="{StaticResource lblText}" Margin="10"/>

<TextBlock Name="txtCount" Width="85" Height="25" FontSize="18" Style="{StaticResource ddlStyle}" />

</StackPanel>

<StackPanel Grid.Row="1" HorizontalAlignment="Left" VerticalAlignment="Bottom" >

<Label Content="Поиск:" Foreground="DarkSlateBlue" FontSize="18"/>

<TextBox Name="txtSearch" Width="125" Height="20" TextChanged="txtSearch\_TextChanged" Margin="10"/>

</StackPanel>

<StackPanel Grid.Row="1" HorizontalAlignment="Center" VerticalAlignment="Bottom" >

<TextBlock Name="txtPosition" Width="200" Height="25" FontSize="18" Style="{StaticResource ddlStyle}" Margin="10"/>

</StackPanel>

<DataGrid Name="dgWarhouse" Grid.Row="2" AutoGenerateColumns="False" IsEnabled="True" CanUserAddRows="False" CellEditEnding="DgWarhouse\_CellEditEnding" >

<DataGrid.ContextMenu>

<ContextMenu>

<MenuItem Header="Обновить" Click="MenuUpdate\_Click">

<MenuItem.Icon>

<Image Source="/Photo/iconupdate.png"/>

</MenuItem.Icon>

</MenuItem>

<MenuItem Header="Распределить" Click="MenuDistribute\_Click">

<MenuItem.Icon>

<Image Source="/Photo/icondistribution.png"/>

</MenuItem.Icon>

</MenuItem>

<MenuItem Header="Проверить распределение" Click="MenuСheck\_Click">

<MenuItem.Icon>

<Image Source="/Photo/iconcheck.png"/>

</MenuItem.Icon>

</MenuItem>

</ContextMenu>

</DataGrid.ContextMenu>

<DataGrid.Columns >

<DataGridTextColumn Header="Название склада" Width="\*" Binding="{Binding Warehouse.Name}" IsReadOnly="True"/>

<DataGridTextColumn Header="Название матеиала" Width="\*" Binding="{Binding Material.Name}" IsReadOnly="True"/>

<DataGridTextColumn Header="Количество" Width="\*" Binding="{Binding Quantity}" IsReadOnly="True"/>

<DataGridTextColumn x:Name="quantityComposition" Width="\*" Header="Отдаваемое количество" Binding="{Binding QuantityComposition, UpdateSourceTrigger=PropertyChanged}"/>

</DataGrid.Columns>

</DataGrid>

</Grid>

***DistributionWarehouse.xaml.cs:***

public partial class DistributionWarehouse : Page

{

ExpenseIvoices expense = new ExpenseIvoices();

public DistributionWarehouse(ExpenseIvoices selected)

{

InitializeComponent();

if(selected != null) expense = selected;

cmbMaterial.ItemsSource = ClassFrame.db.ExpenseComposition.Where(x => x.IdExpenseIvoices == expense.IdExpenseIvoices).ToList();

}

private void cmbMaterial\_SelectionChanged(object sender, SelectionChangedEventArgs e)

{

if (cmbMaterial.SelectedItem is ExpenseComposition c)

{

int? q = ClassFrame.db.Database.SqlQuery<int?>("select sum(PartofQuantity) from Nomenclature.dbo.Movement M join SalesOfMaterial.dbo.ExpenseComposition E on E.IdExpenseComposition = M.idComposition where IdMaterial = @material and ArrivalOrExpenditure = 1 and IdExpenseIvoices = @ivoce", new SqlParameter("@material", c.Material.idMaterial), new SqlParameter("@ivoce", c.IdExpenseIvoices)).Single();

List<Storage> list = ClassFrame.db.Database.SqlQuery<Storage>("select \* from Nomenclature.dbo.Storage where idMaterial = @id", new SqlParameter("@id", c.IdMaterial)).ToList();

foreach (Storage s in list) s.QuantityComposition = ClassFrame.db.Database.SqlQuery<int?>("select sum(PartOfQuantity) from Nomenclature.dbo.Movement M join SalesOfMaterial.dbo.ExpenseComposition C on C.IdExpenseComposition = M.idComposition where M.idComposition = @idC and idMaterial = @idM and idWarehouse = @idW and ArrivalOrExpenditure = 1", new SqlParameter("@idC", c.IdExpenseComposition), new SqlParameter("@idM", c.IdMaterial), new SqlParameter("idW", s.idWarehouse)).Single() == null ? 0 : ClassFrame.db.Database.SqlQuery<int>("select sum(PartOfQuantity) from Nomenclature.dbo.Movement M join SalesOfMaterial.dbo.ExpenseComposition C on C.IdExpenseComposition = M.idComposition where M.idComposition = @idC and idMaterial = @idM and idWarehouse = @idW and ArrivalOrExpenditure = 1", new SqlParameter("@idC", c.IdExpenseComposition), new SqlParameter("@idM", c.IdMaterial), new SqlParameter("@idW", s.idWarehouse)).Single();

dgWarhouse.ItemsSource = list;

txtCount.DataContext = ClassFrame.db.ExpenseComposition.FirstOrDefault(x => x.IdExpenseIvoices == expense.IdExpenseIvoices && x.IdMaterial == c.IdMaterial);

if (q != null) txtCount.Text = ((cmbMaterial.SelectedItem as ExpenseComposition).Quantity - q).ToString();

else txtCount.Text = (cmbMaterial.SelectedItem as ExpenseComposition).Quantity.ToString();

}

if (Convert.ToInt32(txtCount.Text) == 0)

{

txtPosition.Text = "Позиция закрыта";

}

else

txtPosition.Text = "";

quantityComposition.IsReadOnly = false;

}

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

{

ClassFrame.frmObj.Navigate(new PageOrder());

}

private void txtSearch\_TextChanged(object sender, TextChangedEventArgs e)

{

ExpenseComposition c = cmbMaterial.SelectedItem as ExpenseComposition;

if (txtSearch.Text.Count() != 0)

dgWarhouse.ItemsSource = ClassFrame.db.Database.SqlQuery<Storage>("select \* from Nomenclature.dbo.Storage").Where(x => x.idMaterial == c.IdMaterial && x.Warehouse.Name.ToLower().Contains(txtSearch.Text.ToLower())).ToList();

else dgWarhouse.ItemsSource = ClassFrame.db.Database.SqlQuery<Storage>("select \* from Nomenclature.dbo.Storage").Where(x => x.idMaterial == c.IdMaterial).ToList();

}

private void DgWarhouse\_CellEditEnding(object sender, DataGridCellEditEndingEventArgs e)

{

int quantity = ((List<Storage>)dgWarhouse.ItemsSource).Sum(x => x.QuantityComposition);

txtCount.Text = ((cmbMaterial.SelectedItem as ExpenseComposition).Quantity - quantity).ToString();

if (Convert.ToInt32(txtCount.Text) == 0)

{

txtPosition.Text = "Позиция закрыта";

dgWarhouse.ItemsSource = (dgWarhouse.ItemsSource as List<Storage>).Where(x => x.QuantityComposition != 0).ToList();

quantityComposition.IsReadOnly = true;

}

}

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

{

List<Storage> list = dgWarhouse.ItemsSource as List<Storage>;

foreach (Storage s in list)

{

if (s.QuantityComposition > s.Quantity)

{

MessageBox.Show("Количество забираемого товара больше чем количество хранящиеся на складе", "Ошибка ввода", MessageBoxButton.OK, MessageBoxImage.Error);

return;

}

}

if (Convert.ToInt32(txtCount.Text) < 0)

{

MessageBox.Show("Количество забираемого товара больше чем нужно", "Ошибка ввода", MessageBoxButton.OK, MessageBoxImage.Error);

return;

}

if (Convert.ToInt32(txtCount.Text) > 0)

{

MessageBox.Show("Количество забираемого товара меньше чем нужно", "Ошибка ввода", MessageBoxButton.OK, MessageBoxImage.Error);

return;

}

foreach (Storage s in list)

{

try

{

ClassFrame.db.Database.ExecuteSqlCommand("insert into Nomenclature.dbo.Movement values (@warehouse,@composition,@user,@quantity,1)", new SqlParameter("@warehouse", s.idWarehouse), new SqlParameter("@composition", ((ExpenseComposition)cmbMaterial.SelectedItem).IdExpenseComposition), new SqlParameter("@user", Users.GetUsers.ID), new SqlParameter("@quantity", s.QuantityComposition));

ClassFrame.db.Database.ExecuteSqlCommand("update Storage set Quantity = Storage.Quantity - @quantity from Nomenclature.dbo.Storage where idMaterial = @material and idWarehouse = @warhouse", new SqlParameter("@quantity", s.QuantityComposition), new SqlParameter("@material", s.idMaterial), new SqlParameter("@warhouse", s.idWarehouse));

ClassFrame.db.Database.ExecuteSqlCommand("delete Nomenclature.dbo.Storage where idMaterial = @material and Quantity = 0 and idWarehouse = @warhouse", new SqlParameter("@material", s.idMaterial), new SqlParameter("@warhouse", s.idWarehouse));

if (Convert.ToInt32(txtCount.Text) == 0)

txtPosition.Text = "Позиция закрыта";

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

MessageBox.Show("Товар Распределён", "Уведомление", MessageBoxButton.OK);

cmbMaterial.SelectedIndex = cmbMaterial.SelectedIndex + 1;

}

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

{

foreach (ExpenseComposition expense in cmbMaterial.ItemsSource)

{

int? d = ClassFrame.db.Database.SqlQuery<int?>($"select sum(PartOfQuantity) from Nomenclature.dbo.Movement where idComposition = {expense.IdExpenseComposition} and ArrivalOrExpenditure = 1").Single();

if (d == null) d = 0;

if (d != expense.Quantity)

{

MessageBox.Show($"Товар \"{expense.Material.Name}\" не закрыт.", "Уведомление", MessageBoxButton.OK);

return;

}

}

try

{

expense.DistributedInvoice = true;

ClassFrame.db.SaveChanges();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

MessageBox.Show("Заказ закрыт", "Уведомление", MessageBoxButton.OK, MessageBoxImage.Information);

ClassFrame.frmObj.Navigate(new PageOrder());

}

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

{

cmbMaterial.ItemsSource = ClassFrame.db.ExpenseComposition.Where(x => x.IdExpenseIvoices == expense.IdExpenseIvoices).ToList();

}

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

{

ClassFrame.frmObj.Navigate(new PageMenu());

}

private void Back1\_Click(object sender, RoutedEventArgs e)

{

ClassFrame.frmObj.Navigate(new PageOrder());

}

}

***Movment.xaml:***

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="1\*"/>

<RowDefinition Height="2\*"/>

<RowDefinition Height="5\*"/>

<RowDefinition Height="1\*"/>

</Grid.RowDefinitions>

<StackPanel Grid.Row="0" VerticalAlignment="Center" HorizontalAlignment="Left">

<Button x:Name="Back1" Click="Back1\_Click" Width="30" Height="30" BorderThickness="0">

<Button.Background>

<ImageBrush ImageSource="/Photo/iconback.png"/>

</Button.Background>

</Button>

</StackPanel>

<Label Content="Список отданных товаров" Style="{StaticResource lblTitle}"/>

<StackPanel Grid.Row="1" HorizontalAlignment="Left" VerticalAlignment="Bottom" >

<Label Content="Поиск:" Foreground="DarkSlateBlue" FontSize="18"/>

<TextBox Name="txtSearch" Width="125" Height="20" TextChanged="txtSearch\_TextChanged" Margin="10"/>

</StackPanel>

<DataGrid Name="dgMovement" Grid.Row="2" AutoGenerateColumns="False" IsEnabled="True" CanUserAddRows="False" IsReadOnly="True">

<DataGrid.Columns>

<DataGridTextColumn Header="Название склада" Width="\*" Binding="{Binding Warehouse.Name}"/>

<DataGridTextColumn Header="Номер заказа" Width="\*" Binding="{Binding Expense.IdExpenseIvoices}"/>

<DataGridTextColumn Header="ФИО пользователя" Width="\*" Binding ="{Binding User.FIO}"/>

<DataGridTextColumn Header="Количество" Width="\*" Binding="{Binding PartOfQuantity}"/>

</DataGrid.Columns>

</DataGrid>

</Grid>

***Movment.xaml.cs:***

public partial class Movment : Page

{

public Movment()

{

InitializeComponent();

dgMovement.ItemsSource = ClassFrame.db.Database.SqlQuery<Movement>("select \* from Nomenclature.dbo.Movement").ToList();

}

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

{

ClassFrame.frmObj.Navigate(new PageMenu());

}

private void txtSearch\_TextChanged(object sender, TextChangedEventArgs e)

{

if (txtSearch.Text.Count() != 0)

dgMovement.ItemsSource = ClassFrame.db.Database.SqlQuery<Movement>("select \* from Nomenclature.dbo.Movement").Where(x => x.Warehouse.Name.ToLower().Contains(txtSearch.Text.ToLower())).ToList();

else dgMovement.ItemsSource = ClassFrame.db.Database.SqlQuery<Movement>("select \* from Nomenclature.dbo.Movement").ToList();

}

private void Back1\_Click(object sender, RoutedEventArgs e)

{

ClassFrame.frmObj.Navigate(new PageMenu());

}

}

***PageAddExpenseComposition.xaml:***

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="1\*"/>

<RowDefinition Height="2\*"/>

<RowDefinition Height="3\*"/>

</Grid.RowDefinitions>

<Label Grid.Row="0" Content="Добавление заказа" Style="{StaticResource lblTitle}"/>

<StackPanel Grid.Row="1" Orientation="Vertical" HorizontalAlignment="Left" Margin="100,0,0,0">

<Label Content="Выберите материал" Width="125" Height="25"/>

<ComboBox Name="cmbMaterial"

Width="150"

Height="20"

DisplayMemberPath="Name" SelectedValue="{Binding IdMaterial}" SelectedValuePath="idMaterial"/>

<Label Content="Введите количество" Width="150" Height="25" Margin="10,0,0,0"/>

<TextBox Name="txtQuantity" Width="150" Height="20" Text="{Binding Quantity}"/>

<Label Content="Введите цену" Width="125" Height="25" Margin="10,0,0,0"/>

<TextBox Name="txtPrice" Width="150" Height="20" Text="{Binding Price}"/>

</StackPanel>

<StackPanel Grid.Row="1" Orientation="Vertical" HorizontalAlignment="Center" Margin="0,40,0,0">

<Button x:Name="btnAdd" Height="25" Width="160" Content="Добавить" Click="btnAdd\_Click" Margin="10"/>

<Button Width="160" Height="25" Content="Вернуться к списку заказов" Click="Button\_Click"/>

</StackPanel>

<StackPanel Grid.Row="1" Orientation="Vertical" HorizontalAlignment="Right" Margin="0,45,100,0">

<Label Content="Поиск:" Foreground="DarkSlateBlue" FontSize="18"/>

<TextBox Name="txtSearch" Width="150" Height="20" TextChanged="txtSearch\_TextChanged"/>

</StackPanel>

<DataGrid Name="dgExpenseComposition" Grid.Row="2" AutoGenerateColumns="False" IsReadOnly="True" MouseDoubleClick="DgExpenseComposition\_MouseDoubleClick">

<DataGrid.Columns>

<DataGridTextColumn Header="Название материала" Width="\*" Binding="{Binding Material.Name}"/>

<DataGridTextColumn Header="Номер чертежа" Width="\*" Binding="{Binding Material.DrawingNumber}"/>

<DataGridTextColumn Header="Количество" Width="\*" Binding="{Binding Quantity}"/>

<DataGridTextColumn Header="Цена" Width="\*" Binding="{Binding Price}"/>

</DataGrid.Columns>

</DataGrid>

</Grid>

***PageAddExpenseComposition.xaml.cs:***

public partial class PageAddExpenseComposition : Page

{

public int idExpenseIvoices;

public ExpenseComposition \_currentItem = new ExpenseComposition();

public PageAddExpenseComposition(int id)

{

InitializeComponent();

idExpenseIvoices = id;

dgExpenseComposition.ItemsSource = ClassFrame.db.ExpenseComposition.Where(x => x.IdExpenseIvoices == idExpenseIvoices).ToList();

DataContext = \_currentItem;

cmbMaterial.ItemsSource = ClassFrame.db.Database.SqlQuery<Materials>("select \* from Nomenclature.dbo.Materials").ToList();

}

public void LoadDataGridSearch(List<Materials> materials)

{

List<ExpenseComposition> list = new List<ExpenseComposition>();

foreach (ExpenseComposition e in ClassFrame.db.ExpenseComposition.Where(x => x.IdExpenseIvoices == \_currentItem.IdExpenseIvoices).ToList())

if ((from m in materials select m.idMaterial).Contains(e.Material.idMaterial)) list.Add(e);

dgExpenseComposition.ItemsSource = list;

}

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

{

if (cmbMaterial.Text == "" || txtQuantity.Text == "" || txtPrice.Text == "")

{

MessageBox.Show("Введите данные в поля", "Ошибка ввода", MessageBoxButton.OK, MessageBoxImage.Error);

return;

}

Nullable<int> i = ClassFrame.db.Database.SqlQuery<Nullable<int>>("select SUM(Quantity) from Nomenclature.dbo.Storage where idMaterial = @material", new SqlParameter("@material", ((Materials)cmbMaterial.SelectedItem).idMaterial)).FirstOrDefault();

if (i != null && int.Parse(txtQuantity.Text) > i)

{

MessageBox.Show($"Количество не может превышать количество хранимое на складе (На складе: {i})", "Уведомление", MessageBoxButton.OK, MessageBoxImage.Warning);

return;

}

if (i == null)

{

MessageBox.Show($"На складе такого материала нет!", "Уведомление", MessageBoxButton.OK, MessageBoxImage.Warning);

return;

}

ExpenseComposition exp = new ExpenseComposition() { IdExpenseComposition = \_currentItem.IdExpenseComposition, IdMaterial = (int)\_currentItem.IdMaterial, IdExpenseIvoices = idExpenseIvoices, Quantity = \_currentItem.Quantity, Price = \_currentItem.Price };

\_currentItem = null;

\_currentItem = new ExpenseComposition();

DataContext = \_currentItem;

if (exp.IdExpenseComposition == 0) ClassFrame.db.ExpenseComposition.Add(exp);

try

{

ClassFrame.db.SaveChanges();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

dgExpenseComposition.ItemsSource = ClassFrame.db.ExpenseComposition.Where(x => x.IdExpenseIvoices == idExpenseIvoices).ToList();

}

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

{

ClassFrame.frmObj.Navigate(new PageOrder());

}

private void txtSearch\_TextChanged(object sender, TextChangedEventArgs e)

{

if (txtSearch.Text.Count() != 0)

LoadDataGridSearch(ClassFrame.db.Database.SqlQuery<Materials>($"select \* from Nomenclature.dbo.Materials where Name like '%{txtSearch.Text.ToLower()}%' or DrawingNumber like '%{txtSearch.Text.ToLower()}%'").ToList());

else dgExpenseComposition.ItemsSource = ClassFrame.db.ExpenseComposition.Where(x => x.IdExpenseIvoices == idExpenseIvoices).ToList();

}

private void DgExpenseComposition\_MouseDoubleClick(object sender, MouseButtonEventArgs e)

{

\_currentItem = (ExpenseComposition)dgExpenseComposition.SelectedItem;

DataContext = \_currentItem;

}

}

***PageAddMaterials.xaml:***

<Grid>

<StackPanel VerticalAlignment="Center" HorizontalAlignment="Center">

<Label Content="Название товара" Style="{StaticResource lblText2}"/>

<TextBox Name="txtNameMaterial" Width="300" Height="25" Text="{Binding Name}"/>

<Label Content="Тип товара" Style="{StaticResource lblText2}"/>

<ComboBox Name="cmbTypes"

Height="25"

DisplayMemberPath="Name"

SelectedValue="{Binding idType}"

SelectedValuePath="idType"/>

<Label Content="Номер чертежа" Style="{StaticResource lblText2}"/>

<TextBox Name="txtDrawingNumber" Width="300" Height="25" FontSize="18" Text="{Binding DrawingNumber}"/>

</StackPanel>

<StackPanel Style="{StaticResource stc}">

<Button Name="btnAdd" Content="Сохранить" Click="btnAdd\_Click" Style="{StaticResource btn1}"/>

<Button Name="btnBack" Content="Вернуться назад" Click="btnBack\_Click" Margin="10" Style="{StaticResource btn1}"/>

</StackPanel>

</Grid>

***PageAddMaterials.xaml.cs:***

public partial class PageAddMaterials : Page

{

public Materials \_currentItem = new Materials();

public PageAddMaterials(Materials selectedItem)

{

InitializeComponent();

if (selectedItem != null)

{

\_currentItem = selectedItem;

}

DataContext = \_currentItem;

cmbTypes.ItemsSource = ClassFrame.db.Database.SqlQuery<Types>("select \* from Nomenclature.dbo.Types").ToList();

}

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

{

if(txtNameMaterial.Text == "" || cmbTypes.Text == "")

{

MessageBox.Show("Введите данные в поле", "Ошибка ввода", MessageBoxButton.OK, MessageBoxImage.Error);

return;

}

\_currentItem.Name = txtNameMaterial.Text;

\_currentItem.idType = ((Types)cmbTypes.SelectedItem).idType;

//if (\_currentItem.idType == 4) \_currentItem.DrawingNumber = "Отсутствует";

//else \_currentItem.DrawingNumber = txtDrawingNumber.Text;

try

{

if (\_currentItem.idMaterial == 0)

{

if (\_currentItem.DrawingNumber == "Отсутствует") ClassFrame.db.Database.ExecuteSqlCommand("insert into Nomenclature.dbo.Materials values (@name,NULL,@type)", new SqlParameter("@name", \_currentItem.Name), new SqlParameter("@type", \_currentItem.idType));

else ClassFrame.db.Database.ExecuteSqlCommand("insert into Nomenclature.dbo.Materials values (@name,@plan,@type)", new SqlParameter("@name", \_currentItem.Name), new SqlParameter("@plan", \_currentItem.DrawingNumber), new SqlParameter("@type", \_currentItem.idType));

}

else

{

if (\_currentItem.DrawingNumber == "Отсутствует") ClassFrame.db.Database.ExecuteSqlCommand("update Nomenclature.dbo.Materials set Name = @name, DrawingNumber = NULL, idType = @type where idMaterial = @id", new SqlParameter("@name", \_currentItem.Name), new SqlParameter("@type", \_currentItem.idType), new SqlParameter("@id", \_currentItem.idMaterial));

else ClassFrame.db.Database.ExecuteSqlCommand("update Nomenclature.dbo.Materials set Name = @name, DrawingNumber = @plan, idType = @type where idMaterial = @id", new SqlParameter("@name", \_currentItem.Name), new SqlParameter("@plan", \_currentItem.DrawingNumber), new SqlParameter("@type", \_currentItem.idType), new SqlParameter("@id", \_currentItem.idMaterial));

}

ClassFrame.frmObj.Navigate(new PageMaaterials());

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

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

{

ClassFrame.frmObj.Navigate(new PageMaaterials());

}

}

***PageAddOrder.xaml:***

<Grid>

<StackPanel VerticalAlignment="Center" HorizontalAlignment="Center">

<Label Content="Выберите контрагента" Style="{StaticResource lblText2}"/>

<ComboBox Name="cmbСounterparties"

Height="25"

DisplayMemberPath="Nazv\_Сounterparties" SelectedValue="{Binding IdСounterparties}" SelectedValuePath="IdСounterparties"/>

<Label Content="ФИО сотрудника" Style="{StaticResource lblText2}"/>

<TextBlock Name="txtFIO" Width="300" Height="25" FontSize="18" Text="{Binding Employee.FIO}"/>

<DatePicker Name="datePicker1" IsDropDownOpen="False" SelectedDateFormat="Short" SelectedDate="{Binding Date\_Transfer}"/>

</StackPanel>

<StackPanel Style="{StaticResource stc}">

<Button Name="btnAdd" Content="Сохранить" Click="btnAdd\_Click" Style="{StaticResource btn1}"/>

<Button Name="btnBack" Content="Вернуться назад" Click="btnBack\_Click" Margin="10" Style="{StaticResource btn1}"/>

</StackPanel>

</Grid>

PageAddOrder.xaml.cs:

public partial class PageAddOrder : Page

{

public ExpenseIvoices \_currentItem = new ExpenseIvoices();

public PageAddOrder(ExpenseIvoices selectedItem)

{

InitializeComponent();

if (selectedItem != null)

{

\_currentItem = selectedItem;

}

DataContext = \_currentItem;

txtFIO.Text = Users.GetUsers.FIO;

cmbСounterparties.ItemsSource = ClassFrame.db.Сounterparties.ToList();

}

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

{

if (cmbСounterparties.Text == "" && datePicker1.Text == "")

{

MessageBox.Show("Введите данные в поле", "Ошибка ввода", MessageBoxButton.OK, MessageBoxImage.Error);

return;

}

\_currentItem.IdСounterparties = ((Classes.Сounterparties)cmbСounterparties.SelectedItem).IdСounterparties;

\_currentItem.IdEmployee = Users.GetUsers.ID;

\_currentItem.Date\_Transfer = datePicker1.SelectedDate;

\_currentItem.DistributedInvoice = false;

if (\_currentItem.IdExpenseIvoices == 0) ClassFrame.db.ExpenseIvoices.Add(\_currentItem);

try

{

ClassFrame.db.SaveChanges();

ClassFrame.frmObj.Navigate(new PageAddExpenseComposition(\_currentItem.IdExpenseIvoices));

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

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

{

ClassFrame.frmObj.Navigate(new PageOrder());

}

}

***PageAuthorization.xaml:***

<Grid>

<StackPanel HorizontalAlignment="Center">

<Label Content="Авторизация" Style="{StaticResource lblTitle}"/>

</StackPanel>

<StackPanel VerticalAlignment="Center" HorizontalAlignment="Center">

<Label Content="Введите логин:" Style="{StaticResource lblText2}"/>

<TextBox Name="login" Width="180" Height="25" />

<Label Content="Введите пароль:" Style="{StaticResource lblText2}"/>

<PasswordBox Name="password" Width="180" Height="25"/>

<StackPanel Orientation="Horizontal" Margin="0,10,0,0">

<Button Name="btnenter" Content="Войти" Width="55" Height="25" Margin="0,0,5,0" Click="btnenter\_Click">

<Button.Resources>

<Style TargetType="Border">

<Setter Property="CornerRadius" Value="10"></Setter>

</Style>

</Button.Resources>

</Button>

<Button Name="btnregistration" Content="Зарегистрироваться" Width="120" Height="25" Click="btnregistration\_Click">

<Button.Resources>

<Style TargetType="Border">

<Setter Property="CornerRadius" Value="10"></Setter>

</Style>

</Button.Resources>

</Button>

</StackPanel>

</StackPanel>

</Grid>

***PageAuthorization.xaml.cs:***

public partial class PageAuthorization : Page

{

public PageAuthorization()

{

InitializeComponent();

login.Text = "";

password.Password = "";

}

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

{

try

{

if (login.Text == "") MessageBox.Show("Заполните поле логин", "Ошибка ввода данных", MessageBoxButton.OK, MessageBoxImage.Error);

if (password.Password == "") MessageBox.Show("Заполните поле пароль", "Ошибка ввода данных", MessageBoxButton.OK, MessageBoxImage.Error);

else

{

Employee employee = ClassFrame.db.Employee.FirstOrDefault(x => x.Login\_Employee == login.Text && x.Password\_Employee == password.Password);

if (employee == null) MessageBox.Show("Неверный логин или пароль", "Ошибка ввода данных", MessageBoxButton.OK, MessageBoxImage.Error);

else

{

Users \_user = new Users(employee);

ClassFrame.frmObj.Navigate(new PageMenu());

}

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

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

{

ClassFrame.frmObj.Navigate(new PageRegistration());

}

}

***PageExpenseComposition.xaml:***

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="1\*"/>

<RowDefinition Height="1\*"/>

<RowDefinition Height="5\*"/>

<RowDefinition Height="1\*"/>

</Grid.RowDefinitions>

<StackPanel Grid.Row="0" VerticalAlignment="Center" HorizontalAlignment="Right">

<Button x:Name="Back" Click="Back\_Click" Width="30" Height="30" BorderThickness="0">

<Button.Background>

<ImageBrush ImageSource="/Photo/iconhome.png"/>

</Button.Background>

</Button>

</StackPanel>

<StackPanel Grid.Row="0" VerticalAlignment="Center" HorizontalAlignment="Left">

<Button x:Name="Back1" Click="Back1\_Click" Width="30" Height="30" BorderThickness="0">

<Button.Background>

<ImageBrush ImageSource="/Photo/iconback.png"/>

</Button.Background>

</Button>

</StackPanel>

<Label Content="Состав заказа" Style="{StaticResource lblTitle}"/>

<StackPanel Grid.Row="1" Orientation="Horizontal" HorizontalAlignment="Center">

<Label Content="Поиск:" Height="35" Style="{StaticResource lblText1}"/>

<TextBox Name="TxtExpenseComposition" Width="160" Height="25" TextChanged="TxtExpenseComposition\_TextChanged"/>

</StackPanel>

<DataGrid Name="dgExpenseComposition" Grid.Row="2" AutoGenerateColumns="False" IsReadOnly ="True" >

<DataGrid.ContextMenu>

<ContextMenu>

<MenuItem Header="Обновить" Click="MenuUpdate\_Click">

<MenuItem.Icon>

<Image Source="/Photo/iconupdate.png"/>

</MenuItem.Icon>

</MenuItem>

<MenuItem Header="Удалить" Click="MenuDelet\_Click">

<MenuItem.Icon>

<Image Source="/Photo/icondelet.png"/>

</MenuItem.Icon>

</MenuItem>

</ContextMenu>

</DataGrid.ContextMenu>

<DataGrid.Columns>

<DataGridTextColumn Header="Название материала" Width="\*" Binding="{Binding Material.Name}"/>

<DataGridTextColumn Header="Количество" Width="\*" Binding="{Binding Quantity}"/>

<DataGridTextColumn Header="Цена, руб." Width="\*" Binding="{Binding Price}" />

</DataGrid.Columns>

</DataGrid>

</Grid>

***PageExpenseComposition.xaml.cs:***

public partial class PageExpenseComposition : Page

{

public readonly ExpenseIvoices \_ivoices = new ExpenseIvoices();

public PageExpenseComposition(ExpenseIvoices ivoices)

{

InitializeComponent();

\_ivoices = ivoices;

dgExpenseComposition.ItemsSource = ClassFrame.db.ExpenseComposition.Where(x => x.IdExpenseIvoices == \_ivoices.IdExpenseIvoices).ToList();

}

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

{

ClassFrame.frmObj.Navigate(new PageOrder());

}

private void TxtExpenseComposition\_TextChanged(object sender, TextChangedEventArgs e)

{

if (TxtExpenseComposition.Text.Count() != 0)

dgExpenseComposition.ItemsSource = ClassFrame.db.ExpenseComposition.ToList().Where(x => x.IdExpenseIvoices == \_ivoices.IdExpenseIvoices && x.Material.Name.ToLower().Contains(TxtExpenseComposition.Text.ToLower())).ToList();

else dgExpenseComposition.ItemsSource = ClassFrame.db.ExpenseComposition.Where(x => x.IdExpenseIvoices == \_ivoices.IdExpenseIvoices).ToList();

}

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

{

List<ExpenseComposition> ExpenceCompositionRemoving = dgExpenseComposition.SelectedItems.Cast<ExpenseComposition>().ToList();

if (MessageBox.Show("Удалить состав заказа", "Внимание", MessageBoxButton.YesNo, MessageBoxImage.Question) == MessageBoxResult.Yes)

{

try

{

ClassFrame.db.ExpenseComposition.RemoveRange(ExpenceCompositionRemoving);

ClassFrame.db.SaveChanges();

MessageBox.Show("Данные удаленны");

dgExpenseComposition.ItemsSource = ClassFrame.db.ExpenseComposition.Where(x => x.IdExpenseIvoices == \_ivoices.IdExpenseIvoices).ToList();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message.ToString());

}

}

}

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

{

dgExpenseComposition.ItemsSource = ClassFrame.db.ExpenseComposition.Where(x => x.IdExpenseIvoices == \_ivoices.IdExpenseIvoices).ToList();

}

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

{

ClassFrame.frmObj.Navigate(new PageMenu());

}

private void Back1\_Click(object sender, RoutedEventArgs e)

{

ClassFrame.frmObj.Navigate(new PageOrder());

}

}

***PageMaterials.xaml:***

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="1\*"/>

<RowDefinition Height="1\*"/>

<RowDefinition Height="5\*"/>

<RowDefinition Height="1\*"/>

</Grid.RowDefinitions>

<StackPanel Grid.Row="0" VerticalAlignment="Center" HorizontalAlignment="Left">

<Button x:Name="Back1" Click="Back1\_Click" Width="30" Height="30" BorderThickness="0">

<Button.Background>

<ImageBrush ImageSource="/Photo/iconback.png"/>

</Button.Background>

</Button>

</StackPanel>

<Label Content="Материалы" Style="{StaticResource lblTitle}"/>

<StackPanel Grid.Row="1" Orientation="Horizontal" HorizontalAlignment="Center">

<Label Content="Поиск:" Height="35" Style="{StaticResource lblText1}"/>

<TextBox Name="TxtMaterial" Width="160" Height="25" TextChanged="TxtMaterial\_TextChanged"/>

</StackPanel>

<DataGrid Name="dgMaterials" Grid.Row="2" AutoGenerateColumns="False" IsReadOnly ="True">

<DataGrid.ContextMenu>

<ContextMenu>

<MenuItem Header="Обновить" Click="MenuUpdate\_Click">

<MenuItem.Icon>

<Image Source="/Photo/iconupdate.png"/>

</MenuItem.Icon>

</MenuItem>

<MenuItem Header="Добавить" Click="MenuAdd\_Click">

<MenuItem.Icon>

<Image Source="/Photo/iconadd.png"/>

</MenuItem.Icon>

</MenuItem>

<MenuItem Header="Удалить" Click="MenuDelet\_Click">

<MenuItem.Icon>

<Image Source="/Photo/icondelet.png"/>

</MenuItem.Icon>

</MenuItem>

<MenuItem Header="Изменить" Click="MenuItem\_Click">

<MenuItem.Icon>

<Image Source="/Photo/iconchange.png"/>

</MenuItem.Icon>

</MenuItem>

<MenuItem Header="Фильтр">

<MenuItem.Icon>

<Image Source="/Photo/iconfiltr.png"/>

</MenuItem.Icon>

<MenuItem Header="Изделие" Click="MenuFilter\_Click"/>

<MenuItem Header="Узел" Click="MenuFilter\_Click"/>

<MenuItem Header="Деталь" Click="MenuFilter\_Click"/>

<MenuItem Header="Материал" Click="MenuFilter\_Click"/>

<Separator/>

<MenuItem Header="Сброс" Click="MenuClear\_Click"/>

</MenuItem>

</ContextMenu>

</DataGrid.ContextMenu>

<DataGrid.Columns>

<DataGridTextColumn Header="Название товарв" Width="\*" Binding="{Binding Name}"/>

<DataGridTextColumn Header="Номер чертежа" Width="\*" Binding="{Binding DrawingNumber}"/>

<DataGridTextColumn Header="Тип товара" Width="\*" Binding="{Binding Type.Name}"/>

</DataGrid.Columns>

</DataGrid>

</Grid>

***PageMaterials.xaml.cs:***

public partial class PageMaaterials : Page

{

public PageMaaterials()

{

InitializeComponent();

dgMaterials.ItemsSource = ClassFrame.db.Database.SqlQuery<Materials>("select \* from Nomenclature.dbo.Materials").ToList();

}

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

{

ClassFrame.frmObj.Navigate(new PageMenu());

}

private void TxtMaterial\_TextChanged(object sender, TextChangedEventArgs e)

{

if (TxtMaterial.Text.Count() != 0) dgMaterials.ItemsSource = ClassFrame.db.Database.SqlQuery<Materials>("select \* from Nomenclature.dbo.Materials").Where(x => x.Name.ToLower().Contains(TxtMaterial.Text.ToLower()) || x.Type.Name.ToLower().Contains(TxtMaterial.Text.ToLower()) || x.DrawingNumber.ToLower().Contains(TxtMaterial.Text.ToLower())).ToList();

else dgMaterials.ItemsSource = ClassFrame.db.Database.SqlQuery<Materials>("select \* from Nomenclature.dbo.Materials").ToList();

}

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

{

ClassFrame.frmObj.Navigate(new PageAddMaterials((Materials)dgMaterials.SelectedItem));

}

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

{

ClassFrame.frmObj.Navigate(new PageAddMaterials(null));

}

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

{

List<Materials> delet = dgMaterials.SelectedItems.Cast<Materials>().ToList();

if (MessageBox.Show("Удалить данные", "Внимание", MessageBoxButton.YesNo, MessageBoxImage.Question) == MessageBoxResult.Yes)

{

try

{

foreach (Materials s in delet)

{

ClassFrame.db.Database.ExecuteSqlCommand("delete from Nomenclature.dbo.Storage where idMaterial = @id", new SqlParameter("@id", s.idMaterial));

ClassFrame.db.Database.ExecuteSqlCommand("delete from Nomenclature.dbo.Hierarchy where idParent = @id or idChild = @id", new SqlParameter("@id", s.idMaterial));

ClassFrame.db.Database.ExecuteSqlCommand("delete from Nomenclature.dbo.Materials where idMaterial = @id", new SqlParameter("@id", s.idMaterial));

}

MessageBox.Show("Данные удаленны");

dgMaterials.ItemsSource = ClassFrame.db.Database.SqlQuery<Materials>("select \* from Nomenclature.dbo.Materials").ToList();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message.ToString());

}

}

}

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

{

MenuItem m = sender as MenuItem;

switch (m.Header)

{

case "Изделие":

dgMaterials.ItemsSource = ClassFrame.db.Database.SqlQuery<Materials>("select \* from Nomenclature.dbo.Materials").Where(x => x.Type.Name == "Изделие").ToList();

break;

case "Узел":

dgMaterials.ItemsSource = ClassFrame.db.Database.SqlQuery<Materials>("select \* from Nomenclature.dbo.Materials").Where(x => x.Type.Name == "Узел").ToList();

break;

case "Деталь":

dgMaterials.ItemsSource = ClassFrame.db.Database.SqlQuery<Materials>("select \* from Nomenclature.dbo.Materials").Where(x => x.Type.Name == "Деталь").ToList();

break;

case "Материал":

dgMaterials.ItemsSource = ClassFrame.db.Database.SqlQuery<Materials>("select \* from Nomenclature.dbo.Materials").Where(x => x.Type.Name == "Материал").ToList();

break;

}

}

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

{

dgMaterials.ItemsSource = ClassFrame.db.Database.SqlQuery<Materials>("select \* from Nomenclature.dbo.Materials").ToList();

}

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

{

dgMaterials.ItemsSource = ClassFrame.db.Database.SqlQuery<Materials>("select \* from Nomenclature.dbo.Materials").ToList();

}

private void Back1\_Click(object sender, RoutedEventArgs e)

{

ClassFrame.frmObj.Navigate(new PageMenu());

}

}

***PageMenu.xaml:***

public partial class PageMenu : Page

{

public PageMenu()

{

InitializeComponent();

switch (Users.GetUsers.IdTypesEmployee)

{

case 1:

Button btn = new Button() { Name = "btnListOfMaerials", Width = 200, Height = 30, Content = "Товары хранящиеся на складе", Margin = new Thickness(10) };

btn.Click += btnListOfMaerials\_Click;

stplButtons.Children.Add(btn);

Button btn1 = new Button() { Name = "btnDistribution", Width = 200, Height = 30, Content = "Отдаваемые материалы" };

btn1.Click += btnDistribution\_Click;

stplButtons.Children.Add(btn1);

Button btn2 = new Button() { Name = "btnUser", Width = 200, Height = 30, Content = "Пользователи", Margin = new Thickness(10) };

btn2.Click += btnUser\_Click;

stplButtons.Children.Add(btn2);

Button btn3 = new Button() { Name = "btnMovment", Width = 200, Height = 30, Content = "Отданные материалы" };

btn3.Click += btnMovment\_Click;

stplButtons.Children.Add(btn3);

break;

case 3:

Button btn4 = new Button() { Name = "btnListOfMaerials", Width = 200, Height = 30, Content = "Товары хранящиеся на складе", Margin = new Thickness(10) };

btn4.Click += btnListOfMaerials\_Click;

stplButtons.Children.Add(btn4);

Button btn5 = new Button() { Name = "btnDistribution", Width = 200, Height = 30, Content = "Отдаваемые материалы"};

btn5.Click += btnDistribution\_Click;

stplButtons.Children.Add(btn5);

Button btn6= new Button() { Name = "btnMovment", Width = 200, Height = 30, Content = "Отданные материалы", Margin = new Thickness(10) };

btn6.Click += btnMovment\_Click;

stplButtons.Children.Add(btn6);

break;

}

}

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

{

ClassFrame.frmObj.Navigate(new PageOrder());

}

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

{

ClassFrame.frmObj.Navigate(new ListOfMaterials());

}

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

{

ClassFrame.frmObj.Navigate(new PageMaaterials());

}

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

{

ClassFrame.frmObj.Navigate(new PagePersonalAccount());

}

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

{

ClassFrame.frmObj.Navigate(new DistributionList());

}

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

{

ClassFrame.frmObj.Navigate(new Movment());

}

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

{

ClassFrame.frmObj.Navigate(new User());

}

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

{

ClassFrame.frmObj.Navigate(new Сounterparties());

}

}

***PageMenu.xaml.cs:***

public partial class PageMenu : Page

{

public PageMenu()

{

InitializeComponent();

switch (Users.GetUsers.IdTypesEmployee)

{

case 1:

Button btn = new Button() { Name = "btnListOfMaerials", Width = 200, Height = 30, Content = "Товары хранящиеся на складе", Margin = new Thickness(10) };

btn.Click += btnListOfMaerials\_Click;

stplButtons.Children.Add(btn);

Button btn1 = new Button() { Name = "btnDistribution", Width = 200, Height = 30, Content = "Отдаваемые материалы" };

btn1.Click += btnDistribution\_Click;

stplButtons.Children.Add(btn1);

Button btn2 = new Button() { Name = "btnUser", Width = 200, Height = 30, Content = "Пользователи", Margin = new Thickness(10) };

btn2.Click += btnUser\_Click;

stplButtons.Children.Add(btn2);

Button btn3 = new Button() { Name = "btnMovment", Width = 200, Height = 30, Content = "Отданные материалы" };

btn3.Click += btnMovment\_Click;

stplButtons.Children.Add(btn3);

break;

case 3:

Button btn4 = new Button() { Name = "btnListOfMaerials", Width = 200, Height = 30, Content = "Товары хранящиеся на складе", Margin = new Thickness(10) };

btn4.Click += btnListOfMaerials\_Click;

stplButtons.Children.Add(btn4);

Button btn5 = new Button() { Name = "btnDistribution", Width = 200, Height = 30, Content = "Отдаваемые материалы"};

btn5.Click += btnDistribution\_Click;

stplButtons.Children.Add(btn5);

Button btn6= new Button() { Name = "btnMovment", Width = 200, Height = 30, Content = "Отданные материалы", Margin = new Thickness(10) };

btn6.Click += btnMovment\_Click;

stplButtons.Children.Add(btn6);

break;

}

}

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

{

ClassFrame.frmObj.Navigate(new PageOrder());

}

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

{

ClassFrame.frmObj.Navigate(new ListOfMaterials());

}

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

{

ClassFrame.frmObj.Navigate(new PageMaaterials());

}

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

{

ClassFrame.frmObj.Navigate(new PagePersonalAccount());

}

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

{

ClassFrame.frmObj.Navigate(new DistributionList());

}

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

{

ClassFrame.frmObj.Navigate(new Movment());

}

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

{

ClassFrame.frmObj.Navigate(new User());

}

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

{

ClassFrame.frmObj.Navigate(new Сounterparties());

}

}

***PageOrder.xaml:***

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="1\*"/>

<RowDefinition Height="1\*"/>

<RowDefinition Height="5\*"/>

<RowDefinition Height="1\*"/>

</Grid.RowDefinitions>

<StackPanel Grid.Row="0" VerticalAlignment="Center" HorizontalAlignment="Left">

<Button x:Name="Back1" Click="Back1\_Click" Width="30" Height="30" BorderThickness="0">

<Button.Background>

<ImageBrush ImageSource="/Photo/iconback.png"/>

</Button.Background>

</Button>

</StackPanel>

<Label Content="Заказы" Style="{StaticResource lblTitle}"/>

<StackPanel Grid.Row="1" Orientation="Horizontal" HorizontalAlignment="Center">

<Label Content="Поиск:" Height="35" Style="{StaticResource lblText1}"/>

<TextBox Name="TxtOrder" Width="160" Height="25" TextChanged="TxtOrder\_TextChanged"/>

</StackPanel>

<DataGrid Grid.Row="2" Name="dgExpenseIvoices" MouseDoubleClick="dgExpenseIvoices\_MouseDoubleClick" AutoGenerateColumns="False" IsReadOnly ="True">

<DataGrid.ContextMenu>

<ContextMenu>

<MenuItem Header="Обновить" Click="MenuUpdate\_Click">

<MenuItem.Icon>

<Image Source="/Photo/iconupdate.png"/>

</MenuItem.Icon>

</MenuItem>

<MenuItem Header="Изменить" Click="MenuItem\_Click">

<MenuItem.Icon>

<Image Source="/Photo/iconchange.png"/>

</MenuItem.Icon>

</MenuItem>

<MenuItem Header="Удалить" Click="MenuDelet\_Click">

<MenuItem.Icon>

<Image Source="/Photo/icondelet.png"/>

</MenuItem.Icon>

</MenuItem>

<MenuItem Header="Печать" Click="MenuExcel\_Click">

<MenuItem.Icon>

<Image Source="/Photo/iconprint.png"/>

</MenuItem.Icon>

</MenuItem>

</ContextMenu>

</DataGrid.ContextMenu>

<DataGrid.Columns>

<DataGridCheckBoxColumn Header="Распределение" Width="\*" Binding="{Binding DistributedInvoice}" />

<DataGridTextColumn Header="Название контрагента" Width="\*" Binding="{Binding Сounterparties.Nazv\_Сounterparties}"/>

<DataGridTextColumn Header="ФИО сотрудника" Width="\*" Binding="{Binding Employee.FIO}"/>

<DataGridTextColumn Header="Дата отправки" Width="\*" Binding="{Binding Date\_Transfer,StringFormat='{}{0:dd.MM.yyyy}'}"/>

</DataGrid.Columns>

</DataGrid>

</Grid>

***PageOrder.xaml.cs:***

public partial class PageOrder : System.Windows.Controls.Page

{

public PageOrder()

{

InitializeComponent();

dgExpenseIvoices.ItemsSource = ClassFrame.db.ExpenseIvoices.ToList();

switch (Users.GetUsers.IdTypesEmployee)

{

case 1:

System.Windows.Controls.MenuItem item = new System.Windows.Controls.MenuItem() { Header = "Добавить новый заказ", Icon = new Image() { Source = new BitmapImage(new Uri("pack://application:,,,/Photo/iconadd.png")) } };

item.Click += btnAddOrder\_Click;

dgExpenseIvoices.ContextMenu.Items.Insert(2, item);

System.Windows.Controls.MenuItem item1 = new System.Windows.Controls.MenuItem() { Header = "Распределить", Icon = new Image() { Source = new BitmapImage(new Uri("pack://application:,,,/Photo/icondistribution.png")) } };

item1.Click += btnDistribution\_Click;

dgExpenseIvoices.ContextMenu.Items.Insert(2,item1);

break;

case 2:

System.Windows.Controls.MenuItem item2 = new System.Windows.Controls.MenuItem() { Header = "Добавить новый заказ", Icon = new Image() { Source = new BitmapImage(new Uri("pack://application:,,,/Photo/iconadd.png")) } };

item2.Click += btnAddOrder\_Click;

dgExpenseIvoices.ContextMenu.Items.Insert(2, item2);

break;

case 3:

System.Windows.Controls.MenuItem item3 = new System.Windows.Controls.MenuItem() { Header = "Распределить", Icon = new Image() { Source = new BitmapImage(new Uri("pack://application:,,,/Photo/icondistribution.png")) } };

item3.Click += btnDistribution\_Click;

dgExpenseIvoices.ContextMenu.Items.Insert(2, item3);

break;

}

}

private void dgExpenseIvoices\_MouseDoubleClick(object sender, MouseButtonEventArgs e)

{

if (dgExpenseIvoices.SelectedItem != null)

ClassFrame.frmObj.Navigate(new PageExpenseComposition((ExpenseIvoices)dgExpenseIvoices.SelectedItem));

}

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

{

ClassFrame.frmObj.Navigate(new PageAddOrder(null));

}

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

{

ClassFrame.frmObj.Navigate(new PageMenu());

}

private void TxtOrder\_TextChanged(object sender, TextChangedEventArgs e)

{

if (TxtOrder.Text.Count() != 0)

dgExpenseIvoices.ItemsSource = ClassFrame.db.ExpenseIvoices.Where(x => x.Employee.FIO.ToLower().Contains(TxtOrder.Text.ToLower()) || x.Сounterparties.Nazv\_Сounterparties.ToLower().Contains(TxtOrder.Text.ToLower())).ToList();

else dgExpenseIvoices.ItemsSource = ClassFrame.db.ExpenseIvoices.ToList();

}

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

{

ClassFrame.frmObj.Navigate(new PageAddOrder((ExpenseIvoices)dgExpenseIvoices.SelectedItem));

}

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

{

ClassFrame.frmObj.Navigate(new DistributionWarehouse((ExpenseIvoices)dgExpenseIvoices.SelectedItem));

}

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

{

List<ExpenseIvoices> OrderForRemoving = dgExpenseIvoices.SelectedItems.Cast<ExpenseIvoices>().ToList();

if (MessageBox.Show("Удалить заказы", "Внимание", MessageBoxButton.YesNo, MessageBoxImage.Question) == MessageBoxResult.Yes)

{

try

{

foreach (ExpenseIvoices o in OrderForRemoving)

{

List<ExpenseComposition> list = ClassFrame.db.ExpenseComposition.Where(x => x.IdExpenseIvoices == o.IdExpenseIvoices).ToList();

foreach (ExpenseComposition c in list)

ClassFrame.db.Database.ExecuteSqlCommand("delete Nomenclature.dbo.Movement where idComposition = @com and ArrivalOrExpenditure = 1", new SqlParameter("@com", c.IdExpenseComposition));

ClassFrame.db.ExpenseComposition.RemoveRange(ClassFrame.db.ExpenseComposition.Where(x => x.IdExpenseIvoices == o.IdExpenseIvoices));

}

ClassFrame.db.ExpenseIvoices.RemoveRange(OrderForRemoving);

ClassFrame.db.SaveChanges();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message.ToString());

return;

}

MessageBox.Show("Данные удаленны");

dgExpenseIvoices.ItemsSource = ClassFrame.db.ExpenseIvoices.ToList();

}

}

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

{

ClassFrame.frmObj.Navigate(new PageAddOrder(null));

}

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

{

dgExpenseIvoices.ItemsSource = ClassFrame.db.ExpenseIvoices.ToList();

}

private void Back1\_Click(object sender, RoutedEventArgs e)

{

ClassFrame.frmObj.Navigate(new PageMenu());

}

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

{

Excel.Application app = new Excel.Application();

Workbook wb = app.Workbooks.Add();

Worksheet worksheet = app.Worksheets.Item[1];

ExpenseIvoices i = dgExpenseIvoices.SelectedItem as ExpenseIvoices;

worksheet.Cells[1][1] = $"Накладная №{i.IdExpenseIvoices}";

worksheet.Cells[1][3] = $"Менеджер: {i.Employee.FIO}";

worksheet.Cells[1][5] = $"Контрагент: {i.Сounterparties.Nazv\_Сounterparties}";

worksheet.Cells[1][6] = $"ИНН: {i.Сounterparties.Adress\_Сounterparties}";

worksheet.Cells[1][7] = $"Адрес: {i.Сounterparties.INN}";

int indexRows = 9;

worksheet.Cells[1][indexRows] = "Номер";

worksheet.Cells[2][indexRows] = "Материал";

worksheet.Cells[3][indexRows] = "Количество";

worksheet.Cells[4][indexRows] = "Цена";

worksheet.Cells[5][indexRows] = "Сумма";

List<ExpenseComposition> printItems = ClassFrame.db.ExpenseComposition.Where(x => x.IdExpenseIvoices == i.IdExpenseIvoices).ToList();

foreach (ExpenseComposition item in printItems)

{

worksheet.Cells[1][indexRows + 1] = indexRows - 8;

worksheet.Cells[2][indexRows + 1] = item.Material.Name;

worksheet.Cells[3][indexRows + 1] = item.Quantity;

worksheet.Cells[4][indexRows + 1] = item.Price;

worksheet.Cells[5][indexRows + 1] = item.Quantity \* item.Price;

indexRows++;

}

Range range = worksheet.Range[worksheet.Cells[2][indexRows + 1],worksheet.Cells[5][indexRows + 1]];

range.ColumnWidth = 30;

range.HorizontalAlignment = Excel.XlHAlign.xlHAlignLeft;

app.Visible = true;

}

}

***PagePersonalAccount.xaml:***

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="1\*"/>

<RowDefinition Height="5\*"/>

<RowDefinition Height="2\*"/>

</Grid.RowDefinitions>

<StackPanel Grid.Row="0" VerticalAlignment="Center" HorizontalAlignment="Left">

<Button x:Name="Back1" Click="Back1\_Click" Width="30" Height="30" BorderThickness="0">

<Button.Background>

<ImageBrush ImageSource="/Photo/iconback.png"/>

</Button.Background>

</Button>

</StackPanel>

<Label Content="Личный кабинет" Style="{StaticResource lblTitle}"/>

<StackPanel Grid.Row="1" HorizontalAlignment="Center">

<Label Content="ФИО:" Style="{StaticResource lblText}"/>

<TextBlock Name="txtFIO" Text="{Binding FIO}" Width="300" Height="25" FontSize="18" Style="{StaticResource ddlStyle}" MouseDown="TextBlock\_MouseDown"/>

<TextBox Name="txtFIO1" Text="{Binding FIO}" Width="300" Height="25" FontSize="18" Margin="0,-25,0,0" KeyUp="TextBox\_KeyUp" Visibility="Hidden"/>

<Label Content="Логин:" Style="{StaticResource lblText}"/>

<TextBlock Name="txtLogin" Text="{Binding Login\_Employee}" Width="300" Height="25" FontSize="18" MouseDown="TextBlock\_MouseDown"/>

<TextBox Name="txtLogin1" Text="{Binding Login\_Employee}" Width="300" Height="25" FontSize="18" Margin="0,-25,0,0" KeyUp="TextBox\_KeyUp" Visibility="Hidden"/>

<Label Content="Пароль:" Style="{StaticResource lblText}"/>

<TextBlock Name="txtPasword" Text="{Binding Password\_Employee}" Width="300" Height="25" FontSize="18" MouseDown="TextBlock\_MouseDown"/>

<TextBox Name="txtPasword1" Text="{Binding Password\_Employee}" Width="300" Height="25" FontSize="18" Margin="0,-25,0,0" KeyUp="TextBox\_KeyUp" Visibility="Hidden"/>

<Label Content="Тип пользователя:" Style="{StaticResource lblText}"/>

<TextBlock Name="txtType" Text="{Binding Nazv\_type}" Width="300" Height="25" FontSize="18" MouseDown="TextBlock\_MouseDown"/>

<TextBox Name="txtType1" Text="{Binding Nazv\_type}" Width="300" Height="25" FontSize="18" Margin="0,-25,0,0" KeyUp="TextBox\_KeyUp" Visibility="Hidden"/>

</StackPanel>

<StackPanel Grid.Row="2">

<Button Content="Выйти из акаунта" Click="Button\_Click" Style="{StaticResource btn1}"/>

</StackPanel>

</Grid>

***PagePersonalAccount.xaml.cs:***

public partial class PagePersonalAccount : Page

{

Employee \_currentUser = ClassFrame.db.Employee.FirstOrDefault(x => x.IdEmployee == Users.GetUsers.ID);

public PagePersonalAccount()

{

InitializeComponent();

DataContext = Users.GetUsers;

}

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

{

ClassFrame.frmObj.Navigate(new PageMenu());

}

private void TextBox\_KeyUp(object sender, KeyEventArgs e)

{

if (e.Key == Key.Enter)

{

TextBox textBox = e.Source as TextBox;

if (textBox.Name == "txtFIO1")

{

Users.GetUsers.FIO = textBox.Text;

txtFIO1.Visibility = Visibility.Hidden;

\_currentUser.FIO = textBox.Text;

}

if (textBox.Name == "txtLogin1")

{

Users.GetUsers.Login\_Employee = textBox.Text;

txtLogin1.Visibility = Visibility.Hidden;

\_currentUser.Login\_Employee = textBox.Text;

}

if (textBox.Name == "txtPasword1")

{

Users.GetUsers.Password\_Employee = textBox.Text;

txtPasword1.Visibility = Visibility.Hidden;

\_currentUser.Password\_Employee = textBox.Text;

}

ClassFrame.db.SaveChanges();

}

if (e.Key == Key.LeftCtrl)

{

TextBox textBox = e.Source as TextBox;

if (textBox.Name == "txtFIO1")

{

textBox.Text = \_currentUser.FIO;

txtFIO1.Visibility = Visibility.Hidden;

}

if(textBox.Name == "txtLogin1")

{

textBox.Text = \_currentUser.Login\_Employee;

txtLogin1.Visibility = Visibility.Hidden;

}

if (textBox.Name == "txtPasword1")

{

textBox.Text = \_currentUser.Password\_Employee;

txtPasword1.Visibility = Visibility.Hidden;

}

}

}

private void TextBlock\_MouseDown(object sender, MouseButtonEventArgs e)

{

if(e.ClickCount == 2)

{

TextBlock textBlock = sender as TextBlock;

if (textBlock.Name == "txtFIO")

{

txtFIO1.Visibility = Visibility.Visible;

txtFIO1.CaretIndex = txtFIO1.Text.Length;

txtFIO1.Focus();

}

if(textBlock.Name == "txtLogin")

{

txtLogin1.Visibility = Visibility.Visible;

txtLogin1.CaretIndex = txtLogin1.Text.Length;

txtLogin1.Focus();

}

if (textBlock.Name == "txtPasword")

{

txtPasword1.Visibility= Visibility.Visible;

txtPasword1.CaretIndex = txtPasword1.Text.Length;

txtPasword1.Focus();

}

}

}

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

{

ClassFrame.frmObj.Navigate(new PageAuthorization());

}

private void Back1\_Click(object sender, RoutedEventArgs e)

{

ClassFrame.frmObj.Navigate(new PageMenu());

}

}

***PageRegistration.xaml:***

<Grid>

<StackPanel HorizontalAlignment="Center">

<Label Content="Регистрация" Style="{StaticResource lblTitle}"/>

</StackPanel>

<StackPanel VerticalAlignment="Center" HorizontalAlignment="Center">

<Label Content="Введите ФИО:" Style="{StaticResource lblText2}"/>

<TextBox Name="FIO" Width="210" Height="25" />

<Label Content="Введите логин:" Style="{StaticResource lblText2}"/>

<TextBox Name="login" Width="210" Height="25" />

<Label Content="Введите пароль:" Style="{StaticResource lblText2}"/>

<PasswordBox Name="password" Width="210" Height="25"/>

<Label Content="Повторите пароль:" Style="{StaticResource lblText2}"/>

<PasswordBox Name="check" Width="210" Height="25" PasswordChanged="check\_PasswordChanged"/>

<Label Content="Выберите тип пользователя" Style="{StaticResource lblText2}"/>

<ComboBox Name="cmbRole" DisplayMemberPath="Nazv\_type"/>

<StackPanel Orientation="Horizontal" HorizontalAlignment="Center" Margin="0,10,0,0">

<Button Name="btnregistration" Content="Зарегистрироваться" Width="120" Height="25" Click="btnregistration\_Click">

<Button.Resources>

<Style TargetType="Border">

<Setter Property="CornerRadius" Value="10"></Setter>

</Style>

</Button.Resources>

</Button>

<Button Name="btnenter" Content="Войти" Width="55" Height="25" Margin="5,0,0,0" Click="btnenter\_Click">

<Button.Resources>

<Style TargetType="Border">

<Setter Property="CornerRadius" Value="10"></Setter>

</Style>

</Button.Resources>

</Button>

</StackPanel>

</StackPanel>

</Grid>

***PageRegistration.xaml.cs:***

public partial class PageRegistration : Page

{

public PageRegistration()

{

InitializeComponent();

cmbRole.ItemsSource = ClassFrame.db.TypesEmployee.ToList();

}

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

{

Employee employee = new Employee();

if (login.Text == "" && password.Password == "")

{

MessageBox.Show("Введите данные в поле","Ошибка ввода",MessageBoxButton.OK, MessageBoxImage.Error);

return;

}

employee.FIO = FIO.Text;

employee.Login\_Employee = login.Text;

employee.Password\_Employee = password.Password;

TypesEmployee role = (TypesEmployee)cmbRole.SelectedItem;

employee.IdTypesEmployee = role.IdTypesEmployee;

try

{

ClassFrame.db.Employee.Add(employee);

ClassFrame.db.SaveChanges();

MessageBox.Show("Все успешно добавлено");

ClassFrame.frmObj.Navigate(new PageAuthorization());

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

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

{

ClassFrame.frmObj.Navigate(new PageAuthorization());

}

private void check\_PasswordChanged(object sender, RoutedEventArgs e)

{

if (check.Password != password.Password)

{

btnregistration.IsEnabled = false;

check.Background = Brushes.LightCoral;

check.BorderBrush = Brushes.Red;

}

else

{

btnregistration.IsEnabled = true;

check.Background = Brushes.LightGreen;

check.BorderBrush = Brushes.Green;

}

}

}

***PagesListOfMaterials.xaml:***

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="1\*"/>

<RowDefinition Height="1\*"/>

<RowDefinition Height="5\*"/>

<RowDefinition Height="1\*"/>

</Grid.RowDefinitions>

<StackPanel Grid.Row="0" VerticalAlignment="Center" HorizontalAlignment="Left">

<Button x:Name="Back1" Click="Back1\_Click" Width="30" Height="30" BorderThickness="0">

<Button.Background>

<ImageBrush ImageSource="/Photo/iconback.png"/>

</Button.Background>

</Button>

</StackPanel>

<Label Content="Список материалов хранящихся на складе" Style="{StaticResource lblTitle}"/>

<StackPanel Grid.Row="1" Orientation="Horizontal" HorizontalAlignment="Center">

<Label Content="Поиск по номеру чертежа:" Height="35" Style="{StaticResource lblText1}"/>

<TextBox Name="TxtDrawingNumber" Width="160" Height="25" TextChanged="txtDrawingNumber\_TextChanged"/>

</StackPanel>

<DataGrid Name="dgStorage" Grid.Row="2" MouseDoubleClick="dgStorage\_MouseDoubleClick" AutoGenerateColumns="False" IsReadOnly ="True" >

<DataGrid.ContextMenu>

<ContextMenu>

<MenuItem Header="Обновить" Click="MenuUpdate\_Click">

<MenuItem.Icon>

<Image Source="/Photo/iconupdate.png"/>

</MenuItem.Icon>

</MenuItem>

<MenuItem Header="Фильтр">

<MenuItem.Icon>

<Image Source="/Photo/iconfiltr.png"/>

</MenuItem.Icon>

<MenuItem Header="Изделие" Click="MenuFilter\_Click"/>

<MenuItem Header="Узел" Click="MenuFilter\_Click"/>

<MenuItem Header="Деталь" Click="MenuFilter\_Click"/>

<MenuItem Header="Материал" Click="MenuFilter\_Click"/>

<Separator/>

<MenuItem Header="Сброс" Click="MenuClear\_Click"/>

</MenuItem>

</ContextMenu>

</DataGrid.ContextMenu>

<DataGrid.Columns>

<DataGridTextColumn Header="Название склада" Width="\*" Binding="{Binding Warehouse.Name}"/>

<DataGridTextColumn Header="Тип материала" Width="\*" Binding="{Binding Material.Type.Name}"/>

<DataGridTextColumn Header="Название материала" Width="\*" Binding="{Binding Material.Name}"/>

<DataGridTextColumn Header="Номер чертежа" Width="\*" Binding="{Binding Material.DrawingNumber}"/>

<DataGridTextColumn Header="Количество" Width="\*" Binding="{Binding Quantity}"/>

</DataGrid.Columns>

</DataGrid>

</Grid>

***PagesListOfMaterials.xaml.cs:***

public partial class ListOfMaterials : Page

{

public ListOfMaterials()

{

InitializeComponent();

dgStorage.ItemsSource = ClassFrame.db.Database.SqlQuery<Storage>("select \* from Nomenclature.dbo.Storage").ToList();

}

private void dgStorage\_MouseDoubleClick(object sender, MouseButtonEventArgs e)

{

if (dgStorage.SelectedItem != null)

{

Storage s = (Storage)dgStorage.SelectedItem;

if (Convert.ToString(s.Material.DrawingNumber) == "Отсутствует")

{

MessageBox.Show("Нельзя посмотреть так как номер чертежа отсутствует", "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

return;

}

else

{

ClassFrame.frmObj.Navigate(new PageStructure(s.idMaterial));

}

}

}

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

{

List<Storage> delet = dgStorage.SelectedItems.Cast<Storage>().ToList();

if(MessageBox.Show("Удалить данные","Внимание",MessageBoxButton.YesNo, MessageBoxImage.Question) == MessageBoxResult.Yes)

{

try

{

foreach (Storage s in delet) ClassFrame.db.Database.ExecuteSqlCommand("delete from Nomenclature.dbo.Storage where idStorage = @id", new SqlParameter("@id", s.idStorage));

dgStorage.ItemsSource = ClassFrame.db.Database.SqlQuery<Storage>("select \* from Nomenclature.dbo.Storage").ToList();

MessageBox.Show("Данные удаленны");

}

catch ( Exception ex)

{

MessageBox.Show(ex.Message.ToString());

}

}

}

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

{

ClassFrame.frmObj.Navigate(new PageMenu());

}

private void txtDrawingNumber\_TextChanged(object sender, TextChangedEventArgs e)

{

if (TxtDrawingNumber.Text.Count() != 0) dgStorage.ItemsSource = ClassFrame.db.Database.SqlQuery<Storage>($"select S.\* from Nomenclature.dbo.Storage S join Nomenclature.dbo.Materials M on S.idMaterial = M.idMaterial where M.DrawingNumber like '%{TxtDrawingNumber.Text.ToLower()}%'").ToList();

else dgStorage.ItemsSource = ClassFrame.db.Database.SqlQuery<Storage>("select \* from Nomenclature.dbo.Storage").ToList();

}

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

{

MenuItem m = sender as MenuItem;

switch (m.Header)

{

case "Изделие":

dgStorage.ItemsSource = ClassFrame.db.Database.SqlQuery<Storage>("select \* from Nomenclature.dbo.Storage").Where(x => x.Material.Type.Name == "Изделие").ToList();

break;

case "Узел":

dgStorage.ItemsSource = ClassFrame.db.Database.SqlQuery<Storage>("select \* from Nomenclature.dbo.Storage").Where(x => x.Material.Type.Name == "Узел").ToList();

break;

case "Деталь":

dgStorage.ItemsSource = ClassFrame.db.Database.SqlQuery<Storage>("select \* from Nomenclature.dbo.Storage").Where(x => x.Material.Type.Name == "Деталь").ToList();

break;

case "Материал":

dgStorage.ItemsSource = ClassFrame.db.Database.SqlQuery<Storage>("select \* from Nomenclature.dbo.Storage").Where(x => x.Material.Type.Name == "Материал").ToList();

break;

}

}

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

{

dgStorage.ItemsSource = ClassFrame.db.Database.SqlQuery<Storage>("select \* from Nomenclature.dbo.Storage").ToList();

}

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

{

dgStorage.ItemsSource = ClassFrame.db.Database.SqlQuery<Storage>("select \* from Nomenclature.dbo.Storage").ToList();

}

private void Back1\_Click(object sender, RoutedEventArgs e)

{

ClassFrame.frmObj.Navigate(new PageMenu());

}

}

***PageStructure.xaml:***

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="1\*"/>

<RowDefinition Height="5\*"/>

<RowDefinition Height="1\*"/>

</Grid.RowDefinitions>

<StackPanel Grid.Row="0" VerticalAlignment="Center" HorizontalAlignment="Right">

<Button x:Name="Back" Click="Back\_Click" Width="30" Height="30" BorderThickness="0">

<Button.Background>

<ImageBrush ImageSource="/Photo/iconhome.png"/>

</Button.Background>

</Button>

</StackPanel>

<StackPanel Grid.Row="0" VerticalAlignment="Center" HorizontalAlignment="Left">

<Button x:Name="Back1" Click="Back1\_Click" Width="30" Height="30" BorderThickness="0">

<Button.Background>

<ImageBrush ImageSource="/Photo/iconback.png"/>

</Button.Background>

</Button>

</StackPanel>

<Label Content="Состав" Style="{StaticResource lblTitle}"/>

<TreeView Name="trStructure" Grid.Row="1"/>

</Grid>

***PageStructure.xaml.cs:***

public partial class PageStructure : Page

{

public readonly Materials \_material = new Materials();

public PageStructure(int idMaterial)

{

InitializeComponent();

\_material = ClassFrame.db.Database.SqlQuery<Materials>("select \* from Nomenclature.dbo.Materials").FirstOrDefault(x => x.idMaterial == idMaterial);

foreach (Hierarchy hierarchy in ClassFrame.db.Database.SqlQuery<Hierarchy>("select \* from Nomenclature.dbo.Hierarchy").Where(x => x.idParent == \_material.idMaterial).ToList())

{

TreeViewItem tree = new TreeViewItem();

tree.Header = hierarchy.Material1.Name;

foreach(Hierarchy hierarchy1 in ClassFrame.db.Database.SqlQuery<Hierarchy>("select \* from Nomenclature.dbo.Hierarchy").Where(x => x.idParent == hierarchy.Material1.idMaterial).ToList())

{

TreeViewItem tree1 = new TreeViewItem();

tree1.Header = hierarchy1.Material1.Name;

tree.Items.Add(tree1);

}

trStructure.Items.Add(tree);

}

}

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

{

ClassFrame.frmObj.Navigate(new ListOfMaterials());

}

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

{

ClassFrame.frmObj.Navigate(new PageMenu());

}

private void Back1\_Click(object sender, RoutedEventArgs e)

{

ClassFrame.frmObj.Navigate(new ListOfMaterials());

}

}

***User.xaml:***

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="1\*"/>

<RowDefinition Height="2\*"/>

<RowDefinition Height="5\*"/>

<RowDefinition Height="1\*"/>

</Grid.RowDefinitions>

<StackPanel Grid.Row="0" VerticalAlignment="Center" HorizontalAlignment="Left">

<Button x:Name="Back1" Click="Back1\_Click" Width="30" Height="30" BorderThickness="0">

<Button.Background>

<ImageBrush ImageSource="/Photo/iconback.png"/>

</Button.Background>

</Button>

</StackPanel>

<Label Content="Список пользователей" Style="{StaticResource lblTitle}"/>

<StackPanel Grid.Row="1" HorizontalAlignment="Left" VerticalAlignment="Bottom" >

<Label Content="Поиск:" Style="{StaticResource lblText1}"/>

<TextBox Name="txtSearch" Width="125" Height="20" TextChanged="txtSearch\_TextChanged" Margin="10"/>

</StackPanel>

<DataGrid Name="dgUser" Grid.Row="2" AutoGenerateColumns="False" IsEnabled="True" CanUserAddRows="False" IsReadOnly="True">

<DataGrid.ContextMenu>

<ContextMenu>

<MenuItem Header="Обновить" Click="MenuUpdate\_Click">

<MenuItem.Icon>

<Image Source="/Photo/iconupdate.png"/>

</MenuItem.Icon>

</MenuItem>

<MenuItem Header="Добавить" Click="MenuAdd\_Click">

<MenuItem.Icon>

<Image Source="/Photo/iconadd.png"/>

</MenuItem.Icon>

</MenuItem>

<MenuItem Header="Удалить" Click="MenuDelet\_Click">

<MenuItem.Icon>

<Image Source="/Photo/icondelet.png"/>

</MenuItem.Icon>

</MenuItem>

<MenuItem Header="Изменить" Click="MenuItem\_Click">

<MenuItem.Icon>

<Image Source="/Photo/iconchange.png"/>

</MenuItem.Icon>

</MenuItem>

<MenuItem Header="Фильтр">

<MenuItem.Icon>

<Image Source="/Photo/iconfiltr.png"/>

</MenuItem.Icon>

<MenuItem Header="Администратор" Click="MenuFilter\_Click"/>

<MenuItem Header="Сотрудник" Click="MenuFilter\_Click"/>

<MenuItem Header="Кладовщик" Click="MenuFilter\_Click"/>

<Separator/>

<MenuItem Header="Сброс" Click="MenuClear\_Click"/>

</MenuItem>

</ContextMenu>

</DataGrid.ContextMenu>

<DataGrid.Columns>

<DataGridTextColumn Header="ФИО пользователя" Width="\*" Binding="{Binding FIO}"/>

<DataGridTextColumn Header="Логин пользователя" Width="\*" Binding="{Binding Login\_Employee}"/>

<DataGridTextColumn Header="Тип пользователя" Width="\*" Binding ="{Binding TypesEmployee.Nazv\_type}"/>

</DataGrid.Columns>

</DataGrid>

</Grid>

ConnectODB.cs:

class ConnectODB

{

public static EquipmentEntities equipObj = new EquipmentEntities();

public static int id\_entered\_role;

public static int id\_entered\_fio;

}

***User.xaml.cs:***

public partial class User : Page

{

public User()

{

InitializeComponent();

dgUser.ItemsSource = ClassFrame.db.Employee.ToList();

}

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

{

ClassFrame.frmObj.Navigate(new PageMenu());

}

private void txtSearch\_TextChanged(object sender, TextChangedEventArgs e)

{

if (txtSearch.Text.Count() != 0)

dgUser.ItemsSource = ClassFrame.db.Employee.Where(x => x.FIO.Contains(txtSearch.Text.ToLower()) || x.Login\_Employee.Contains(txtSearch.Text.ToLower()) || x.TypesEmployee.Nazv\_type.Contains(txtSearch.Text.ToLower())).ToList();

else dgUser.ItemsSource = ClassFrame.db.Employee.ToList();

}

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

{

ClassFrame.frmObj.Navigate(new AddUser((Employee)dgUser.SelectedItem));

}

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

{

MenuItem m = sender as MenuItem;

switch (m.Header)

{

case "Администратор":

dgUser.ItemsSource = ClassFrame.db.Employee.Where(x => x.TypesEmployee.Nazv\_type == "Администратор").ToList();

break;

case "Сотрудник":

dgUser.ItemsSource = ClassFrame.db.Employee.Where(x => x.TypesEmployee.Nazv\_type == "Сотрудник").ToList();

break;

case "Кладовщик":

dgUser.ItemsSource = ClassFrame.db.Employee.Where(x => x.TypesEmployee.Nazv\_type == "Кладовщик").ToList();

break;

}

}

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

{

dgUser.ItemsSource = ClassFrame.db.Employee.ToList();

}

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

{

dgUser.ItemsSource = ClassFrame.db.Employee.ToList();

}

private void Back1\_Click(object sender, RoutedEventArgs e)

{

ClassFrame.frmObj.Navigate(new PageMenu());

}

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

{

ClassFrame.frmObj.Navigate(new AddUser(null));

}

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

{

List<Employee> delet = dgUser.SelectedItems.Cast<Employee>().ToList();

if (MessageBox.Show("Удалить пользователя", "Внимание", MessageBoxButton.YesNo, MessageBoxImage.Question) == MessageBoxResult.Yes)

{

try

{

ClassFrame.db.Employee.RemoveRange(delet);

ClassFrame.db.SaveChanges();

MessageBox.Show("Данные удаленны");

dgUser.ItemsSource = ClassFrame.db.Employee.ToList();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message.ToString());

}

}

}

}

***Сounterparties.xaml:***

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="1\*"/>

<RowDefinition Height="2\*"/>

<RowDefinition Height="5\*"/>

<RowDefinition Height="1\*"/>

</Grid.RowDefinitions>

<StackPanel Grid.Row="0" VerticalAlignment="Center" HorizontalAlignment="Left">

<Button x:Name="Back1" Click="Back1\_Click" Width="30" Height="30" BorderThickness="0">

<Button.Background>

<ImageBrush ImageSource="/Photo/iconback.png"/>

</Button.Background>

</Button>

</StackPanel>

<Label Content="Список контрагентов" Style="{StaticResource lblTitle}"/>

<StackPanel Grid.Row="1" HorizontalAlignment="Left" VerticalAlignment="Bottom" >

<Label Content="Поиск:" Foreground="DarkSlateBlue" FontSize="18"/>

<TextBox Name="txtSearch" Width="125" Height="20" TextChanged="txtSearch\_TextChanged" Margin="10"/>

</StackPanel>

<DataGrid Name="dgСounterparties" Grid.Row="2" AutoGenerateColumns="False" IsEnabled="True" CanUserAddRows="False" IsReadOnly="True">

<DataGrid.ContextMenu>

<ContextMenu>

<MenuItem Header="Обновить" Click="MenuUpdate\_Click">

<MenuItem.Icon>

<Image Source="/Photo/iconupdate.png"/>

</MenuItem.Icon>

</MenuItem>

<MenuItem Header="Добавить" Click="MenuAdd\_Click">

<MenuItem.Icon>

<Image Source="/Photo/iconadd.png"/>

</MenuItem.Icon>

</MenuItem>

<MenuItem Header="Удалить" Click="MenuDelet\_Click">

<MenuItem.Icon>

<Image Source="/Photo/icondelet.png"/>

</MenuItem.Icon>

</MenuItem>

<MenuItem Header="Изменить" Click="MenuItem\_Click">

<MenuItem.Icon>

<Image Source="/Photo/iconchange.png"/>

</MenuItem.Icon>

</MenuItem>

<MenuItem Header="Фильтр">

<MenuItem.Icon>

<Image Source="/Photo/iconfiltr.png"/>

</MenuItem.Icon>

<MenuItem Header="ООО" Click="MenuFilter\_Click"/>

<MenuItem Header="АО" Click="MenuFilter\_Click"/>

<Separator/>

<MenuItem Header="Сброс" Click="MenuClear\_Click"/>

</MenuItem>

</ContextMenu>

</DataGrid.ContextMenu>

<DataGrid.Columns>

<DataGridTextColumn Header="Название контрагента" Width="2\*" Binding="{Binding Nazv\_Сounterparties}"/>

<DataGridTextColumn Header="ИНН" Width="1\*" Binding="{Binding INN}"/>

<DataGridTextColumn Header="Адрес" Width="4\*" Binding ="{Binding Adress\_Сounterparties}"/>

</DataGrid.Columns>

</DataGrid>

</Grid>

***Сounterparties.xaml.cs:***

public partial class Сounterparties : Page

{

public Сounterparties()

{

InitializeComponent();

dgСounterparties.ItemsSource = ClassFrame.db.Сounterparties.ToList();

}

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

{

ClassFrame.frmObj.Navigate(new PageMenu());

}

private void txtSearch\_TextChanged(object sender, TextChangedEventArgs e)

{

if (txtSearch.Text.Count() != 0)

dgСounterparties.ItemsSource = ClassFrame.db.Сounterparties.Where(x => x.Nazv\_Сounterparties.Contains(txtSearch.Text.ToLower()) || x.INN.Contains(txtSearch.Text.ToLower()) || x.Adress\_Сounterparties.Contains(txtSearch.Text.ToLower())).ToList();

else dgСounterparties.ItemsSource = ClassFrame.db.Сounterparties.ToList();

}

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

{

ClassFrame.frmObj.Navigate(new AddСounterparties(null));

}

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

{

ClassFrame.frmObj.Navigate(new AddСounterparties((Classes.Сounterparties)dgСounterparties.SelectedItem));

}

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

{

MenuItem m = sender as MenuItem;

switch (m.Header)

{

case "ООО":

dgСounterparties.ItemsSource = ClassFrame.db.Сounterparties.Where(x => x.Nazv\_Сounterparties == "ООО").ToList();

break;

case "АО":

dgСounterparties.ItemsSource = ClassFrame.db.Сounterparties.Where(x => x.Nazv\_Сounterparties == "АО").ToList();

break;

}

}

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

{

dgСounterparties.ItemsSource = ClassFrame.db.Сounterparties.ToList();

}

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

{

dgСounterparties.ItemsSource = ClassFrame.db.Сounterparties.ToList();

}

private void Back1\_Click(object sender, RoutedEventArgs e)

{

ClassFrame.frmObj.Navigate(new PageMenu());

}

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

{

List<Classes.Сounterparties> delet = dgСounterparties.SelectedItems.Cast<Classes.Сounterparties>().ToList();

if (MessageBox.Show("Удалить контрагента", "Внимание", MessageBoxButton.YesNo, MessageBoxImage.Question) == MessageBoxResult.Yes)

{

try

{

ClassFrame.db.Сounterparties.RemoveRange(delet);

ClassFrame.db.SaveChanges();

MessageBox.Show("Данные удаленны");

dgСounterparties.ItemsSource = ClassFrame.db.Сounterparties.ToList();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message.ToString());

}

}

}

}