федеральное государственное бюджетное образовательное учреждение высшего образования

«**Вологодский государственный университет**»

университетский колледж

Информационные системы и программирование

(наименование специальности)

**КУРСОВОЙ ПРОЕКТ**

Дисциплина: «МДК 01.01 Разработка программных модулей»

Наименование темы: «Разработка модуля информационной системы универсального интернет-магазина Wildberries»

Код работы КП 09.02.07.13.00.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Код специальности, код УК, регистрационный номер по журналу, год

Руководитель Дегтярёв Михаил Евгеньевич

(уч. степень, звание, должность. Ф.И.О)

Выполнил (а) обучающийся Плотников Алексей Николаевич

(Ф.И.О)

Группа, курс ИСП 31-20, 3 курс

Дата сдачи \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дата защиты \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Оценка по защите \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись преподавателя)

Вологда

2023 г.

федеральное государственное бюджетное образовательное учреждение высшего образования

«**Вологодский государственный университет**»

университетский колледж

**Задание на курсовой проект**

По МДК 01.01 Разработка программных модулей

Обучающемуся Плотникову Алексею Николаевичу

Специальности 09.02.07 Информационные системы и программирование

Тема: «Разработка модуля информационной системы универсального интернет-магазина Wildberries»

Исходные данные к курсовому проекту: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Содержание и объём курсового проекта: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Председатель П(Ц)К\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/

Руководитель курсового проекта\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/

Обучающийся\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/

Дата выдачи задания«\_\_»\_\_\_20\_\_г.

Срок выполнения«\_\_»\_\_\_20\_\_г.

Курсовой проект защищен с оценкой\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/

(Подпись руководителя)

**Содержание**

Введение 4

1 АНАЛИТИЧЕСКИЙ ОБЗОР

* 1. Описание требований
  2. Выбор средств разработки
  3. Описание базы данных

2 РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

2.1 Разработка графического интерфейса

2.2 Разработка программного функционала

3 ТЕСТИРОВАНИЕ

4 РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

ЗАКЛЮЧЕНИЕ

Список использованных источников

ПриложениЯ

ВВЕДЕНИЕ

Тема курсового проекта: «Разработка модуля информационной системы универсального интернет-магазина Wildberries».

Работа интернет-магазина Wildberries состоит из множества сложных процессов. Постоянно происходят оформления заказов, перевозки и сортировки товаров. Для успешного существования бизнеса необходимо наладить контроль над этими процессами, обладать достаточным набором данных. Интернет-магазин Wildberries обладает базой данных со всей необходимой информацией, но сотрудники не должны иметь прямого доступа к ней. Актуальность проекта заключается в том, что благодаря будущему модулю информационной системы можно повысить качество бизнеса. Информационная система может предоставить сотрудникам только необходимые данные в понятной форме.

Интернет-магазин Wildberries занимается продажей и доставкой товаров. Wildberries работает в нескольких странах: в России, Казахстане, Беларуси, Киргизии, Армении и Узбекистане. Товары доставляются в пункты выдачи. Пользователь может выбирать сам, в какой пункт выдачи доставить товар. Чтобы забрать товар, пользователю нужно прийти в пункт выдачи и показать сотруднику QR-код или назвать свои фамилию, имя и отчество и трёхзначный код, который меняется каждые сутки. У покупателя один код действует на все товары. Интернет-магазин владеет множеством сортировочных складов, где происходит распределение товаров и выбор дальнейшего их маршрута. На складах и пунктах выдачи работают сотрудники интернет-магазина. В приложении Wildberries можно посмотреть историю доставки заказа, показывающую, когда и как менялся статус доставки, например, на какой разгрузочный склад он поступил и в какое время. К одному изменению статуса доставки могут быть причастны несколько сотрудников, например, доставить товар на сортировочный склад могут несколько людей. В интернет-магазине товары имеют категории, которые могут в свою очередь иметь подкатегории. У товаров могут быть различные параметры, например цвет, вес, количество. Также у продуктов есть описание. Пользователь способен оставить отзыв под товаром, если он был им приобретён. Пользователи могут «отложить» товары, которые хотят купить позже. Товары имеют поставщиков, которых можно посмотреть в приложении, а также есть возможность узнать все товары, которые они поставляют. Поставщики поставляют свои товары на склады интернет-магазина и получают проценты с продаж этих товаров. Приложение позволяет посмотреть историю цен на товар.

Цель работы: разработка информационной системы для интернет-магазина Wildberries.

Задачи:

* описание требований;
* выбор средств разработки;
* разработка графического интерфейса;
* разработка программного функционала;
* тестирование;
* написание руководство пользователя.

1 АНАЛИТИЧЕСКИЙ ОБЗОР

* 1. Описание требований

Требования к разрабатываемому программному обеспечению находятся в приложении 1.

* 1. Выбор средств разработки

Перед выполнением разработки информационной системы необходимо определиться с тем, с помощью каких средств она будет разрабатываться. Было решено начать выбор с языка программирования.

В настоящее время для разработки настольных приложений используются такие языки программирования, как C++, Java, C#, Python. Но для разработки качественных настольных приложений с этими языками используются дополнительные фреймворки и библиотеки.

Популярным фреймворком для разработки настольных приложений на C++ является Qt. На данный момент последняя версия - шестая. Qt позволяет создавать кроссплатформенные приложения. Данный фреймворк содержит модули не только для работы с графическим интерфейсом, но также и для работы с базами данных, сетью, звуком, XML-файлами. К недостаткам Qt можно отнести то, что приложения с его использованием могут занимать много памяти.

JavaFX – платформа на основе Java для создания приложений с графическим интерфейсом. Благодаря тому, что программы на Java могут работать на любых устройствах, на которых есть виртуальная машина Java (JVM), JavaFX позволяет создавать кроссплатформенные приложения с графическим интерфейсом. Из-за JVM приложения могут иметь проблемы с оптимизацией и производительностью. В текущий момент данная платформа теряет популярность.

На C# для разработки приложений с графическим интерфейсом используются Windows Forms и WPF.

Windows Forms – интерфейс программирования приложений (API), позволяющий создавать графический интерфейс пользователя. Windows Forms является частью Microsoft .NET Framework. Windows Forms позволяет создавать приложения только под операционную систему Windows из-за использования обёртки для Win32 API. Данное API легко в освоении.

WPF (Windows Presentation Foundation) – система для построения клиентских приложений в операционной системе Windows. В основе WPF лежит векторная система визуализации, благодаря которой можно создавать графический интерфейс, не зависящий от разрешения устройства. WPF, использует DirectX, имеющий аппаратное ускорения графики, за счёт чего обладает высокой производительностью. Приложения на WPF могут потреблять больше памяти, относительно программ на Windows Form. Также у приложений на WPF отсутствует кроссплатформенность.

Для разработки настольных приложений на Python используются такие средства разработки, как tkinter, PyQt. Tkinter – графическая библиотека, входящая в состав стандартных библиотек Python. Обладает небольшим набором графических компонентов, достаточно проста в изучении. Подходит для разработки несложных приложений.

PyQt представляет собой набор расширений Qt для языка программирования Python. Обладает широким набором возможностей, позволяет разрабатывать кроссплатформенные приложения. К минусам можно отнести большое время, необходимое для изучения возможностей, а также дорогостоящую лицензию для коммерческого использования.

В связи со сложностью будущего приложения, tkinter явно не подходит для разработки из-за малого набора возможностей. JavaFX теряет популярность и имеет проблемы с производительностью, из-за чего могут возникнуть трудности с поддержкой приложения в дальнейшем. Qt, PyQt, Windows Forms и WPF имеют множество преимуществ, но так как будущее приложение не требует наличия кроссплатформенности и предназначено для операционных систем Windows, для разработки больше подходят Windows Forms и WPF. WPF является современной технологией, с множеством полезных возможностей, что делает его более приоритетным, чем Windows Forms. Таким образом, для разработки приложения выбран язык C# с фреймворком WPF.

Далее необходимо определиться с интегрированной средой разработки, в которой будет разрабатываться будущее приложение. Наиболее удобными и популярными средами разработки для создания приложений на WPF являются Visual Studio и JetBrains Rider.

Visual Studio содержит редактор исходного кода, технологию IntelliSense, способную автоматически дописывать код. Visual Studio предоставляет возможности рефакторинга и отладки кода. Данная среда разработки имеет редактор форм для создания графического интерфейса приложения. Visual Studio позволяет создавать и подключать сторонние дополнения, расширяющие базовые возможности среды разработки. Имеет бесплатную версию Community для обучения.

Rider является кроссплатформенной средой разработки и может работать в Windows, Linux, MacOS. Rider имеет более продвинутые средства рефакторинга, чем Visual Studio. Позволяет работать с базами данных прямиком из среды разработки. Имеет широкий набор дополнений. Имеет средства запуска юнит-тестов. Использование Rider требует наличие лицензии, но также есть бесплатная пробная версия на 30 дней.

Итак, JetBrains Rider имеет множество преимуществ, но из-за необходимости в лицензии для его использования было принято решение воспользоваться средой разработки Visual Studio, конкретнее Visual Studio Community 22.

Далее необходимо определиться с системой управления баз данных (СУБД), в которой будут храниться данные приложения. Наиболее популярными СУБД являются MySQL, Microsoft SQL Server, Oracle Database и PostgreSQL.

MySQL хорош тем, что прост в работе, имеет широкий набор функций для обеспечения безопасности, а также обладает высокой скоростью выполнения операций. Недостатки MySQL заключаются в ограниченных возможностях относительно других СУБД, а также в проблемах с надёжностью при выполнении транзакций и параллельных операций.

Преимущества Microsoft SQL Server, заключаются в том, что она оптимизирована под операционную систему Windows, способна взаимодействовать с другими продуктами от Microsoft, такими как Excel, предоставляя возможность импортировать хранившиеся данные из электронных таблиц. Также данная СУБД обладает высокой отказоустойчивостью. Недостатком является высокая стоимость лицензии, но есть бесплатная версия – Microsoft SQL Server Express, имеющая ограничения в используемой памяти и производительности.

Oracle Database обладает широким набором возможностей. Данная СУБД способна обеспечивать безопасность и надёжность баз данных. Широко распространена среди крупных компаний. Недостатками Oracle являются высокая сложность и стоимость приобретения лицензии, большие требования к оборудованию.

PostgreSQL – это бесплатная СУБД, имеющая открытый исходный код. Она предоставляет разработчикам дополнительные возможности в разработке относительно остальных СУБД благодаря объектно-реляционной модели данных. Недостаток данной системы управления базами данных – меньшая производительность по сравнению с другими СУБД.

MySQL не подходит из-за возможных проблем с параллельными операциями при большом числе пользователей. Microsoft SQL Server имеет много преимуществ. Oracle является хорошим вариантом, но обладает большой ценой на лицензию, а также высоким порогом входа. PostgreSQL обладает меньшей производительностью, чем ранее перечисленные СУБД. Из-за необходимости приобретать лицензию для Microsoft SQL Server было решено использовать его бесплатную версию Microsoft SQL Server 2022 Express Edition. Также для того, чтобы использовать данную СУБД, необходимо установить утилиту для администрирования её компонентов– Microsoft SQL Management Studio.

* 1. Описание базы данных

ER-диаграмма базы данных представлена в соответствии с рисунками ХХ, ХХ, ХХ и тд. Таблицы в базе данных:

* cards (платёжные карты);
* categories (категории товаров);
* countries (страны);
* deferred\_products (товары в корзинах пользователей);
* order\_history (история изменений статусов заказа);
* order\_statuses (статусы доставки заказа);
* orders (доставки);
* posts (должности сотрудников);
* price\_history (история цен на товары);
* product\_images (изображения товаров);
* products (товары);
* products\_on\_storages (количество товаров на складах);
* products\_parameters (параметры товаров);
* receipt\_of\_products\_to\_storages (поступления товаров на склады);
* review\_images (изображения к отзывам о доставках товаров);
* reviews (отзывы на доставки товаров);
* rights (права на действия в разделах);
* section\_rights (права должностей на разделы информационной системы);
* sections (разделы информационной системы);
* storage\_types (типы складов);
* storage\_worker\_shifts (смены сотрудников на складах);
* storages (склады);
* suppliers (поставщики);
* users (пользователи);
* workers (сотрудники);
* workers\_in\_orders (участия сотрудников в изменениях статуса доставки).

Таблицы users, countries, cards содержат информацию о пользователях, странах их проживания, платёжных картах. Они нужны для формирования заказов и анализа сведений о клиентах интернет-магазина.

Таблицы products, categories, suppliers, products\_parameters, product\_images и price\_history содержат информацию о товарах, их категориях, поставщиках, параметрах, изображениях и истории изменения цен. Они помогают управлять ассортиментом товаров, отслеживать изменения в ценах.

Таблицы orders, order\_history, order\_statuses, deferred\_products, reviews и review\_images содержат информацию о заказах, их статусах, содержимом корзин, истории изменений статусов заказов, отзывах о доставках товаров и изображениях к отзывам. Это помогает магазину управлять заказами, контролировать их выполнение и собирать обратную связь от клиентов.

Таблицы storages, storage\_types, products\_on\_storages и receipt\_of\_products\_to\_storages, storage\_worker\_shifts содержат информацию о складах, типах складов, количестве товаров на складах, поступлениях товаров на склады и смены сотрудников на складах. Они служат для контроля запасов, планирования закупок, отслеживания поставок товаров, а также контроля работы сотрудников.

Таблицы workers, posts, section\_rights и rights содержат информацию о сотрудниках, их должностях и правах на действия в разделах информационной системы. Это помогает управлять доступом к функционалу системы и контролировать действия сотрудников.

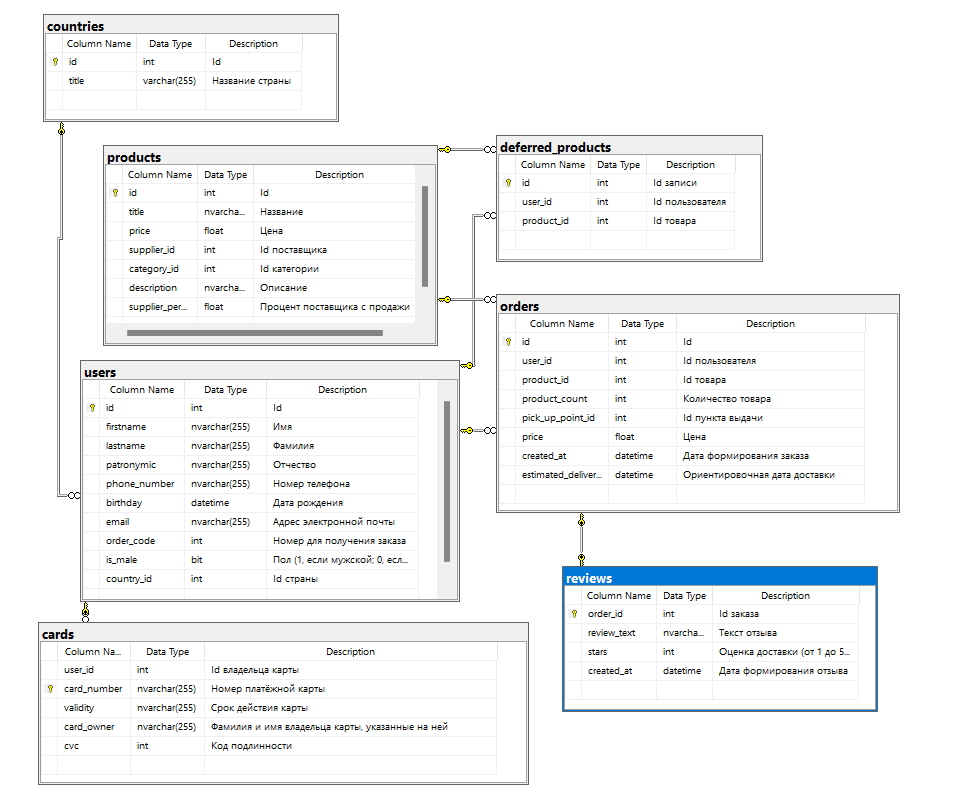


Рисунок 1 - ER-диаграмма базы данных.

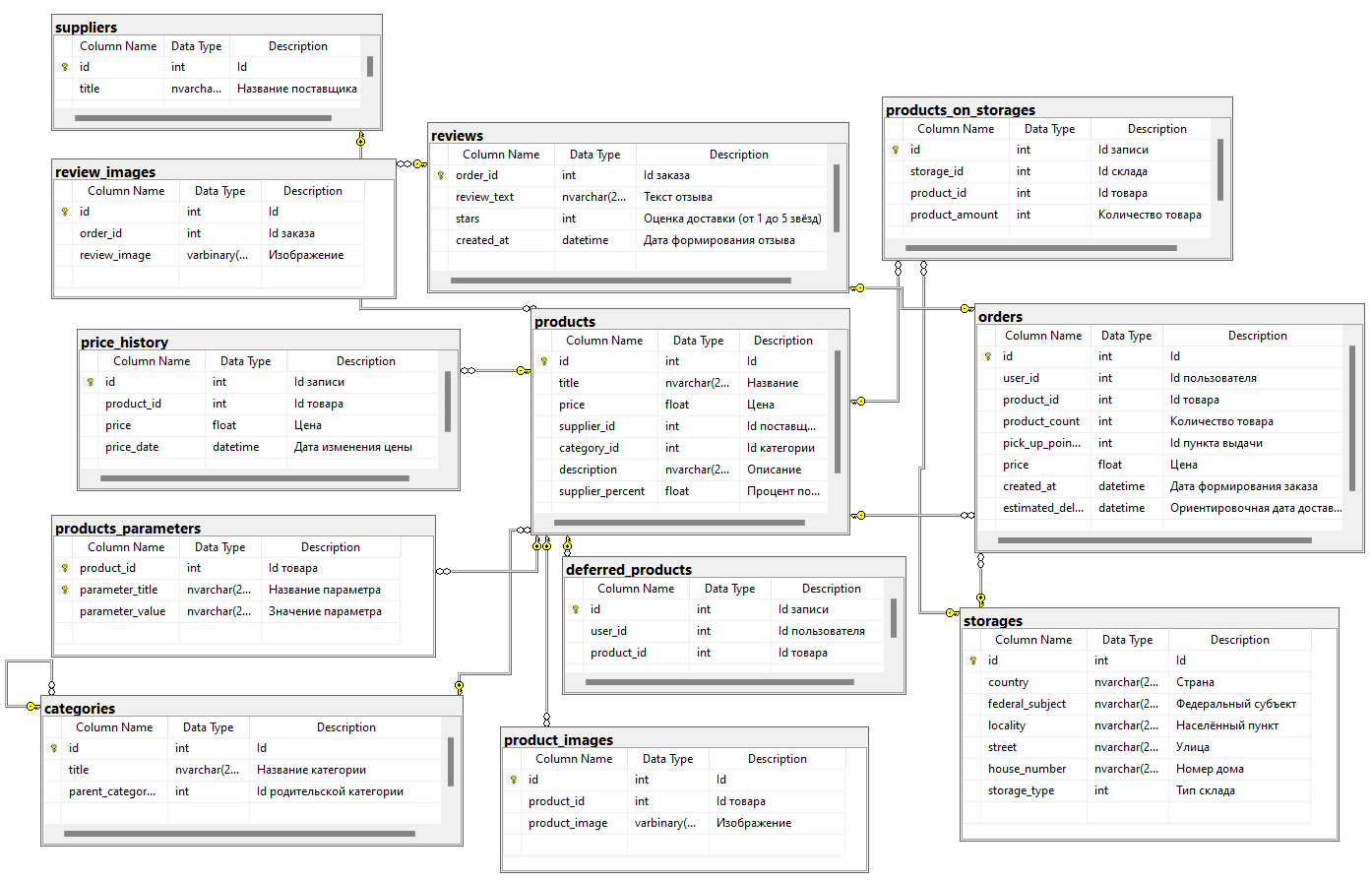


Рисунок 2 - ER-диаграмма базы данных.

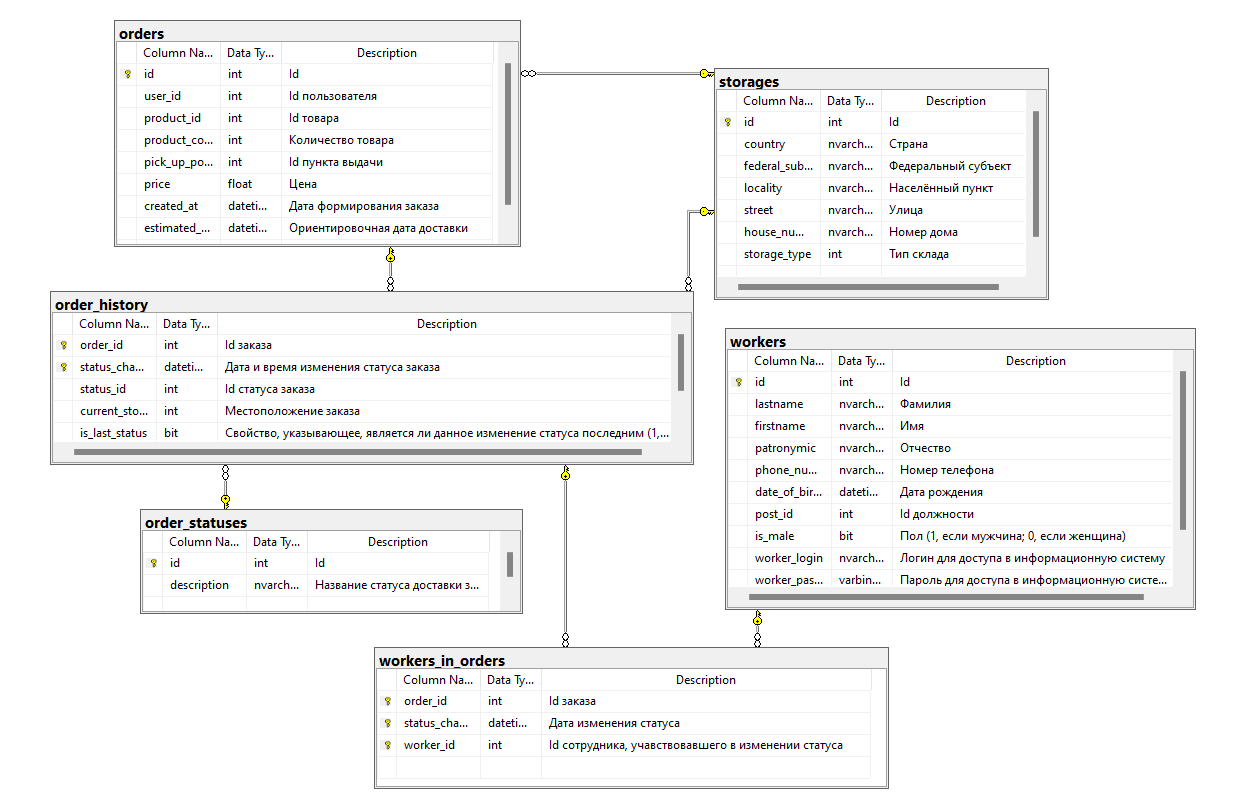


Рисунок 3- ER-диаграмма базы данных.

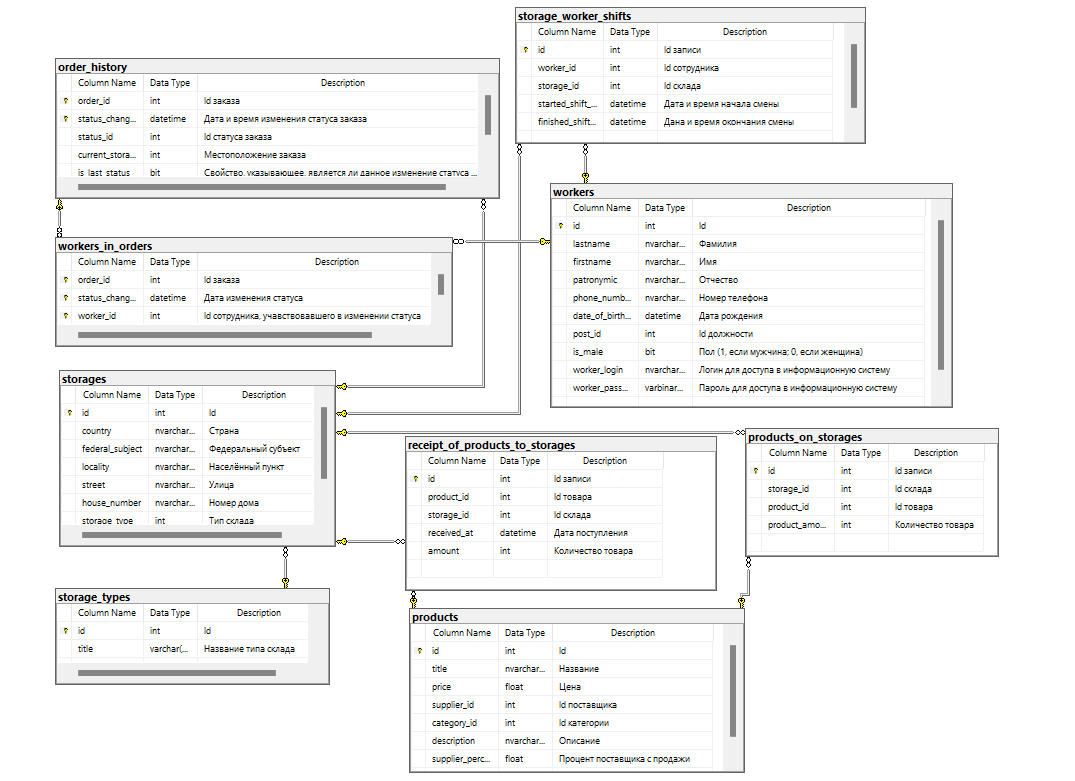


Рисунок 4 - ER-диаграмма базы данных.

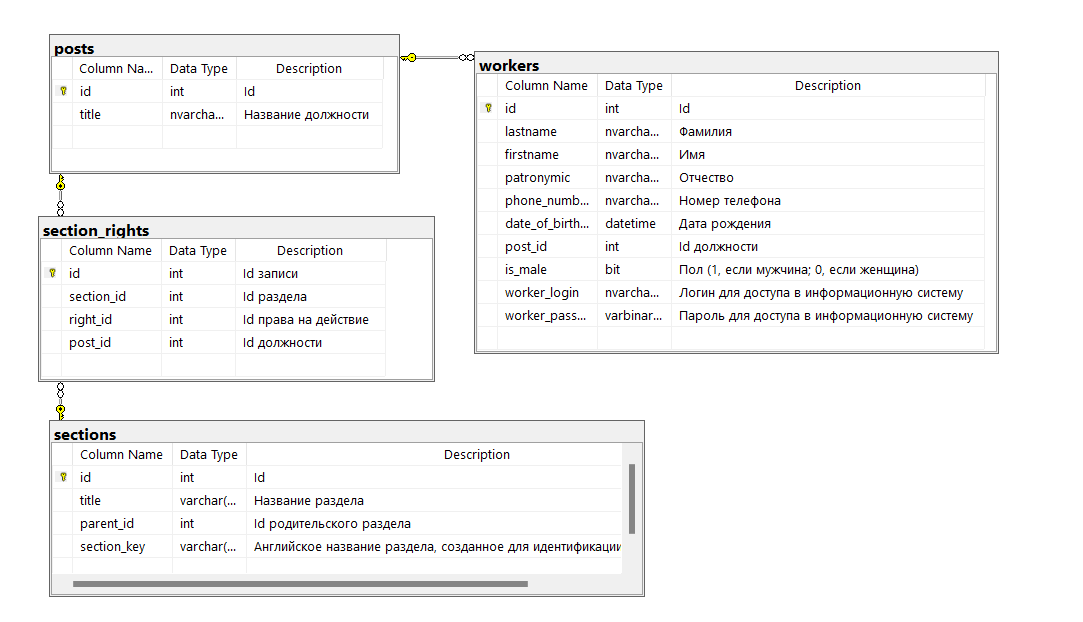


Рисунок 5 - ER-диаграмма базы данных.

2 РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

2.1 Разработка графического интерфейса

Для начала были написаны стили и шаблоны для элементов управления в приложении: для кнопки – шаблон commonButton, для выпадающего списка - стиль CommonComboBox, для элемента управления меню – стиль CommonMenu, для текстового поля ввода – стиль CommonTextBox, для поля ввода пароля – PasswordBox. Стили были вынесены в разные файлы, для более простого поиска и поддержки. Пример стиля CommonTextBox:

*<SolidColorBrush x:Key="CommonColor" Color="#CB11AB"/>*

*<Style TargetType="TextBox" x:Key="CommonTextBox" >*

*<Setter Property="BorderBrush" Value="{StaticResource CommonColor}" />*

*<Setter Property="BorderThickness" Value="2"/>*

*<Style.Resources>*

*<Style TargetType="{x:Type Border}">*

*<Setter Property="CornerRadius" Value="5" />*

*</Style>*

*</Style.Resources>*

*</Style>*

Все стили подключаются в файле App.xaml, так как используются в большинстве окон приложения. Подключение стилей:

*<Application x:Class="WpfApp1.App"*

*xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"*

*xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"*

*xmlns:local="clr-namespace:WpfApp1"*

*ShutdownMode="OnMainWindowClose"*

*DispatcherUnhandledException="Application\_DispatcherUnhandledException">*

*<Application.Resources>*

*<ResourceDictionary>*

*<ResourceDictionary.MergedDictionaries>*

*<ResourceDictionary Source="/Resources/ButtonTheme.xaml"></ResourceDictionary>*

*<ResourceDictionary Source="/Resources/ComboBoxTheme.xaml"></ResourceDictionary>*

*<ResourceDictionary Source="/Resources/TextBoxTheme.xaml"></ResourceDictionary>*

*<ResourceDictionary Source="/Resources/MenuTheme.xaml"/>*

*</ResourceDictionary.MergedDictionaries>*

*</ResourceDictionary>*

*</Application.Resources>*

*</Application>*

Окно входа в систему (Login).

Окно входа в систему содержит поля ввода логина и пароля, а также кнопку «Войти», в соответствие с рисунком ХХ. Код окна входа:

*<Window x:Class="WpfApp1.Views.Login"*

*xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"*

*xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"*

*xmlns:d="http://schemas.microsoft.com/expression/blend/2008"*

*xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"*

*mc:Ignorable="d"*

*Title="Вход" Height="185"*

*Width="471" Icon="\Resources\wb.ico"*

*ResizeMode="NoResize">*

*<Grid VerticalAlignment="Top">*

*<Grid.ColumnDefinitions>*

*<ColumnDefinition Width="80"/>*

*<ColumnDefinition Width="0,8\*"/>*

*</Grid.ColumnDefinitions>*

*<Grid.RowDefinitions>*

*<RowDefinition />*

*<RowDefinition />*

*<RowDefinition />*

*</Grid.RowDefinitions>*

*<Label Content="Логин" Grid.Column="0"*

*Grid.Row="0" FontSize="16"*

*HorizontalAlignment="Left"*

*Height="31"*

*Margin="10 10 10 0"*

*VerticalAlignment="Top"*

*Width="72" Grid.ColumnSpan="2"/>*

*<TextBox Text="{Binding UserLogin, UpdateSourceTrigger=PropertyChanged}"*

*HorizontalAlignment="Left"*

*Grid.Column="1" Grid.Row="0"*

*FontSize="16" Height="31"*

*Margin="10 10 0 0"*

*TextWrapping="Wrap"*

*VerticalAlignment="Top"*

*Width="348" Style="{StaticResource CommonTextBox}"/>*

*<Label Content="Пароль"*

*Grid.Column="0" Grid.Row="1"*

*FontSize="16" HorizontalAlignment="Left"*

*Height="31"*

*Margin="10 10 10 0"*

*VerticalAlignment="Top"*

*Width="72" Grid.ColumnSpan="2"/>*

*<PasswordBox x:Name="tbPassword"*

*HorizontalAlignment="Left"*

*Grid.Column="1" Grid.Row="1"*

*FontSize="16" Height="31"*

*Margin="10 10 0 0"*

*VerticalAlignment="Top"*

*Width="348" Style="{StaticResource CommonPasswordBox}"/>*

*<Button Template="{StaticResource commonButton}"*

*Command="{Binding AuthenticationCommand}"*

*CommandParameter="{Binding ElementName = tbPassword}"*

*Grid.Column="0" Grid.Row="2"*

*Grid.ColumnSpan="2"*

*FontSize="16" HorizontalAlignment="Center"*

*VerticalAlignment="Top"*

*Margin="5" Width="117">Войти</Button>*

*</Grid>*

*</Window>*

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 6 - Окно входа в систему.

Главное окно приложения (MainWindow).

Главное окно приложения содержит меню, в котором находится список доступных вошедшему сотруднику разделов в соответствие с рисунком ХХ. Также в главном окне должны отображаться вкладки с разделами. При выборе раздела в меню, в главном окне должна появляться вкладка с соответствующим разделом. Код формы главного окна:

*<Window x:Class="WpfApp1.Views.MainWindow"*

*xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"*

*xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"*

*xmlns:d="http://schemas.microsoft.com/expression/blend/2008"*

*xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"*

*mc:Ignorable="d"*

*Title="Информационная система"*

*Height="529" Width="877"*

*Loaded="Window\_Loaded" Closing="Window\_Closing"*

*Icon="\Resources\wb.ico">*

*<Grid>*

*<Grid.ColumnDefinitions>*

*<ColumnDefinition Width="\*"/>*

*</Grid.ColumnDefinitions>*

*<Grid.RowDefinitions>*

*<RowDefinition Height="21"></RowDefinition>*

*<RowDefinition Height="\*"></RowDefinition>*

*</Grid.RowDefinitions>*

*<Menu HorizontalAlignment="Stretch" VerticalAlignment="Top" Grid.Row="0" x:Name="mainMenu" Grid.ColumnSpan="2" Style="{StaticResource CommonMenu}">*

*<MenuItem x:Name="users" Header="Пользователи" Command="{Binding SectionChoosedCommand}" CommandParameter="users">*

*<MenuItem x:Name="users\_general\_info" Header="Общие сведения" Command="{Binding SectionChoosedCommand}" CommandParameter="users\_general\_info"></MenuItem>*

*<MenuItem x:Name="users\_avg\_cost" Header="Средние затраты пользователей" Command="{Binding SectionChoosedCommand}" CommandParameter="users\_avg\_cost"></MenuItem>*

*<MenuItem x:Name="users\_deffered\_products" Header="Список товаров в корзине" Command="{Binding SectionChoosedCommand}" CommandParameter="users\_deffered\_products"></MenuItem>*

*</MenuItem>*

*<MenuItem x:Name="orders" Header="Доставки" CommandParameter="orders">*

*<MenuItem x:Name="order\_list" Header="Список доставок" Command="{Binding SectionChoosedCommand}" CommandParameter="order\_list"></MenuItem>*

*<MenuItem x:Name="orders\_ready\_to\_receive" Header="Доставки, готовые к получению" Command="{Binding SectionChoosedCommand}" CommandParameter="orders\_ready\_to\_receive"></MenuItem>*

*<MenuItem x:Name="order\_history" Header="История заказов" Command="{Binding SectionChoosedCommand}" CommandParameter="order\_history"></MenuItem>*

*</MenuItem>*

*<MenuItem x:Name="storages" Header="Склады" CommandParameter="storages">*

*<MenuItem x:Name="storages\_general\_info" Header="Общие сведения" Command="{Binding SectionChoosedCommand}" CommandParameter="storages\_general\_info"></MenuItem>*

*<MenuItem x:Name="storages\_receipts" Header="Поступления на склады" Command="{Binding SectionChoosedCommand}" CommandParameter="storages\_receipts"></MenuItem>*

*<MenuItem x:Name="storages\_worker\_shifts" Header="Работа сотрудников на складе" Command="{Binding SectionChoosedCommand}" CommandParameter="storages\_worker\_shifts"></MenuItem>*

*<MenuItem x:Name="storages\_product\_amount" Header="Товары на складах" Command="{Binding SectionChoosedCommand}" CommandParameter="storages\_product\_amount"></MenuItem>*

*</MenuItem>*

*<MenuItem x:Name="products" Header="Товары" CommandParameter="products">*

*<MenuItem x:Name="products\_general\_info" Header="Общие сведения" Command="{Binding SectionChoosedCommand}" CommandParameter="products\_general\_info"></MenuItem>*

*<MenuItem x:Name="products\_reviews" Header="Отзывы к товару" Command="{Binding SectionChoosedCommand}" CommandParameter="products\_reviews"></MenuItem>*

*<MenuItem x:Name="products\_categories" Header="Категории товаров" Command="{Binding SectionChoosedCommand}" CommandParameter="products\_categories"></MenuItem>*

*<MenuItem x:Name="products\_price\_history" Header="История цен" Command="{Binding SectionChoosedCommand}" CommandParameter="products\_price\_history"></MenuItem>*

*</MenuItem>*

*<MenuItem x:Name="suppliers" Header="Поставщики" CommandParameter="suppliers">*

*<MenuItem x:Name="suppliers\_general\_info" Header="Общие сведения" Command="{Binding SectionChoosedCommand}" CommandParameter="suppliers\_general\_info"></MenuItem>*

*<MenuItem x:Name="suppliers\_profit" Header="Прибыль" Command="{Binding SectionChoosedCommand}" CommandParameter="suppliers\_profit"></MenuItem>*

*</MenuItem>*

*<MenuItem x:Name="workers" Header="Сотрудники" CommandParameter="workers">*

*<MenuItem x:Name="workers\_list" Header="Список сотрудников" Command="{Binding SectionChoosedCommand}" CommandParameter="workers\_list"></MenuItem>*

*</MenuItem>*

*</Menu>*

*<TabControl x:Name="mainTabControl"*

*Grid.Row="1" HorizontalAlignment="Stretch"*

*VerticalAlignment="Stretch"*

*Margin="10" >*

*</TabControl>*

*</Grid>*

*</Window>*

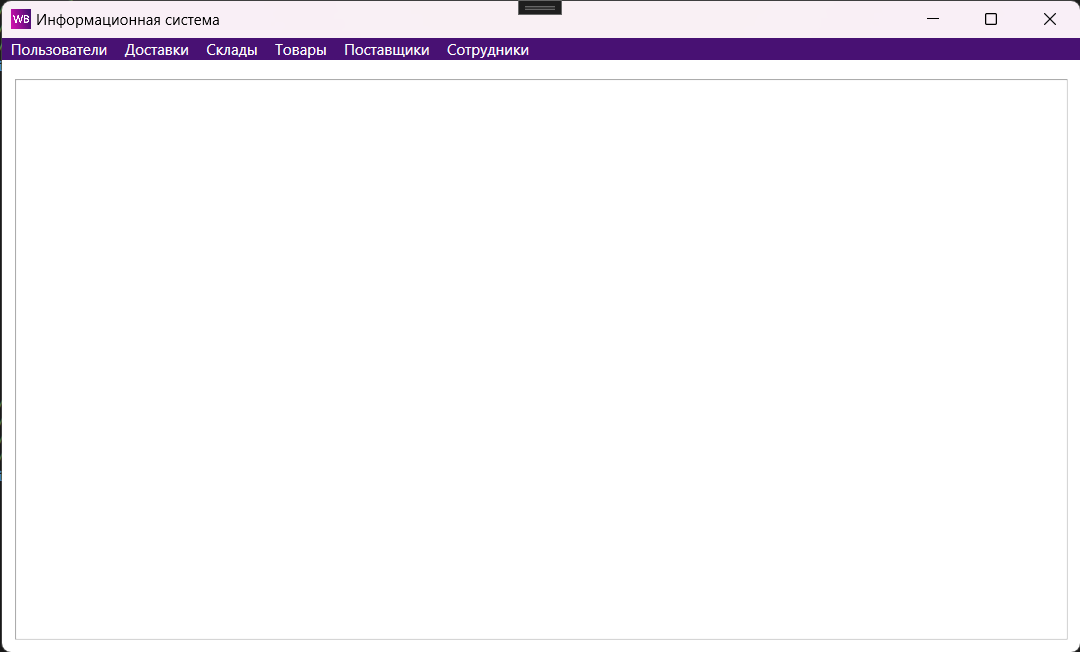


Рисунок 7 - Главное окно приложения.

После того, как главное окно создано, был разработан первый раздел – «Пользователи» - «Общие сведения». В данном разделе согласно требованиям должна быть таблица с такими данными, как id, ФИО, номер телефона, дата рождения, email, пол, страна пользователя; кнопки «Вставить», «Изменить», «Удалить», «Просмотр», «В PDF». Раздел изображён в соответствии с рисунком ХХ. Код раздела «Пользователи» - «Общие сведения»:

*<local:SectionWidget x:Class="WpfApp1.Views.UserGeneralInfoSectionWidget"*

*xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"*

*xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"*

*xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"*

*xmlns:d="http://schemas.microsoft.com/expression/blend/2008"*

*xmlns:local="clr-namespace:WpfApp1.Views"*

*mc:Ignorable="d" HorizontalAlignment="Stretch" VerticalAlignment="Stretch">*

*<Grid Background="White">*

*<Grid.RowDefinitions>*

*<RowDefinition Height="88\*" />*

*<RowDefinition Height="auto" />*

*</Grid.RowDefinitions>*

*<DataGrid x:Name="dataGrid" Grid.Row="0" AutoGenerateColumns="False" IsReadOnly="True">*

*<DataGrid.Resources>*

*<Style TargetType="DataGridColumnHeader">*

*<EventSetter Event="MouseDoubleClick" Handler="DataGridColumnHeader\_DoubleClick" />*

*</Style>*

*</DataGrid.Resources>*

*<DataGrid.Columns>*

*<DataGridTextColumn Header="Id" Binding="{Binding Path=Id}" Width="40" />*

*<DataGridTextColumn Header="Фамилия" Binding="{Binding Path=Lastname}" Width="90" />*

*<DataGridTextColumn Header="Имя" Binding="{Binding Path=Firstname}" Width="90" />*

*<DataGridTextColumn Header="Отчество" Binding="{Binding Path=Patronymic}" Width="90" />*

*<DataGridTextColumn Header="Номер телефона" Binding="{Binding Path=PhoneNumber}" Width="110" />*

*<DataGridTextColumn Header="Дата рождения" Binding="{Binding Path=Birthday}" Width="110" />*

*<DataGridTextColumn Header="Email" Binding="{Binding Path=Email}" Width="90" />*

*<DataGridTextColumn Header="Пол" Binding="{Binding Path=Gender}" Width="80" />*

*<DataGridTextColumn Header="Страна" Binding="{Binding Path=Country.Title}" Width="90" />*

*</DataGrid.Columns>*

*</DataGrid>*

*<DockPanel Grid.Row="1" Margin="10" Height="32">*

*<Button x:Name="btnPDF" Width="90" Template="{StaticResource commonButton}" Command="{Binding PDFCommand}">В PDF</Button>*

*<StackPanel DockPanel.Dock="Right" Orientation="Horizontal" FlowDirection="RightToLeft">*

*<Button x:Name="btnClose" Width="90" Margin="10 0" Template="{StaticResource commonButton}" Command="{Binding CloseCommand}">Закрыть</Button>*

*<Button x:Name="btnRead" Width="90" Margin="10 0" Template="{StaticResource commonButton}" Command="{Binding ReadCommand}">Просмотр</Button>*

*<Button x:Name="btnDelete" Width="90" Margin="10 0" Template="{StaticResource commonButton}" Command="{Binding DeleteCommand}">Удалить</Button>*

*<Button x:Name="btnUpdate" Width="90" Margin="10 0" Template="{StaticResource commonButton}" Command="{Binding UpdateCommand}">Изменить</Button>*

*<Button x:Name="btnInsert" Width="90" Margin="10 0" Template="{StaticResource commonButton}" Command="{Binding InsertCommand}">Вставить</Button>*

*</StackPanel>*

*</DockPanel>*

*</Grid>*

*</local:SectionWidget >*

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 8 - Раздел "Пользователи" - "Общие сведения".

Затем было создано окно работы с записью раздела «Пользователи» - «Общие сведения». Окно работы с записью раздела должно содержать id, ФИО, номер телефона в виде текстовых полей, дату рождения в виде элемента управления ввода даты, email, пол в виде переключателей, страну в виде выпадающего списка. Также должны быть кнопки «Закрыть», «Сохранить» или «Изменить», в зависимости от режима работы с данными. Окно работы с записью в соответствие с рисунком ХХ. Код окна работы с записью:

*<local:ItemForm x:Class="WpfApp1.Views.Users.GeneralInfo.UserGeneralInfoItem"*

*xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"*

*xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"*

*xmlns:d="http://schemas.microsoft.com/expression/blend/2008"*

*xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"*

*xmlns:local="clr-namespace:WpfApp1.Views"*

*mc:Ignorable="d" ResizeMode="NoResize"*

*Icon="\Resources\wb.ico"*

*Height="419" Width="344">*

*<Grid>*

*<Grid.RowDefinitions>*

*<RowDefinition />*

*<RowDefinition Height="50" />*

*</Grid.RowDefinitions>*

*<Grid Grid.Row="0" Grid.ColumnSpan="3">*

*<Grid.ColumnDefinitions>*

*<ColumnDefinition Width="120"/>*

*<ColumnDefinition Width="\*"/>*

*</Grid.ColumnDefinitions>*

*<Grid.RowDefinitions>*

*<RowDefinition/>*

*<RowDefinition/>*

*<RowDefinition/>*

*<RowDefinition/>*

*<RowDefinition/>*

*<RowDefinition/>*

*<RowDefinition/>*

*<RowDefinition/>*

*<RowDefinition/>*

*</Grid.RowDefinitions>*

*<Label Grid.Column="0" Grid.Row="0" Content="Id" VerticalAlignment="Center"/>*

*<TextBox x:Name="tbId"*

*Width="200" Grid.Column="1"*

*Grid.Row="0" HorizontalAlignment="Left"*

*Margin="0 5" Text="{Binding CurrentItem.Id, Mode=OneWay}"*

*IsReadOnly="True" Style="{StaticResource CommonTextBox}">*

*</TextBox>*

*<Label Grid.Column="0" Grid.Row="1" Content="Фамилия" VerticalAlignment="Center"/>*

*<TextBox x:Name="tbLastname"*

*Grid.Column="1" Grid.Row="1"*

*Margin="0 5" Width="200"*

*HorizontalAlignment="Left"*

*Text="{Binding CurrentItem.Lastname, UpdateSourceTrigger=PropertyChanged}"*

*Style="{StaticResource CommonTextBox}"/>*

*<Label Grid.Column="0" Grid.Row="2" Content="Имя" VerticalAlignment="Center"/>*

*<TextBox x:Name="tbFirstname"*

*Grid.Column="1" Grid.Row="2"*

*Margin="0 5" Width="200"*

*HorizontalAlignment="Left"*

*Style="{StaticResource CommonTextBox}" Text="{Binding CurrentItem.Firstname, UpdateSourceTrigger=PropertyChanged}" />*

*<Label Grid.Column="0" Grid.Row="3" Content="Отчество" VerticalAlignment="Center"/>*

*<TextBox x:Name="tbPatronymic"*

*Grid.Column="1" Grid.Row="3"*

*Margin="0 5" Width="200"*

*HorizontalAlignment="Left" Style="{StaticResource CommonTextBox}"*

*Text="{Binding CurrentItem.Patronymic, UpdateSourceTrigger=PropertyChanged}" />*

*<Label Grid.Column="0" Grid.Row="4" Content="Номер телефона" VerticalAlignment="Center"/>*

*<TextBox x:Name="tbPhoneNumber"*

*Grid.Column="1" Grid.Row="4"*

*Margin="0 5" Width="200"*

*HorizontalAlignment="Left"*

*Style="{StaticResource CommonTextBox}"*

*Text="{Binding CurrentItem.PhoneNumber, UpdateSourceTrigger=PropertyChanged}" />*

*<Label Grid.Column="0" Grid.Row="5" Content="Дата рождения" VerticalAlignment="Center"/>*

*<DatePicker x:Name="calendarBirthday"*

*Grid.Column="1" Grid.Row="5"*

*SelectedDate="{Binding CurrentItem.Birthday}"*

*Width="200" Margin="0 5"*

*HorizontalAlignment="Left"/>*

*<Label Grid.Column="0" Grid.Row="6" Content="Email" VerticalAlignment="Center"/>*

*<TextBox x:Name="tbEmail"*

*Grid.Column="1" Grid.Row="6"*

*Margin="0 5" Width="200"*

*HorizontalAlignment="Left"*

*Style="{StaticResource CommonTextBox}"*

*Text="{Binding CurrentItem.Email, UpdateSourceTrigger=PropertyChanged}" />*

*<Label Grid.Column="0" Grid.Row="7" Content="Пол" VerticalAlignment="Center"/>*

*<StackPanel Grid.Row="7" Grid.Column="1">*

*<RadioButton x:Name="rbMale" GroupName="Gender" Content="Мужчина" IsChecked="{Binding CurrentItem.IsMale, Mode=OneWay}"/>*

*<RadioButton x:Name="rbFemale" GroupName="Gender" Content="Женщина" IsChecked="{Binding CurrentItem.IsFemale, Mode=OneWay}"/>*

*</StackPanel>*

*<Label Grid.Column="0" Grid.Row="8" Content="Страна" VerticalAlignment="Center"/>*

*<ComboBox x:Name="cbCountry"*

*Grid.Column="1" Grid.Row="8"*

*Width="200" ItemsSource="{Binding Countries}"*

*HorizontalAlignment="Left"*

*SelectedValue="{Binding CurrentItem.Country}"*

*Style="{StaticResource CommonComboBox}">*

*<ComboBox.ItemTemplate>*

*<DataTemplate>*

*<TextBlock Text="{Binding Title}"/>*

*</DataTemplate>*

*</ComboBox.ItemTemplate>*

*</ComboBox>*

*</Grid>*

*<DockPanel Grid.Row="1" Margin="10" Grid.ColumnSpan="3">*

*<Button x:Name="btnDataAction" Margin="10 0" Width="90" HorizontalAlignment="Left" Template="{StaticResource commonButton}" Command="{Binding SaveCommand}">Сохранить</Button>*

*<Button x:Name="btnClose" Margin="10 0" Width="90" HorizontalAlignment="Right" Template="{StaticResource commonButton}" Click="btnClose\_Click">Закрыть</Button>*

*</DockPanel>*

*</Grid>*

*</local:ItemForm>*

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 9 - Окно работы с записью раздела "Пользователи" - "Общие сведения".

В приложении запланированы разделы, в которых должна быть возможность работать с изображениями. Следующим разделом было решено сделать «Товары» - «Общие сведения». В окне работы с записью раздела должны быть id товара, его название, описание, цена в виде текстовых полей; id и название поставщика в виде выпадающего списка; id и название категории в виде выпадающего списка; процент, который получает поставщик с продажи продукта в виде текстового поля. Должны быть кнопки «Закрыть», «Сохранить» или «Изменить», в зависимости от режима работы с данными. Также в окне работы с записью раздела должен быть список изображений товара, кнопка «Добавить». Окно работы с записью в соответствие с рисунком ХХ. Код окна работы с записью:

*<local:ItemWithImages x:Class="WpfApp1.Views.Products.GeneralInfo.ProductsGeneralInfoItemWithImages"*

*xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"*

*xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"*

*xmlns:d="http://schemas.microsoft.com/expression/blend/2008"*

*xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"*

*xmlns:local="clr-namespace:WpfApp1.Views"*

*mc:Ignorable="d" ResizeMode="NoResize"*

*Icon="\Resources\wb.ico"*

*Title="ItemForm" Height="577" Width="450">*

*<Grid>*

*<Grid.RowDefinitions>*

*<RowDefinition />*

*<RowDefinition />*

*<RowDefinition Height="41" />*

*</Grid.RowDefinitions>*

*<Grid Grid.Row="0">*

*<Grid.ColumnDefinitions>*

*<ColumnDefinition Width="140"/>*

*<ColumnDefinition Width="\*"/>*

*</Grid.ColumnDefinitions>*

*<Grid.RowDefinitions>*

*<RowDefinition />*

*<RowDefinition />*

*<RowDefinition />*

*<RowDefinition />*

*<RowDefinition />*

*<RowDefinition />*

*<RowDefinition />*

*</Grid.RowDefinitions>*

*<Label Grid.Column="0" Grid.Row="0" Content="Id" VerticalAlignment="Center"/>*

*<TextBox x:Name="tbId" Width="280"*

*Grid.Column="1"Grid.Row="0"*

*HorizontalAlignment="Left"*

*Margin="0 5" Text="{Binding CurrentItem.Id, Mode=OneWay}"*

*IsReadOnly="True" Style="{StaticResource CommonTextBox}" />*

*<Label Grid.Column="0" Grid.Row="1" Content="Название" VerticalAlignment="Center"/>*

*<TextBox x:Name="tbTitle"*

*Width="280" Grid.Column="1"*

*Grid.Row="1" HorizontalAlignment="Left"*

*Margin="0 5" Text="{Binding CurrentItem.Title, UpdateSourceTrigger=PropertyChanged}" Style="{StaticResource CommonTextBox}" />*

*<Label Grid.Column="0" Grid.Row="2" Content="Описание" VerticalAlignment="Center"/>*

*<TextBox x:Name="tbDescription" Width="280"*

*Grid.Column="1" Grid.Row="2"*

*HorizontalAlignment="Left" Margin="0 5"*

*Text="{Binding CurrentItem.Description, UpdateSourceTrigger=PropertyChanged}"*

*Style="{StaticResource CommonTextBox}" />*

*<Label Grid.Column="0" Grid.Row="3" Content="Цена" VerticalAlignment="Center"/>*

*<TextBox x:Name="tbPrice" Width="280"*

*Grid.Column="1" Grid.Row="3"*

*HorizontalAlignment="Left" Margin="0 5"*

*Text="{Binding CurrentItem.Price, UpdateSourceTrigger=PropertyChanged}"*

*Style="{StaticResource CommonTextBox}" />*

*<Label Grid.Column="0" Grid.Row="4" Content="Поставщик" VerticalAlignment="Center"/>*

*<ComboBox x:Name="cbSupplier" Grid.Row="4"*

*Grid.Column="1" Style="{StaticResource CommonComboBox}"*

*Width="280" ItemsSource="{Binding Suppliers}"*

*HorizontalAlignment="Left" SelectedValue="{Binding CurrentItem.Supplier}">*

*<ComboBox.ItemTemplate>*

*<DataTemplate>*

*<StackPanel Orientation="Horizontal">*

*<Label Content="{Binding Id}"/>*

*<Label Content="{Binding Title}"/>*

*</StackPanel>*

*</DataTemplate>*

*</ComboBox.ItemTemplate>*

*</ComboBox>*

*<Label Grid.Column="0" Grid.Row="5" Content="Категория" VerticalAlignment="Center" />*

*<ComboBox x:Name="cbCategory"*

*Grid.Row="5" Grid.Column="1"*

*Style="{StaticResource CommonComboBox}"*

*Width="280" ItemsSource="{Binding Categories}"*

*HorizontalAlignment="Left" SelectedValue="{Binding CurrentItem.Category}">*

*<ComboBox.ItemTemplate>*

*<DataTemplate>*

*<StackPanel Orientation="Horizontal">*

*<Label Content="{Binding Id}" />*

*<Label Content="{Binding Title}" />*

*</StackPanel>*

*</DataTemplate>*

*</ComboBox.ItemTemplate>*

*</ComboBox>*

*<TextBlock Margin="5 0" Grid.Column="0" Grid.Row="7" Text="Процент поставщика с продажи" TextWrapping="Wrap" VerticalAlignment="Center"/>*

*<TextBox x:Name="tbPercent"*

*Width="280" Grid.Column="1"*

*Grid.Row="7" HorizontalAlignment="Left"*

*Margin="0 5" Text="{Binding CurrentItem.SupplierPercent, UpdateSourceTrigger=PropertyChanged}"*

*Style="{StaticResource CommonTextBox}" />*

*</Grid>*

*<Grid Grid.Row="1" Margin="10 10 10 5">*

*<Grid.RowDefinitions>*

*<RowDefinition Height="\*"/>*

*<RowDefinition Height="41"/>*

*</Grid.RowDefinitions>*

*<ListBox x:Name="lbImages" Grid.Row="0" PreviewMouseDoubleClick="lbImages\_PreviewMouseDoubleClick">*

*<ListBox.ItemTemplate>*

*<DataTemplate>*

*<Image Width="100" Height="100" Source="{Binding ProductImage1}"/>*

*</DataTemplate>*

*</ListBox.ItemTemplate>*

*</ListBox>*

*<Button x:Name="btnAddImage" HorizontalAlignment="Right" Margin="10 5" Width="110" Grid.Row="1" Template="{StaticResource commonButton}" Command="{Binding InsertImageCommand}">Добавить</Button>*

*</Grid>*

*<DockPanel Grid.Row="2">*

*<Button x:Name="btnDataAction" Margin="20 5" Width="110" HorizontalAlignment="Left" Template="{StaticResource commonButton}" Command="{Binding SaveCommand}">Сохранить</Button>*

*<Button x:Name="btnClose" Margin="20 5" Width="110" HorizontalAlignment="Right" Template="{StaticResource commonButton}" Click="btnClose\_Click">Закрыть</Button>*

*</DockPanel>*

*</Grid>*

*</local:ItemWithImages>*

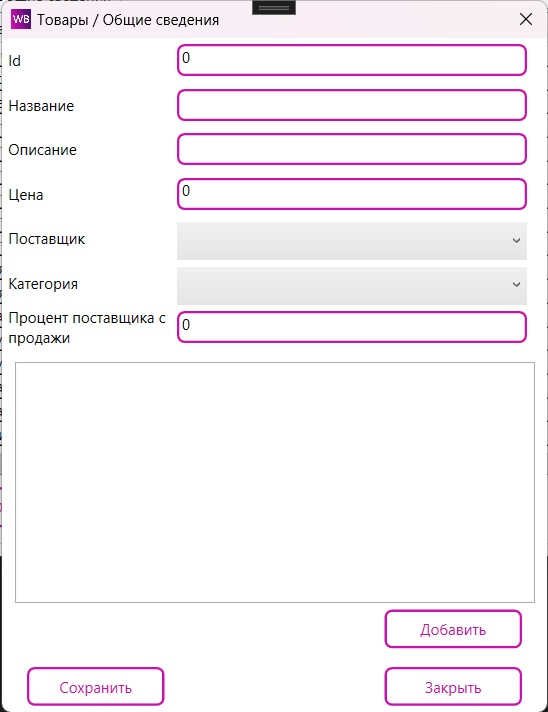


Рисунок 10 - Окно работы с записью раздела "Товары" - "Общие сведения".

Остальные разделы были сделаны аналогичным образом. В некоторых разделах убраны кнопки «Добавить», «Удалить», «Изменить». В разделе «Доставки» - «История заказов» есть подраздел с таблицей сотрудников, причастных к изменению статуса заказа. Окно работы с записью раздела «Доставки» - «История заказов» в соответствие с рисунком ХХ.

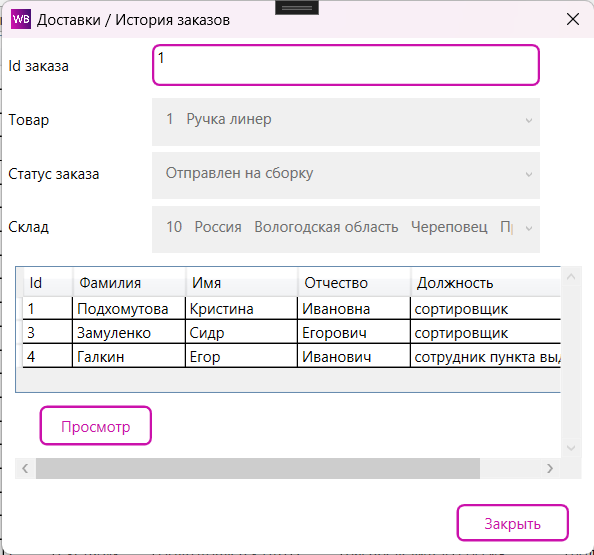


Рисунок 11 - Окно работы с записью раздела "Доставки" - "История заказа".

2.2 Разработка программного функционала

После разработки графического интерфейса разделов было необходимо обеспечить саму работу с данными в них. В большинстве разделов работа с данными одинакова. При нажатии на кнопки «Вставить», «Изменить», «Просмотр» должно появляться окно, в котором можно соответственно добавлять новую запись, изменять или просматривать текущую. При нажатии на кнопку «Удалить» запись должна удаляться. Чтобы не прописывать одинаковую логику работы с данными для каждого раздела, было решено вынести её в абстрактный класс SectionWidgetViewModel. Для обработки исключительных ситуации необходимо подключить пространство имён System строкой «using System;». В разделе есть таблица с данными, которые могут изменяться. Чтобы воспользоваться механизмом привязки (binding) в WPF и сделать так, чтобы при изменении списка записей раздела, данные в таблице обновлялись автоматически, необходимо использовать класс ObservableCollection, который находится в пространстве имён System.Collections.ObjectModel. Чтобы создавать диалоговые окна с предупреждениями об ошибках, необходимо использовать пространство имён System.Windows. Для взаимодействия с заголовком таблицы данных раздела нужно подключить пространство имён System.Windows.Controls.Primitives. Генерация PDF-документов и реализация авторизации в приложении вынесены в отдельные классы, которые располагаются в пространстве имён WpfApp1.Services. Элементы графического интерфейса находятся в пространстве имён WpfApp1.Views. Конструктор класса имеет следующий код:

*protected SectionWidgetViewModel(SectionWidget sectionWidget)*

*{*

*SectionWidget = sectionWidget;*

*\_accessService = App.AccessService;*

*\_pdfGenarator = App.PDFGenerator;*

*CollapseButtonsWithoutRights();*

*FilterService = new FilterService(this);*

*MakeCurrentItemEmpty();*

*UpdateSectionData();*

*}.*

Конструктор принимает в качестве параметра ссылку на представление раздела sectionWidget и запоминает её значение. Затем он берёт ссылки на объекты классов генерации PDF-файлов и авторизации из класса App, в котором они создаются. Далее конструктор вызывает метод, скрывающий кнопки в разделе, на которых у пользователя нет прав. Создаётся новый экземпляр класса, фильтрующего записи в таблице раздела. Вызывается метод, создающий пустой экземпляр записи раздела и заполняется таблица с данными раздела.

Методы, вызываемые при нажатии на кнопки «Вставить», «Изменить», «Просмотр», «Удалить», «В PDF», «Закрыть» в разделе, а также на кнопку «Сохранить» в окно работы с данными раздела реализованы в виде команд:

*/// <summary>*

*/// Команда вызова окна для вставки записи в базу данных.*

*/// </summary>*

*protected RelayCommand? \_insertCommand;*

*/// <summary>*

*/// Команда вызова окна для удаления записи из базы данных.*

*/// </summary>*

*protected RelayCommand? \_deleteCommand;*

*/// <summary>*

*/// Команда вызова окна для изменения записи в базе данных.*

*/// </summary>*

*protected RelayCommand? \_updateCommand;*

*/// <summary>*

*/// Команда вызова окна для просмотра записи.*

*/// </summary>*

*protected RelayCommand? \_readCommand;*

*/// <summary>*

*/// Команда закрытия вкладки SectionWidget в главном окне.*

*/// </summary>*

*protected RelayCommand? \_closeCommand;*

*/// <summary>*

*/// Команда сохранения изменений из окна работы с записью в базу данных.*

*/// </summary>*

*protected RelayCommand? \_saveCommand;*

*/// <summary>*

*/// Команда создания PDF-документа на основе данных в разделе.*

*/// </summary>*

*protected RelayCommand? \_pdfCommand;*

*/// <summary>*

*/// Команда вызова окна для вставки записи в базу данных.*

*/// </summary>*

*public RelayCommand InsertCommand {*

*get {*

*return \_insertCommand ??*

*(\_insertCommand = new RelayCommand((object obj) => {*

*TryInsert();*

*},*

*(obj) => \_accessService.HasWorkerRightToInsert(SectionWidget.Section.SectionKey)));*

*}*

*}*

*/// <summary>*

*/// Команда вызова окна для удаления записи из базы данных.*

*/// </summary>*

*public RelayCommand DeleteCommand {*

*get {*

*return \_deleteCommand ??*

*(\_deleteCommand = new RelayCommand((object obj) => {*

*TryDelete();*

*},*

*(obj) => \_accessService.HasWorkerRightToDelete(SectionWidget.Section.SectionKey)));*

*}*

*}*

*/// <summary>*

*/// Команда вызова окна для изменения записи в базе данных.*

*/// </summary>*

*public RelayCommand UpdateCommand {*

*get {*

*return \_updateCommand ??*

*(\_updateCommand = new RelayCommand((object obj) => {*

*TryUpdate();*

*},*

*(obj) => \_accessService.HasWorkerRightToUpdate(SectionWidget.Section.SectionKey)));*

*}*

*}*

*/// <summary>*

*/// Команда вызова окна для просмотра записи.*

*/// </summary>*

*public RelayCommand ReadCommand {*

*get {*

*return \_readCommand ??*

*(\_readCommand = new RelayCommand((object obj) => {*

*Read();*

*},*

*(obj) => \_accessService.HasWorkerRightToRead(SectionWidget.Section.SectionKey)));*

*}*

*}*

*/// <summary>*

*/// Команда закрытия вкладки SectionWidget в главном окне.*

*/// </summary>*

*public RelayCommand CloseCommand {*

*get {*

*return \_closeCommand ??*

*(\_closeCommand = new RelayCommand((object obj) => {*

*Close();*

*}));*

*}*

*}*

*/// <summary>*

*/// Команда сохранения изменений из окна работы с записью в базу данных.*

*/// </summary>*

*public RelayCommand SaveCommand {*

*get {*

*return \_saveCommand ??*

*(\_saveCommand = new RelayCommand((object obj) =>*

*{*

*Save();*

*},*

*(obj) => \_itemFormMode == ItemFormMode.Insert || \_itemFormMode == ItemFormMode.Update));*

*}*

*}*

*/// <summary>*

*/// Команда создания PDF-документа на основе данных в разделе.*

*/// </summary>*

*public RelayCommand PDFCommand {*

*get {*

*return \_pdfCommand ??*

*(\_pdfCommand = new RelayCommand((object obj) => {*

*ToPDF();*

*},*

*(obj) => \_accessService.HasWorkerRightToPDF(SectionWidget.Section.SectionKey)));*

*}*

*}*

Данные команды в свою очередь вызывают методы с соответствующей реализацией. Метод открытия окна на вставку записей создаёт новую пустую запись, устанавливает режим вставки записи, создаёт и вызывает форму работы с записью:

*protected virtual void TryInsert() {*

*MakeCurrentItemEmpty();*

*\_itemFormMode = ItemFormMode.Insert;*

*CreateNewItemForm();*

*ItemForm.Title = SectionTitle;*

*ItemForm.Mode = \_itemFormMode;*

*ItemForm.ShowDialog();*

*}*

Остальные методы открытия окна работы с записью имеют схожий принцип, имеют небольшие отличия.

Метод, сохраняющий запись в базу данных, запоминает состояние записи в контексте и пытается выполнить вставку в базу данных, если возникает ошибка, то он возвращает запись в контексте в исходное состояние и вывод сообщение. Здесь используется абстрактный метод AddCurrentItem, который добавляет в контекст новую запись. Код метода вставки записи в БД:

*protected virtual void Insert() {*

*var entry = App.Context.Entry(CurrentItem);*

*try {*

*AddCurrentItem();*

*App.Context.SaveChanges();*

*UpdateItems();*

*} catch (Exception ex) {*

*entry.Reload();*

*MessageBox.Show("Добавление записи завершилось ошибкой");*

*}*

*ItemForm.Close();*

*}*

Методы обновления и удаления данных имеют аналогичные принципы работы:

*protected virtual void Update() {*

*try {*

*CurrentItemFromContext.Copy(CurrentItem);*

*App.Context.SaveChanges();*

*UpdateItems();*

*} catch(Exception ex) {*

*MessageBox.Show("Изменение записи завершилось ошибкой");*

*}*

*ItemForm.Close();*

*}*

*protected virtual void Delete() {*

*var entry = App.Context.Entry(CurrentItemFromContext);*

*try {*

*DeleteCurrentItem();*

*App.Context.SaveChanges();*

*UpdateItems();*

*} catch (Exception ex) {*

*entry.Reload();*

*MessageBox.Show("Вы не можете удалить эту запись, так как она используется в другом разделе");*

*} finally {*

*MakeCurrentItemEmpty();*

*}*

*}*

Благодаря вынесению большей части логики в абстрактный класс, при разработке функциональной части разделов можно будет меньше писать кода. Рассмотри реализацию UsersGeneralInfoViewModel. В данном классе переопределяется свойство ItemForm, указывающее на окно работы с записью; метод, создающий пустую запись MakeCurrentItemEmpty; метод создания нового окна работы с записью CreateNewItemForm. Также переопределяются методы AddCurrentItem и DeleteCurrentItem, создающий новую и удаляющий существующую записи соответственно; метод заполнения сущности данными FillItem и проверки ошибок GetErrors. Также в классе есть список стран Countries, необходимый для заполнения выпадающего списка в окне работы с записью раздела. Код класса UsersGeneralInfoViewModel:

*using System;*

*using System.Collections.Generic;*

*using System.Linq;*

*using System.Text;*

*using WpfApp1.Views;*

*using WpfApp1.Views.Users.GeneralInfo;*

*using WpfApp1.Models;*

*using WpfApp1.Services;*

*using ValidationLib;*

*namespace WpfApp1.ViewModels.Users*

*{*

*/// <summary>*

*/// Модель представления для раздела "Пользователи / Общие сведения".*

*/// </summary>*

*internal class UsersGeneralInfoViewModel : SectionWidgetViewModel*

*{*

*/// <summary>*

*/// Ссылка на окно работы с записью раздела.*

*/// </summary>*

*private UserGeneralInfoItem \_itemForm;*

*public override ItemForm ItemForm*

*{*

*get => \_itemForm as object as ItemForm;*

*set => \_itemForm = value as UserGeneralInfoItem;*

*}*

*/// <summary>*

*/// Коллекция стран, используется для заполнения выпадающего списка в окне работы с записью раздела.*

*/// </summary>*

*public List<Countries> Countries { get; set; }*

*/// <summary>*

*/// Конструктор класса UsersGeneralInfoViewModel, в качестве параметра принимает ссылку на представление раздела.*

*/// </summary>*

*/// <param name="sectionWidget">Представление раздела.</param>*

*public UsersGeneralInfoViewModel(SectionWidget sectionWidget) : base(sectionWidget) {}*

*protected override void MakeCurrentItemEmpty() {*

*CurrentItem = new Models.Users();*

*}*

*protected override void CreateNewItemForm() {*

*Countries = App.Context.Countries.ToList();*

*\_itemForm = new UserGeneralInfoItem(this);*

*}*

*protected override void AddCurrentItem() {*

*App.Context.Users.Add(CurrentItem);*

*}*

*protected override void DeleteCurrentItem() {*

*App.Context.Users.Remove(CurrentItemFromContext);*

*}*

*public override void UpdateSectionData() {*

*SectionData = UserService.GetUserGeneralInfo();*

*}*

*protected override void FillItem() {*

*CurrentItem.IsMale = \_itemForm.rbMale.IsChecked ?? false;*

*}*

*protected override string GetErrors() {*

*StringBuilder errorBuilder = new StringBuilder();*

*if (!StringValidator.IsValid(CurrentItem.Lastname))*

*{*

*errorBuilder.AppendLine("Поле \"Фамилия\" обязательно для заполнения, максимальная длина - 255 символов;");*

*}*

*if (!StringValidator.IsValid(CurrentItem.Firstname))*

*{*

*errorBuilder.AppendLine("Поле \"Имя\" обязательно для заполнения, максимальная длина - 255 символов;");*

*}*

*if (CurrentItem.Patronymic.Length > 255)*

*{*

*errorBuilder.AppendLine("Максимальная длина поля \"Отчество\" - 255 символов;");*

*}*

*if (!PhoneNumberValidator.IsValid(CurrentItem.PhoneNumber))*

*{*

*errorBuilder.AppendLine("Неверное значение поля \"Номер телефона\";");*

*}*

*if (!EmailValidatior.IsValid(CurrentItem.Email))*

*{*

*errorBuilder.AppendLine("Неверное значение поля \"Email\";");*

*}*

*if (CurrentItem.Country == null)*

*{*

*errorBuilder.AppendLine("Свойство \"Страна\" обязательно для заполнения;");*

*}*

*if (CurrentItem.Birthday == null || CurrentItem.Birthday < new DateTime(1900, 1, 1) || CurrentItem.Birthday > new DateTime(3000, 12, 31))*

*{*

*errorBuilder.AppendLine("Свойство \"Дата рождения\" обязательно для заполнения, допустимые значения от 1900.01.01 до 3000.12.31;");*

*}*

*return errorBuilder.ToString();*

*}*

*}*

*}*

С помощью хорошо разработанной архитектуры не составило труда написать логику для других разделов. Была добавлена фильтрация, работа с изображениями, выгрузка отчётов в PDF-документы.

3 ТЕСТИРОВАНИЕ

Лорем ипсум.

4 РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

Лорем ипсум

ЗАКЛЮЧЕНИЕ

Лорем ипсум

Список использованных источников

Лорем ипсум

Приложение 1

Бизнес-требования

Суть проекта:

Разработка приложения, в котором будут возможности работы с данными интернет-магазина Wildberries.

Цели проекта:

* Предоставление ограниченного доступа к данным для сотрудников, участвующих в бизнес-процессах интернет-магазина Wildberries;
* Упрощение поиска нужных сведений за счёт группировки данных в разделы.

Системные характеристики

1. Приложение – настольное.
2. Приложение необходимо разработать с применением .Net 6.0.
3. Целевая операционная система – Windows 10.
4. Для хранения данных следует использовать СУБД Microsoft SQL Server 2022 Express Edition.

Пользовательские требования

1. Пользователи системы:

* администратор информационной системы;
* сортировщик на складе;
* сотрудник пункта выдачи;
* бизнес-аналитик.

1. Запуск и закрытие приложения.
   1. Запуск приложения осуществляется двойным нажатием по исполняемому файлу приложения.
   2. Закрытие приложения осуществляется нажатием на крестик в верхнем правом углу.
2. Вход в систему.
   1. При запуске приложения появляется окно входа в систему (см. ДС-1).
   2. Пароль и логин сотрудникам выдаёт администратор (см. БП-1).
3. Выход из системы.
   1. Выход из системы происходит автоматически при закрытии приложения.
4. Работа с разделами системы.
   1. После удачного входа в систему открывается главное окно приложения (см. ДС-2), содержащее меню групп разделов информационной системы.
5. Работа с данными информационный системы.
   1. Данные в информационной системе должны располагаться в соответствующих им разделах (см. ДС-3).
   2. Пользователи системы должны иметь возможность работать с данными о пользователях, такими как: общая информация о пользователях; данные о товарах в корзине; данные о средних затратах пользователей (см. ДС-7).
   3. Пользователи системы должны иметь возможность работать с данными о доставках, такими как: данные о доставках пользователей; данные о заказах, готовых к получению; данные об изменениях статусов доставки заказа (см. ДС-8).
   4. Пользователи системы должны иметь возможность работать с данными о складах, такими как: общая информация о складах; данные о поступлениях товаров на склады; данные о работе сотрудников на складах; данные о количестве товаров на складах (см. ДС-9).
   5. Пользователи системы должны иметь возможность работать с данными о товарах, такими как: общая информация о товарах; данные об отзывах к доставленным товарам; данные о категориях товаров; данные об истории цен (см. ДС-10).
   6. Пользователи системы должны иметь возможность работать с данными об поставщиках, такими как: общая информация о поставщиках; данные о прибыли поставщиков (см. ДС-11).
   7. Пользователи системы должны иметь возможность работать с данными о сотрудниках интернет-магазина (см. ДС-12).
6. Выгрузка данных в PDF-файл.
   1. Должна быть возможность выгрузки данных в PDF-файлы.
7. Возможности сотрудников в соответствии с рисунком 1.
   1. Администратор информационной системы имеет доступ ко всем разделам в системе (см. ДС-3), а также способен выполнять любые доступные операции над данными.
   2. Бизнес-аналитик имеет доступ ко всем разделам в системе (см. ДС-3), но способен только просматривать данные.
   3. Сортировщик имеет доступ к разделам «Склады» − «Поступления на склады» (см. ДС-9.2), «Склады» − «Товары на складах» (см. ДС-9.4). Данные для сортировщика доступны только для просмотра.
   4. Сотрудник пункта выдачи может просматривать данные в разделе «Доставки» – «Доставки, готовые к получению» (см. ДС-8.2).
   5. Сотрудник пункта выдачи может изменять статус доставки заказа с «готов к получению» на «получен» или «оформлен возврат» в разделе «Доставки» – «Доставки, готовые к получению» (см. ДС-8.2).
   6. Возможность выгрузки данных в PDF файл должна быть доступна только бизнес-аналитику и администратору.



Рисунок 1 – Use-case диаграмма.

Бизнес-правила

1. Доступ к информационной системе.
   1. Пароль и логин сотрудник получает от администратора информационной системы после того, как администратор добавит сотрудника через раздел «Сотрудники» − «Список сотрудников» (см. ДС-12.1).
2. Цветовая гамма приложения должна соответствовать корпоративной цветовой гамме интернет-магазина Wildberries. Допустимыми цветами являются фиолетовый (#cb11ab), индиго (#481173), бледный пурпурно-синий (#c8b8c5), розовый (#e6a0dc), белый, черный.

Детальные спецификации

1. Окно входа в систему.
   1. Окно входа содержит поля ввода логина и пароля, а также кнопку «Войти в систему».
   2. При открытии окна входа должна произойти проверка наличия доступа к базе данных.
      1. Если соединение открыть не получится, то должно появиться окно с сообщением «Не удалось подключиться к базе данных!» и кнопкой «ОК».
      2. При нажатии на кнопку «ОК» приложение закроется.
   3. Если пользователь оставил одно из полей ввода пустым и нажал на кнопку входа, должно появиться предупреждение «Необходимо заполнить все поля!».
   4. Если пользователь заполнил все поля и нажал кнопку входа, но пользователя с таким логином и паролем не найдено, должно появиться окно с надписью «Неверный логин или пароль!» и кнопкой «ОК».
   5. Если логин и пароль верны, то окно входа закрывается и открывается главное окно приложения (см. ДС-2).
2. Главное окно приложения.
   1. Главное окно содержит только меню групп разделов информационной системы (см. ДС-3).
3. Меню групп разделов информационной системы.
   1. Меню представляет собой полосу с группами разделов: «Пользователи», «Доставки», «Склады», «Товары», «Поставщики», «Сотрудники».
   2. Если сотрудник не имеет доступа к группе разделов, то она не должна отображаться на экране (см. О-1.1).
   3. В группу разделов «Пользователи» вложены разделы: «Общие сведения», «Средние затраты пользователей», «Список товаров в корзине» (см. ДС-7).
   4. В группу разделов «Доставки» вложены разделы: «Список доставок», «Доставки, готовые к получению», «История заказов» (см. ДС-8).
   5. В группу разделов «Склады» вложены разделы: «Общие сведения», «Поступления на склады», «Работа сотрудников на складах», «Товары на складах» (см. ДС-9).
   6. В группу разделов «Товары» вложены разделы: «Общие сведения», «Отзывы к товару», «Категории товаров», «История цен» (см. ДС-10).
   7. В группу разделов «Поставщики» вложены разделы: «Общие сведения», «Прибыль» (см. ДС-11).
   8. В группу разделов «Сотрудники» вложен раздел «Список сотрудников» (см. ДС-12).
   9. При выборе раздела в меню, в главном окне должно появится окно раздела (см. ДС-4).
4. Окно раздела.
   1. Окно раздела содержит таблицу с данными раздела.
   2. Также в окне раздела должны быть кнопки «Вставить», «Изменить», «Просмотр», «Удалить», если такие операции с данными доступны в разделе (см. О-1.2).
   3. В окне раздела должна быть также кнопка «Закрыть», нажатие на которую закрывает окно раздела.
   4. В окне раздела должна быть кнопка «В PDF» (см. ПТ-7).
      1. Нажатие на кнопку «В PDF» откроет окно проводника, в котором можно будет указать место, куда будет создан PDF-файл. После чего, в том месте появится PDF-файл, заполненный текущими данными из таблицы окна раздела.
   5. При нажатии на кнопку «Вставить» должно открываться окно элемента с возможностью вставки (см. ДС-5, ДС-5.2).
   6. При нажатии на кнопку «Изменить» должно открываться окно элемента с возможностью изменения данных (см ДС-5, ДС-5.3).
   7. При нажатии на кнопку «Просмотр» должно открываться окно элемента для просмотра данных (см ДС-5, ДС-5.4).
   8. При нажатии на кнопку «Удалить» должно появляться окно с надписью «Вы уверены, что хотите удалить запись?» и кнопками «Да», «Нет». Если будет нажата кнопка «Да», то запись удалится из базы данных.
      1. Если в базе данных на запись есть ссылки в других таблицах или в этой, то запись не должна удаляться. Вместо этого появляется окно с сообщением «Не удалось удалить запись, так как на неё ссылаются другие объекты!».
      2. Если к записи прикреплены изображения, то они должны быть удалены вместе с записью.
5. Окно элемента.
   1. Окно элемента содержит данные записи, размещённые в текстовых полях, выпадающих списках, либо таблицах.
      1. Для работы с данными в таблицах нужны кнопки "Вставить", "Изменить", "Просмотр", "Удалить", подобные тем, что используются в окне раздела (см. ДС-4).
   2. Если окно элемента открыто для вставки, то все поля должны быть пустыми и в окне должна присутствовать кнопка «Сохранить».
      1. При нажатии на кнопку «Сохранить» окно элемента закрывается, а данные в таблице окна раздела обновляются.
   3. Если окно элемента открыто для изменения, то поля и списки должны содержать данные текущей записи, в окне должна быть кнопка «Изменить».
      1. При нажатии на кнопку «Изменить», должно появиться окно с надписью «Вы уверены, что хотите изменить запись?» и кнопками «Да», «Нет».
      2. Если будет нажата кнопка «Да», то окно элемента закроется, данные в таблице окна раздела обновятся.
   4. Если окно элемента открыто для просмотра, то поля и списки должны быть заполнены данными записи, но недоступны для изменения значений.
6. Работа с изображениями в окне элемента.
   1. В некоторых случаях должна быть возможность работы с изображениями, связанных с записью.
   2. В окне элемента после полей и списков с данными должны располагаться изображения, относящиеся к записи.
   3. Должна быть кнопка «Добавить изображение», которая откроет проводник для выбора изображения.
      1. Необходима проверка формата файла (см. О-2.1).
      2. Если формат файла неверный, то должно появиться окно с сообщением «Доступные форматы: JPEG, PNG, JPG, GIF» и кнопкой «ОК».
      3. Если будет попытка прикрепить больше 10 файлов к одной записи (см. О-2.2), должно появиться окно с сообщением «Нельзя прикрепить больше 10 файлов!» и кнопкой «ОК».
      4. Если будет попытка прикрепить файл размером более 10 МБ (см. О-2.3), должно появиться окно с сообщением «Изображение должно быть не больше 10 МБ!» и кнопкой «ОК».
   4. При нажатии на изображение в окне элемента должно возникнуть окно с этим изображением и кнопками «Закрыть» и «Удалить».
      1. Если нажать на кнопку «Закрыть», окно с изображением закроется.
      2. Если нажать на кнопку «Удалить», то это изображение должно быть удалено. Изображения в форме элемента должны обновиться.
      3. Если изображение не было найдено, но в системе оно значится за записью, то должна появиться надпись «Изображение имя\_файла не найдено!».
7. Группа разделов «Пользователи».
   1. Раздел «Общие сведения» должен содержать список пользователей.
      1. Данные в таблице раздела: id, ФИО, номер телефона, дата рождения, email, пол, страна.
      2. В разделе доступны все виды операций с данными.
      3. Данные в окне элемента: id, ФИО, номер телефона в виде текстовых полей, дата рождения в виде элемента управления ввода даты, email в виде текстового поля, пол в виде переключателей, страна в виде выпадающего списка.
   2. Раздел «Средние затраты пользователей».
      1. Данные в таблице раздела: id пользователя, ФИО и его средние затраты за всё время.
      2. Раздел доступен только для просмотра данных.
      3. Данные в окне элемента: id пользователя, ФИО и его средние затраты за всё время.
   3. Раздел «Список товаров в корзине».
      1. Данные в таблице раздела: id записи, id и ФИО пользователя, id продукта в корзине пользователя, его название и цена.
      2. Доступны все операции работы с данными.
      3. Данные в окне элемента: id записи в виде текстового поля, id и ФИО пользователя в виде выпадающего списка; id, название продукта в виде выпадающего списка.
      4. Должна быть возможность просмотра изображений товаров.
8. Группа разделов «Доставки».
   1. Раздел «Список доставок».
      1. Данные в таблице раздела: id заказа; id и ФИО пользователя; id продукта, его название, цена, количество товара; стоимость заказа; id и адрес (страна, субъект, город, улица, дом) точки выдачи заказа; дата формирования заказа и ориентировочная дата выдачи.
      2. В разделе доступны все операции с данными.
      3. Данные в окне элемента: id заказа в виде текстового поля; id и ФИО пользователя в виде выпадающего списка; id, название, цена товара в виде выпадающего списка; количество товара и стоимость заказа, в виде текстовых полей; id и адрес (страна, субъект, город, улица, дом) пункта выдачи заказа в виде выпадающего списка; дата формирования и ориентировочная дата выдачи заказа в виде элементов управления ввода даты.
   2. Раздел «Доставки, готовые к получению».
      1. Отображаемые разделом данные: id, адрес (страна, субъект, город, улица, дом) пункта выдачи; id заказа, id, ФИО и код для получения заказа пользователя, id товара, название товара, цена заказа, количество товара.
      2. Окно элемента содержит: id, адрес склада в виде выпадающего списка; id заказа в виде текстового поля; id и ФИО пользователя в виде выпадающего списка; id и название товара в виде выпадающего списка; цену заказа, количество товара в виде текстовых полей, статус заказа в виде выпадающего списка.
      3. Доступные операции: просмотр и изменение.
         1. Изменять в данном разделе можно только статус заказа, он должен представлять собой выпадающий список с тремя значениями: «готов к получению», «получен», «оформлен возврат».
      4. Должна быть возможность просмотра изображений товаров.
   3. Раздел «История заказов» содержит сведения об изменениях статусов доставки заказа.
      1. Данные в таблице раздела: id заказа; id и название продукта; статус заказа; id склада, в котором находился заказ, тип склада, адрес (страна, субъект, город, улица, дом); дата изменения статуса; id, ФИО, должность сотрудника, причастного к изменению статуса заказа.
      2. Доступны все операции с данными.
      3. Окно элемента должно содержать: id заказа в качестве текстового поля; id и название продукта в виде выпадающего списка; статус заказа в виде выпадающего списка; id, адрес и тип склада в виде выпадающего списка; дата изменения статуса в виде элемента управления ввода даты; список сотрудников, причастных к изменению статуса заказа в виде таблицы, содержащей id, ФИО, должность сотрудника.
      4. Окно элемента для работы с данными из таблицы сотрудников, причастных к изменению статуса должно содержать выпадающий список, содержащий id, ФИО, должность сотрудника.
9. Группа разделов «Склады».
   1. Раздел «Общие сведения».
      1. Данные в таблице раздела: id склада, тип склада, адрес (страна, субъект, город, улица, дом).
      2. Доступны все операции работы с данными.
      3. Окно элемента должно содержать: id склада, адрес (страна, субъект, город, улица, дом) в виде текстовых полей, тип склада в виде выпадающего списка.
   2. Раздел «Поступления на склады» содержит сведения о поступлениях товаров на склады.
      1. Данные в таблице раздела: id записи, id склада, тип склада, адрес (страна, субъект, город, улица, дом), id, название и количество товара, дата поступления.
      2. Доступны все операции работы с данными.
      3. Окно элемента должно содержать: id записи в виде текстового поля; id склада, тип и адрес в виде выпадающего списка; id и название товара в виде выпадающего списка; количество товара в виде текстового поля; дату поступления в виде элемента управления ввода даты.
   3. Раздел «Работа сотрудников на складах».
      1. Данные в таблице раздела: id записи, склада, адрес (страна, субъект, город, улица, дом), дата и время начала смены, дата и время конца смены, id сотрудника, его ФИО, должность.
      2. Доступны все операции работы с данными.
      3. Данные в окне элемента раздела: id записи в виде текстового поля; id, тип и адрес склада в виде выпадающего списка; дата и время начала смены, дата и время конца смены в виде элемента управления ввода даты; id сотрудника, его ФИО и должность в виде выпадающего списка.
   4. Раздел «Товары на складах» содержит сведения о текущем количестве товаров на складах.
      1. Данные в таблице раздела: id записи, id склада, тип, адрес (страна, субъект, город, улица, дом), id продукта, его название, количество продукта на складе.
      2. Доступны все операции над данными.
      3. Окно элемента должно содержать: id записи в текстового поля; id, тип и адрес склада в виде выпадающего списка; id и название продукта в виде выпадающего списка; количество продукта на складе в виде текстового поля.
10. Группа разделов «Товары».
    1. Раздел «Общие сведения».
       1. Данные в таблице раздела: id товара, его название, описание, цена, id и название поставщика, id и название категории, процент, который получает поставщик с продажи продукта.
       2. Доступны все операции работы с данными.
       3. Окно элемента должно содержать id товара, его название, описание, цену в виде текстовых полей; id и название поставщика в виде выпадающего списка; id и название категории в виде выпадающего списка; процент, который получает поставщик с продажи продукта в виде текстового поля.
       4. Должна быть возможность просмотра и изменения изображений товаров.
    2. Раздел «Отзывы к товару».
       1. Данные в таблице раздела: id отзыва, id товара, его название, id пользователя, его ФИО, текст отзыва, оценка доставки.
       2. Доступны все операции работы с данными.
       3. Данные в окне элемента раздела: id отзыва в виде текстового поля; id и название товара в виде выпадающего списка; id и ФИО пользователя в виде выпадающего списка; текст отзыва и оценка в виде текстовых полей.
       4. Должна быть возможность просмотра и изменения изображений, прикреплённых к отзывам.
    3. Раздел «Категории товаров».
       1. Данные в таблице раздела: id категории, название, id и название родительской категории.
       2. Доступны все операции работы с данными.
       3. Данные в окне элемента раздела: id, название категории, id и название родительской категории в виде выпадающего списка.
    4. Раздел «История цен».
       1. Данные в таблице раздела: id записи, id товара, его название, цена, дата и время изменения цены.
       2. Доступны все операции работы с данными.
       3. Данные в окне элемента раздела: id записи в виде текстового поля; id и название товара в виде выпадающего списка; цена, дата и время изменения цены в виде текстового поля.
11. Группа разделов «Поставщики.
    1. Раздел «Общие сведения».
       1. Данные в таблице раздела: id и название поставщика.
       2. Доступны все операции работы с данными.
       3. Данные в окне элемента раздела: id и название поставщика в виде текстовых полей.
    2. Раздел «Прибыль».
       1. Данные в таблице раздела: id поставщика, его название, суммарная прибыль за всё время.
       2. Раздел доступен только для чтения.
       3. Данные в окне элемента раздела: id, название поставщика, суммарная прибыль за всё время в виде текстовых полей.
12. Группа разделов «Сотрудники».
    1. Раздел «Список сотрудников».
       1. Данные в таблице раздела: id сотрудника, его логин, ФИО, дата рождения, пол, номер телефона, должность.
       2. Доступны все операции работы с данными.
       3. Данные в окне элемента раздела: id сотрудника, его логин, ФИО в виде текстовых полей; дата рождения в виде элемента управления ввода даты; пол в виде переключателя, номер телефона в виде текстового поля, должность в виде выпадающего списка.
       4. В окне элемента должна быть кнопка «Изменить пароль», нажатие на которую вызовет окно с текстовым полем для ввода пароля и кнопками «Изменить» и «Отмена».
       5. Нажатие на кнопку «Изменить» приводит к закрытию окна и успешному изменению пароля.
       6. Нажатие на кнопку «Отмена» приводит к закрытию окна.

Атрибуты качества

1. Фильтрация данных и сортировка в окне раздела (см. ДС-4).
   1. Должна быть возможность фильтровать значения в таблице окна разделов по любому столбцу.
   2. Условия фильтрации: равно, не равно, больше, меньше, содержит.
   3. Условие «содержит» проверяет наличие подстроки.
   4. Также должна быть возможность сортировать записи в таблицах в убывающем или возрастающем порядке.
2. В полях ввода числовых данных должны вводиться только цифры и запятые.

Требования к данным

1. Пароли.
   1. Пароли в базе данных должны храниться в зашифрованном виде. В качестве алгоритма шифрования использовать SHA-256.

Ограничения

1. Доступ к возможностям.
   1. Сотрудник должен видеть только те группы разделов и разделы, к которым имеет доступ (см. БП-2).
   2. У сотрудника не должны отображаться кнопки действий, которых он не может совершить.
2. Изображения в окне элемента.
   1. Доступные форматы: JPEG, PNG, JPG, GIF.
   2. К одной записи в нельзя прикрепить больше 10 изображений, в целях экономии памяти.
   3. Допустимы изображения объёмом до 10 МБ.