1. **Создание проекта и разметка MainWindow.xaml**

Создадим новый проект – выбираем «Приложение WPF (.NET Framework)»

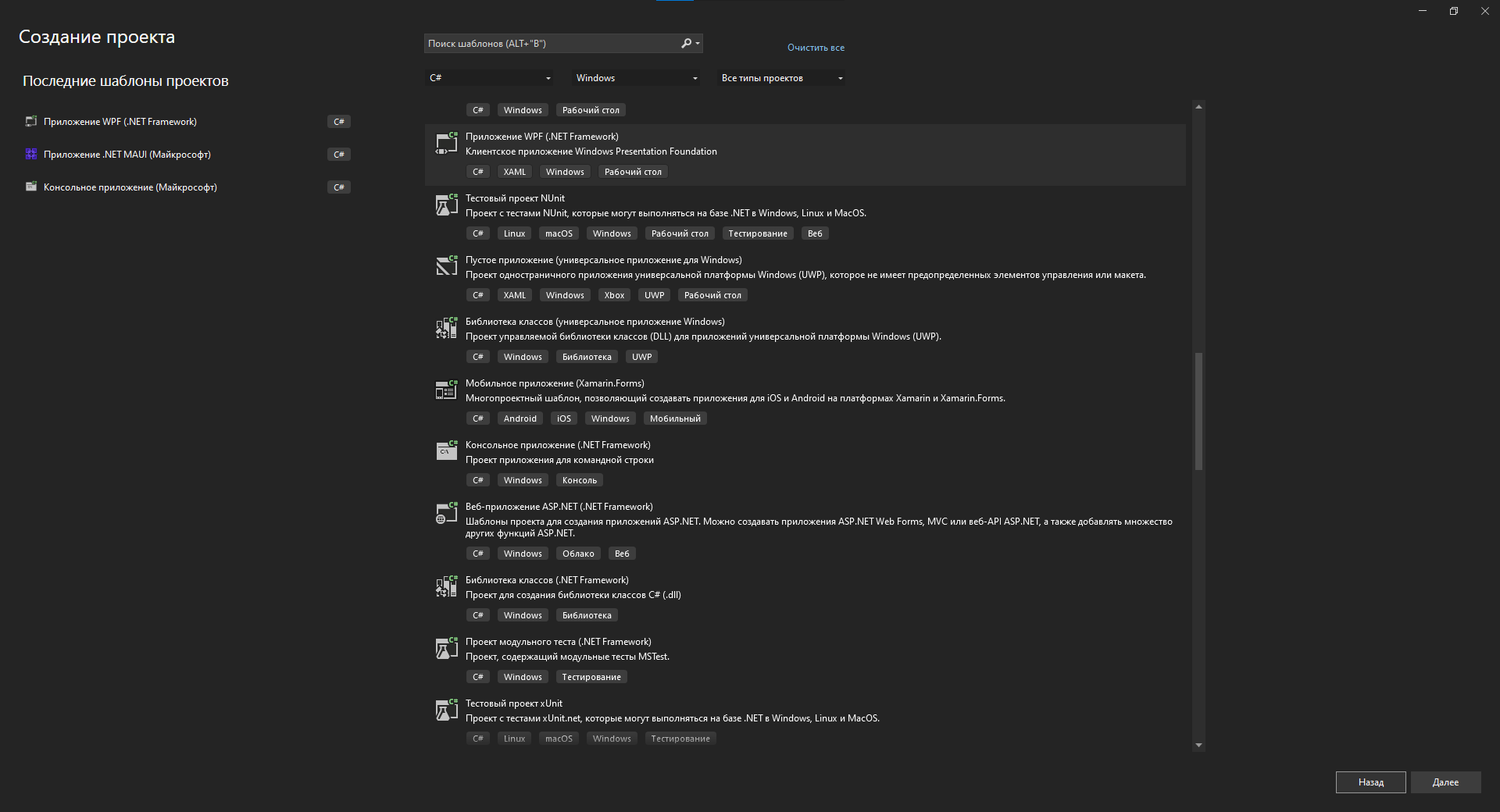


Рисунок 1.1 – создание проекта

В окне «Настроить новый проект» заполняем поля:

* Имя проекта;
* Расположение;
* Выбираем платформу.

После заполнения всех полей нажимаем на кнопку «Создать»

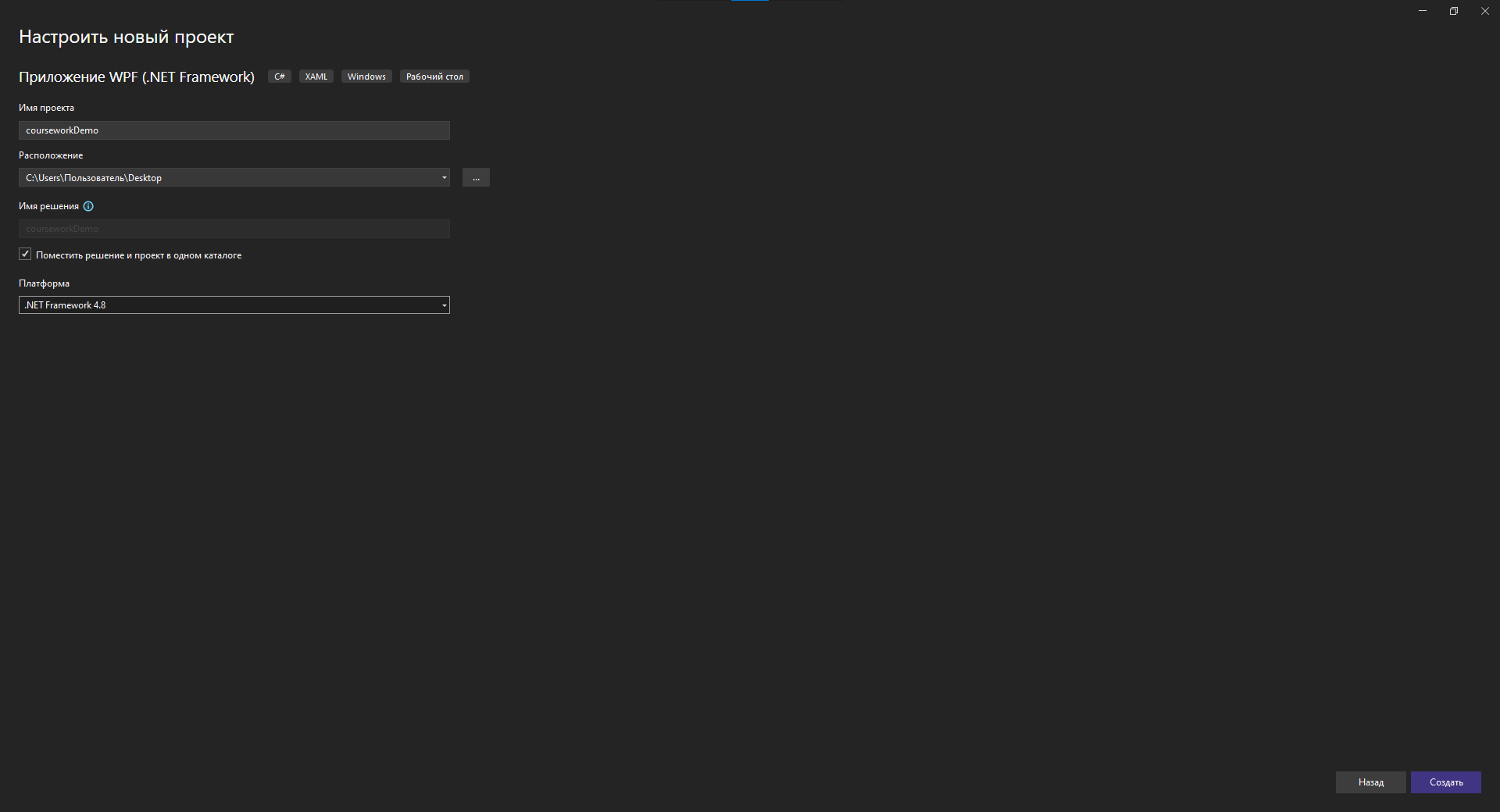


Рисунок 1.2 – настройка нового проекта

Создадим общую структуру проекта. Она будет состоять из следующих директорий:

* Core – директория содержащая вспомогательные классы;
* Images – директория которая содержит графические ресурсы приложения;
* Model – директория содержащая классы данных;
* View – директория которая содержит: окна, страницы, и пользовательские элементы используемые в приложении.

Теперь необходим установить графическую библиотеку «Material Design». Для этого ПКМ нажимаем по проекту в появившемся окне выбираем «Управление пакетами NuGet…» выбираем вкладку «Обзор», в поиске вводим «Material Design» и устанавливаем два компонента «MaterialDesignThemes» и «MaterialDesignColors».

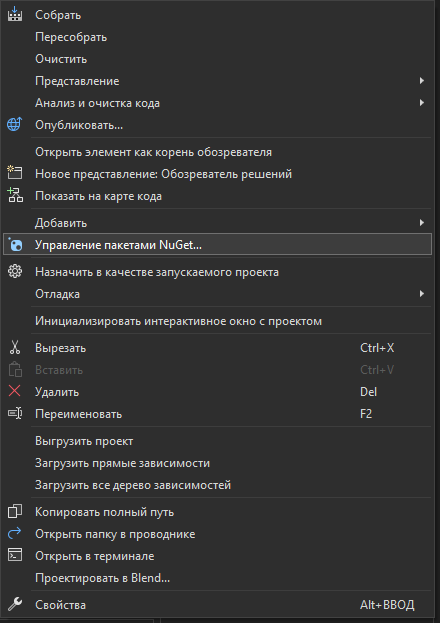


Рисунок 1.3 – Контекстное меню «Управление пакетами NuGet…»

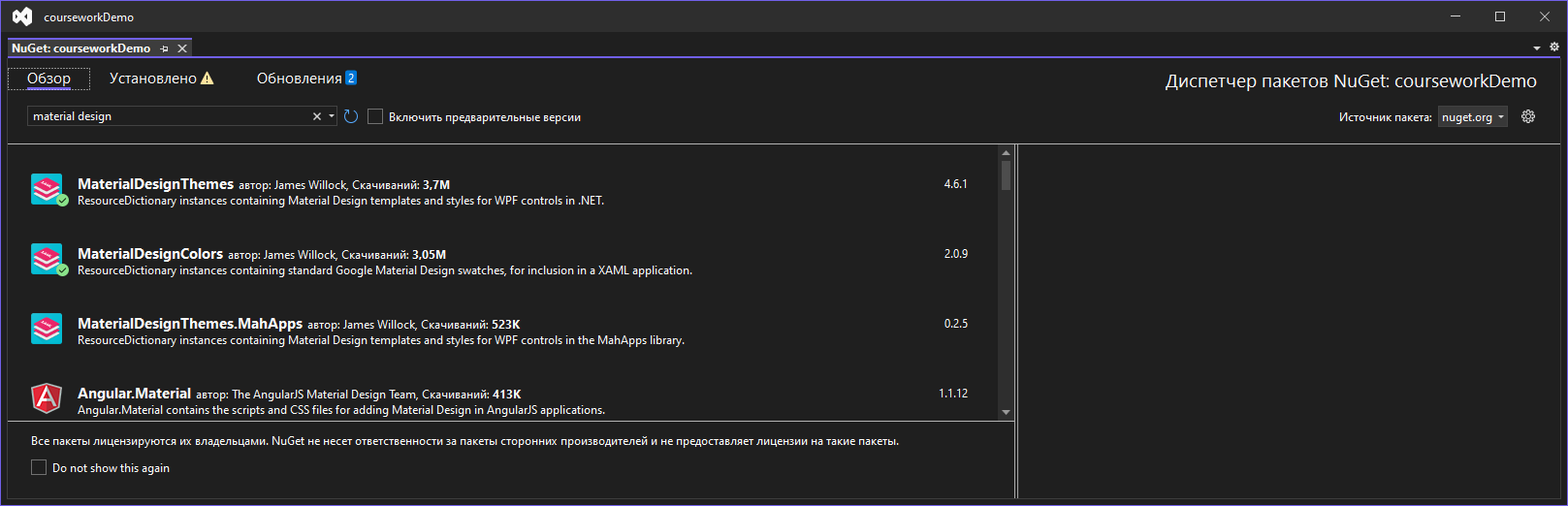


Рисунок 1.4 – Установка «MaterialDesignThemes» и «MaterialDesignColors»

Теперь перенесём «MainWindow.xaml» в директорию «View» и изменим файл «App.xaml». Изменим путь к «MainWindow.xaml» в «StartupUri» и добавим справочник ресурсов, а также добавим новое пространство имён «materialDesign».

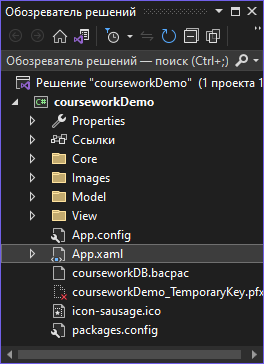


Рисунок 1.5 – выбор файла «App.xaml»

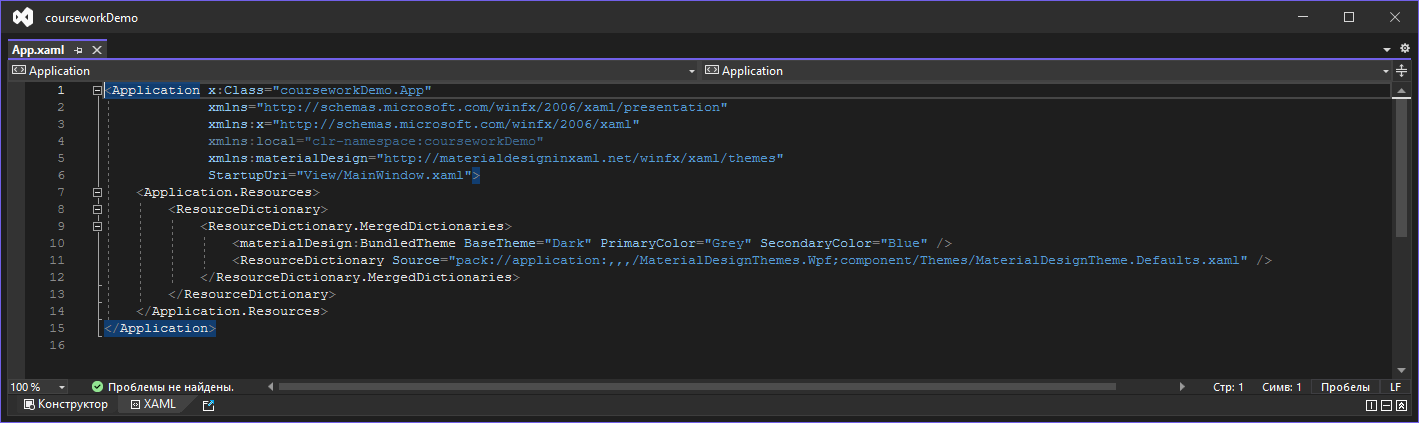


Рисунок 1.6 – редактирование файла «App.xaml»

Далее переходим в директорию «Model». В данную директорию импортируем, используя «EntityFramework» объектно-ориентированную модель базы данных. Для этого ПКМ нажимаем по директории «Model». В открывшемся окне выбираем «Добавить».

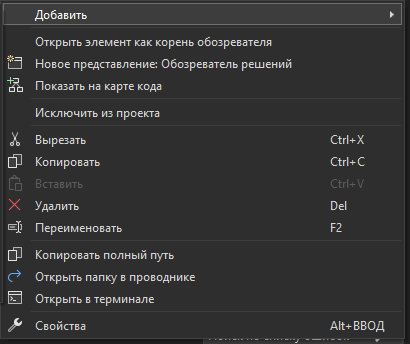


Рисунок 1.7 – выбор кнопки «Добавить»

Далее выбираем «Создать элемент…». В появившемся окне выбираем пункт «Данные» и далее выбираем «Модель ADO.NET EDM», и выбираем кнопку «Далее»

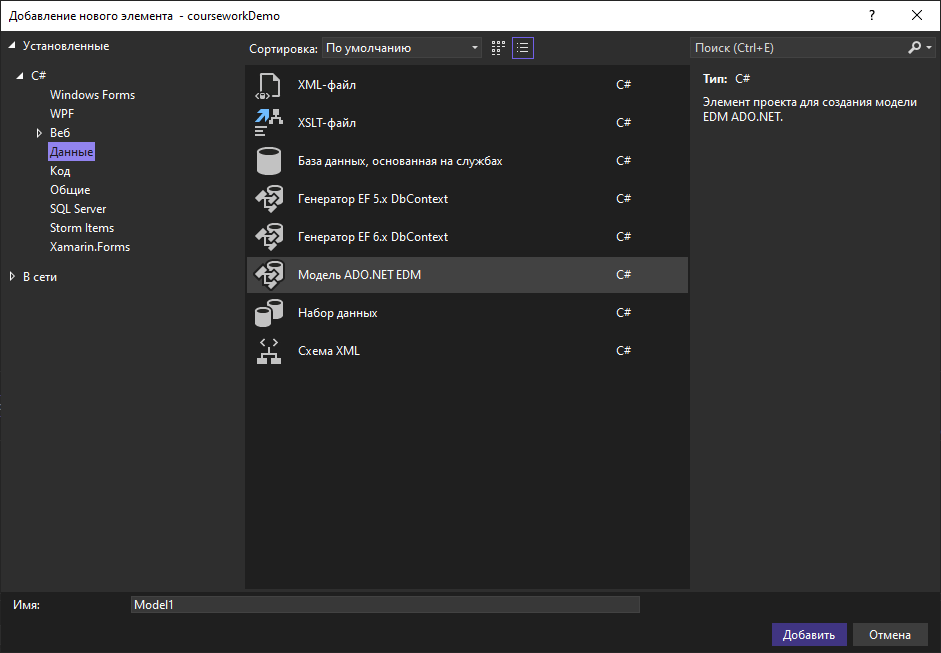


Рисунок 1.8 – Выбор модели «Модель ADO.NET EDM»

В появившемся окне выбираем пункт «Конструктор EF из базы данных» затем нажимаем на кнопку «Далее».

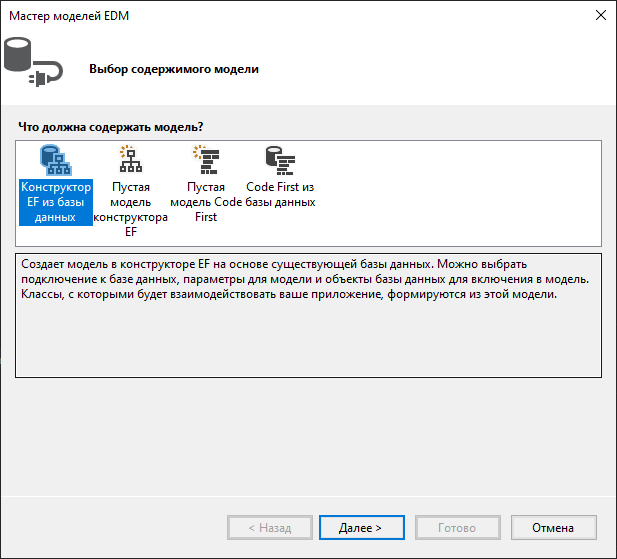


Рисунок 1.9 – Первая страница «Мастер модель EDM»

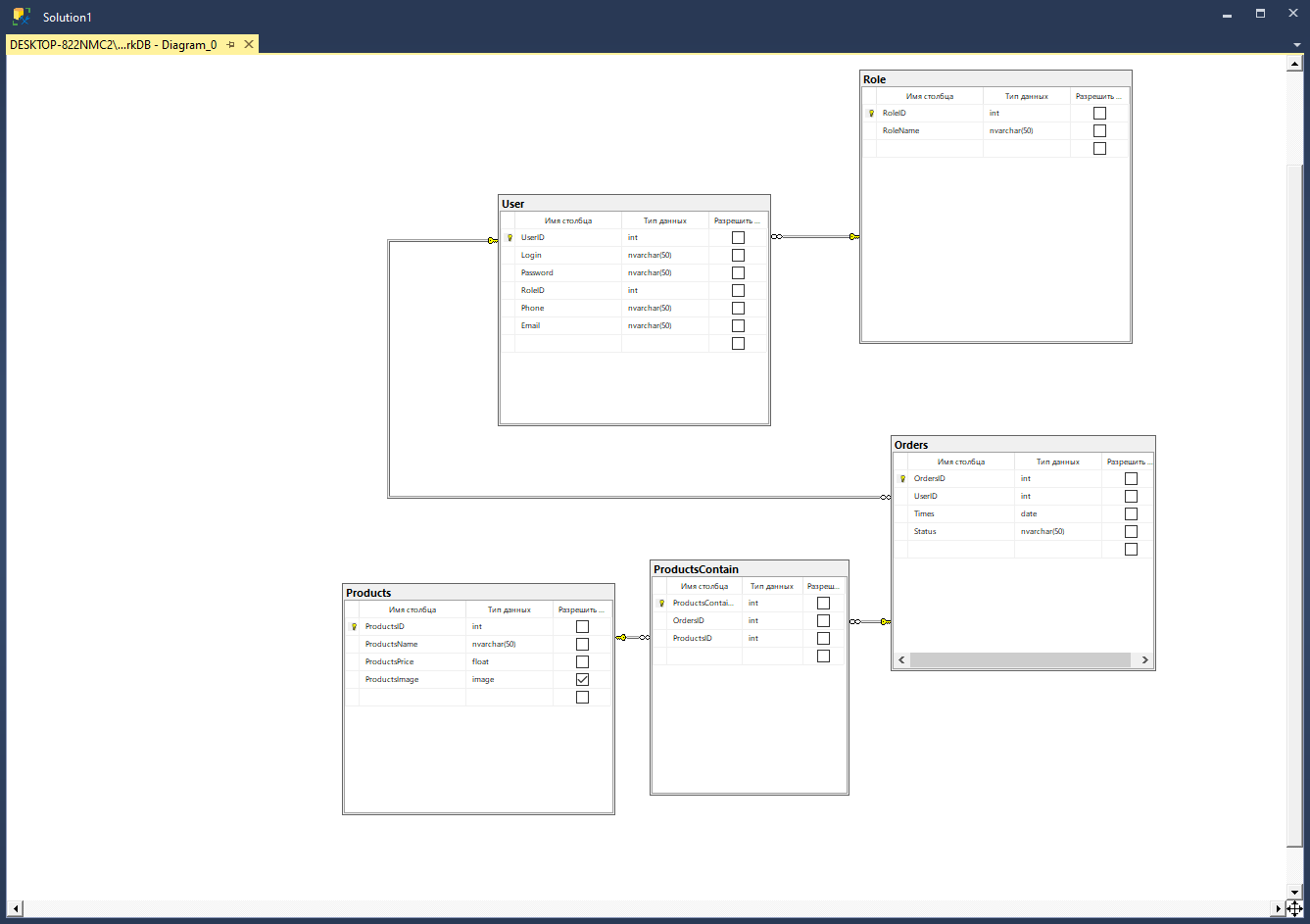


Рисунок 1.10 – Диаграмма базы данных

После импорта переходим к разметке «Main Window». В «MainWindow» добавим ссылку на «materialDesign», изменим параметры высоты и ширины, отключим стандартные стили окна, добавим событие «MouseDown» (данное событие будет отвечать за возможность перемещения пользователем окна), добавим иконку, поставим прозрачный фон, добавим свойство «AllowTransparency» которое разрешает быть окну прозрачным (это нужно для того чтобы сделать страницы “парящими” в воздухе) и запретим пользователю менять размер окна (Разметка «MainWindow» предоставлена на рисунках 1.11 – 1.12).

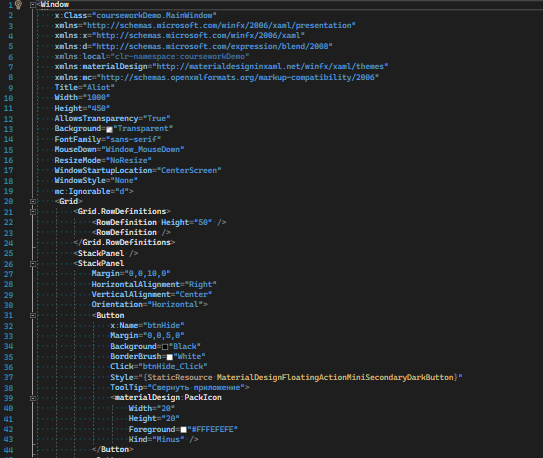


Рисунок 1.11

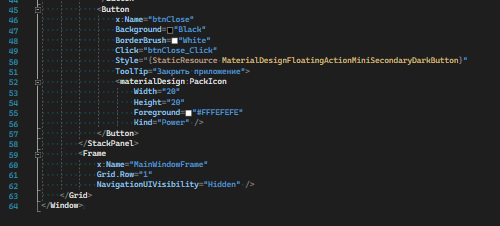


Рисунок 1.12

Перейдем в директорию «Core». В этой директории создадим статический класс «FrameNavigate» с тремя статическими свойствами: «FrameObject» (будет иметь тип данных «Frame»), «DB» (будет иметь тип данных «courseworkDBEntities»), «idUser» (будет иметь тип данных «int». Поле «FrameObject» будет использоваться для упрощенной навигации в приложении. Поле «DB» предоставит доступ к объектно-ориентированной модели данных. Поле «idUser» будет хранить ID пользователя который запустил приложение.

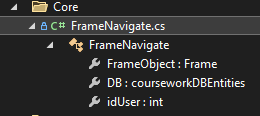


Рисунок 1.13 – Содержание директории «Core»

Далее переходим в директорию «View». Создадим три директории:

- AdministrationPage – будет содержать страницы для роли администратор,

- LoginPage – будет содержать страницы для входа и регистрации в приложении,

- UserPage – будет содержать страницы для роли пользователя.

Директории «LoginPage» создадим две страницы «MainWindowLoginPage» и «MainWindowRegistrationPage». Далее переходим в файл «MainWindow.xaml.cs». В этом файле свяжем поле (из класса «FrameNavigation») «FrameObject» с «MainWindowFrame», а в поле «DB» создадим экземпляр «courseworkDBEntities». Также реализуем события: «Window\_MouseDown» – данное событие разрешить пользователю перемещать «окно» приложения; «BtnClose\_Click» – данное событие будет завершать работу приложения; «btnHide\_Click» - данное событие будет сворачивать приложение.



Рисунок 1.14 - Реализация логики в файле «MainWindow.xaml.cs»

Переходим на страницу «MainWindowLoginPage». Импортируем пространство имен «materialDesign» и переходим к разметке (Разметка «MainWindowLoginPage» приведена на рисунках 1.15 – 1.18)

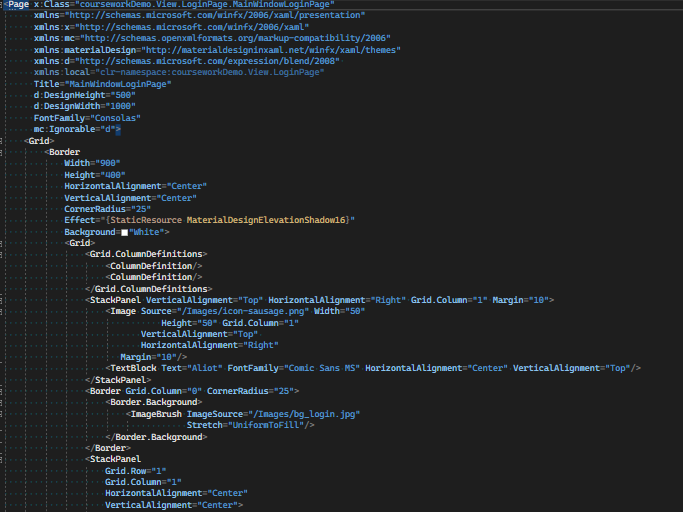


Рисунок 1.15



Рисунок 1.16

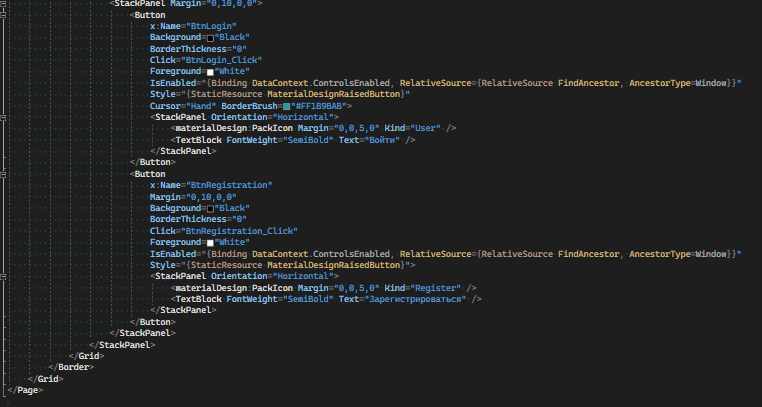


Рисунок 1.17

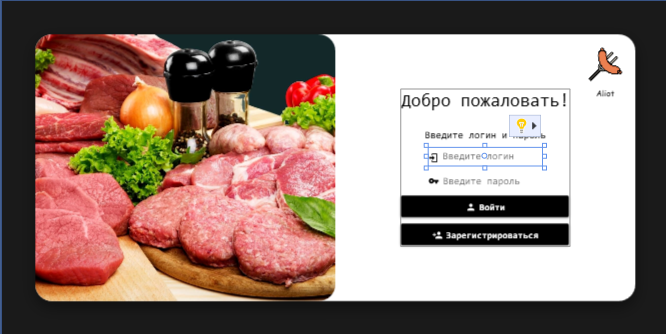


Рисунок 1.18

Далее переходим в файл «MainWindowLoginPage.cs». Этот файл будет содержать логику авторизации при клике на кнопку «Войти», выдачу сообщения об ошибке через MessageBox (Реализация логики приведена на рисунке 1.19).

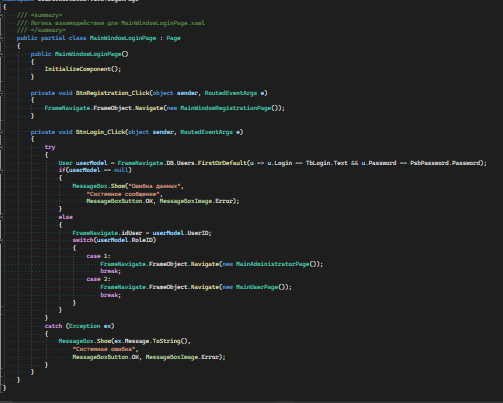


Рисунок 1.19

**2 Реализация «MainWindowRegistrationPage»**

Логика «MainWindowRegistrationPage» будет следующая: пользователь вводит данные, если все в порядке, то выводится сообщение об успешной регистрации

Далее создадим разметку «MainWindowRegistrationPage» (Реализация представлена на рисунках 2.1 – 2.4)

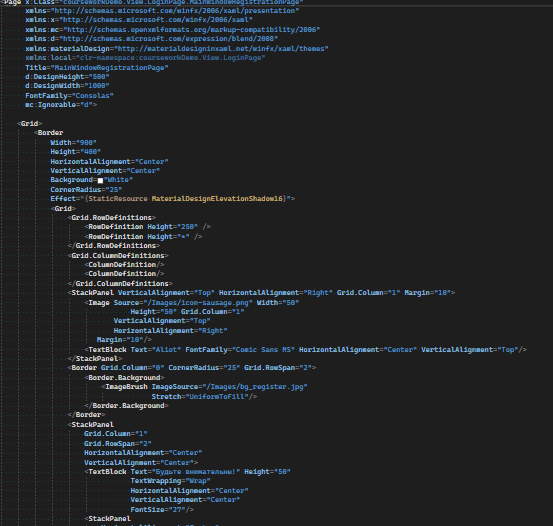


Рисунок 2.1



Рисунок 2.2

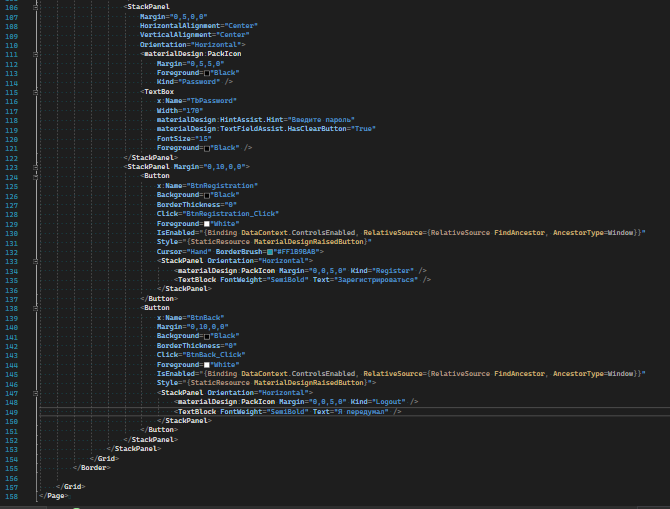


Рисунок 2.3

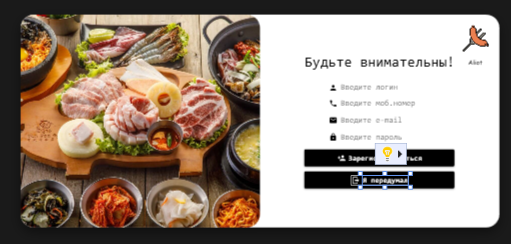


Рисунок 2.4

Перейдём в файл «MainWindowRegistrationPage» для реализации логики регистрации. При заполнении всех полей будет всплывать MessageBox с сообщением об успешной регистрации.(Реализации логики на рисунке 2.5)



Рисунок 2.5

**3 Реализация AdministrationPage**

Теперь переходим в директорию «AdministrationPage» и создаём страницу «MainAdministratorPage» в которой будет реализовано Menu с содержанием всех AdminControl. (Реализация приведена на рисунках 3.1 – 3.3)

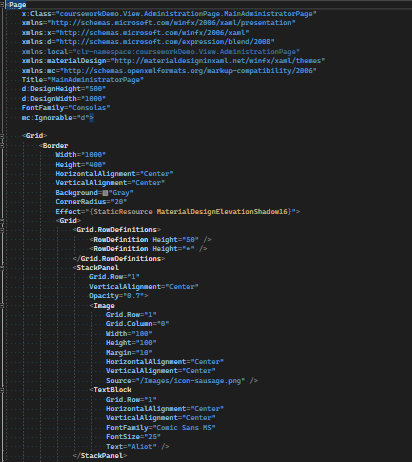


Рисунок 3.1

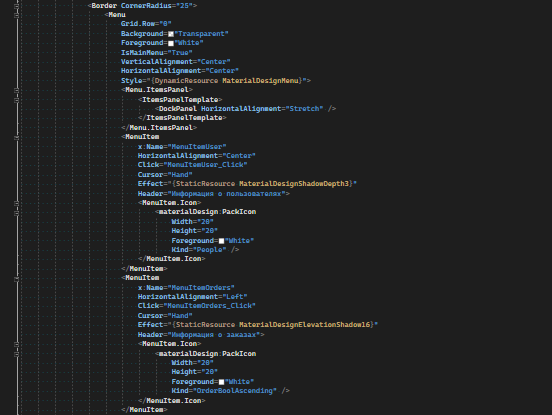


Рисунок 3.2



Рисунок 3.3

Далее перейдём в файл «MainAdministratorPage.cs» в котором пропишем логику переключения между AdminControl. (Реализация приведена на рисунке 3.4).

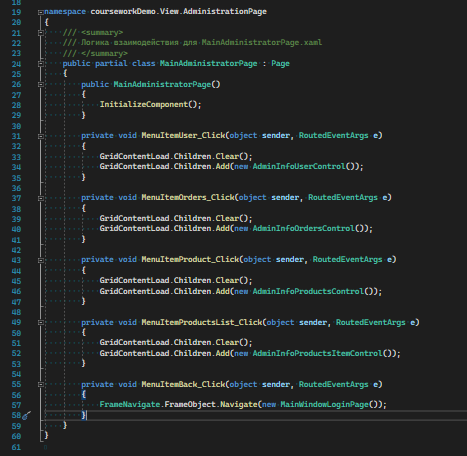


Рисунок 3.4

Теперь создадим в директории «AdministrationPage» директорию «UserControl» в которой будут содержаться «Пользовательские элементы управления»:

* AdminInfoOrdersControl – будет реализована логика принятия или отклонения заказа. Также будет выведена таблица со всеми заказами.
* AdminInfoProductsControl – будет реализована логика добавления товара.
* AdminInfoProductsItemControl - будет реализована логика удаления товара. Также будет выведена таблица со всеми товарами.
* AdminInfoUserControl - будет реализована логика удаления пользователей. Также будет выведена таблица со всеми пользователями.

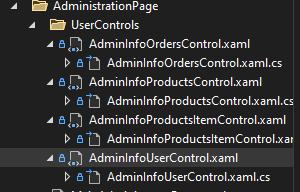


Рисунок 3.5 - Пользовательские элементы управления директории «AdministrationPage»

Создадим разметку «AdminInfoOrdersControl» (Реализация представлена на рисунках 3.6 – 3.7)

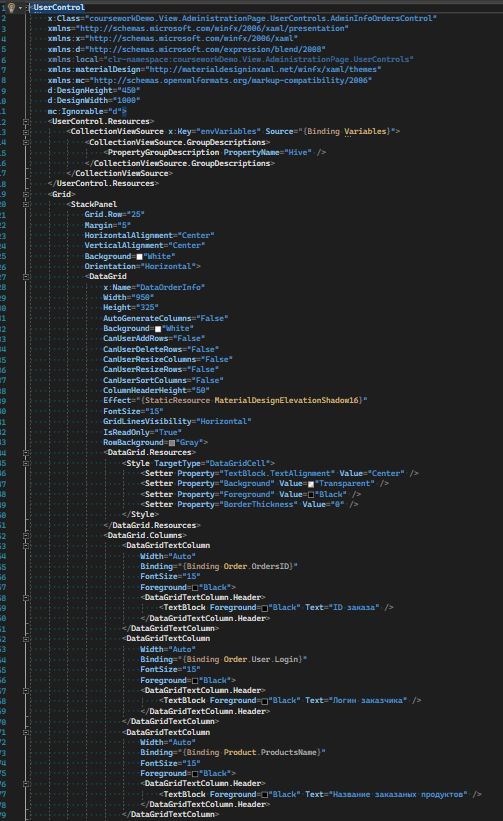


Рисунок 3.6

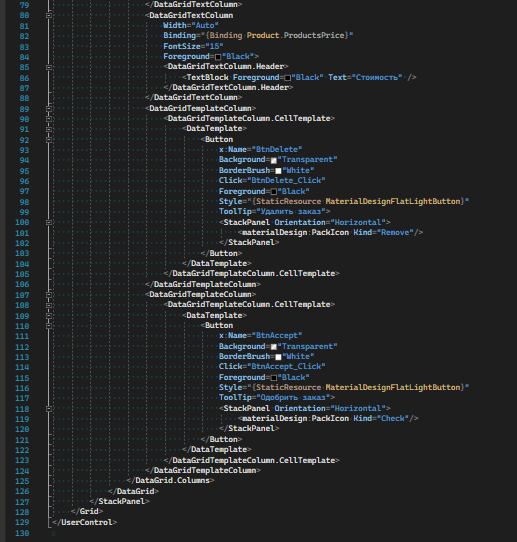


Рисунок 3.7

Далее перейдём в файл «AdminInfoOrdersControl.cs» в котором будет реализована логика принятия или отклонения заказа. (Реализация приведена на рисунке 3.8)

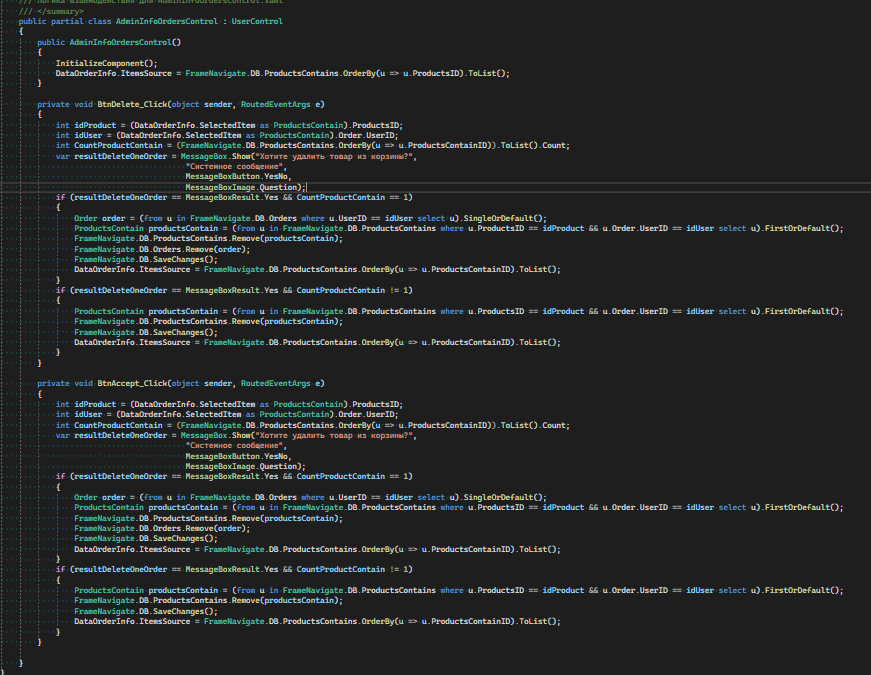


Рисунок 3.8

Создадим разметку «AdminInfoProductsControl» (Реализация представлена на рисунках 3.9 – 3.10)



Рисунок 3.9

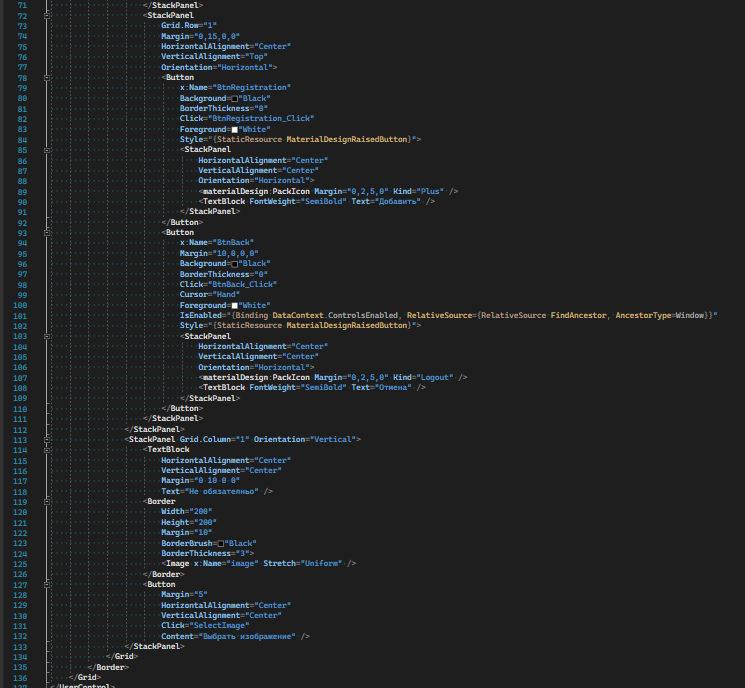


Рисунок 3.10

Далее перейдём в файл «AdminInfoProductsControl.cs» в котором будет реализована логика добавления товара. (Реализация представлена на рисунке 3.11)



Рисунок 3.11

Создадим разметку «AdminInfoProductsItemControl» (Реализация представлена на рисунках 3.12 – 3.)

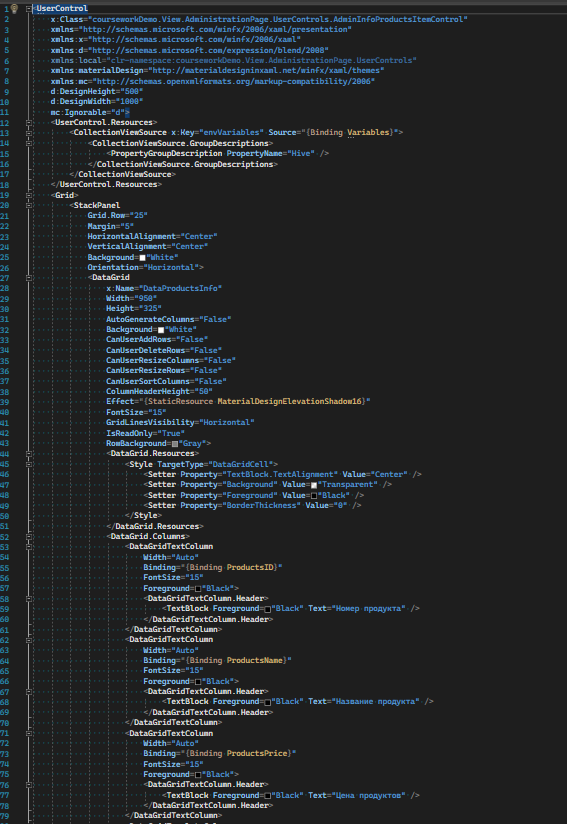


Рисунок 3.12



Рисунок 3.13

Далее перейдём в файл «AdminInfoProductsItemControl.cs» в котором будет реализована логика удаления товара. (Реализация представлена на рисунке 3.14)

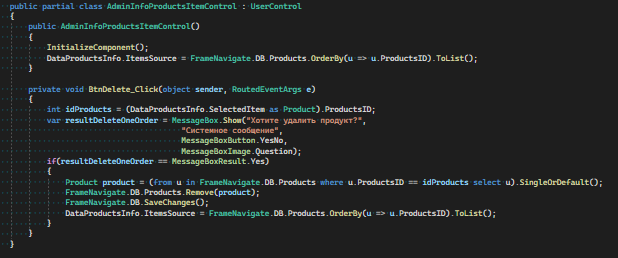


Рисунок 3.14

Создадим разметку «AdminInfoUserControl» (Реализация представлена на рисунках 3.15 – 3.)

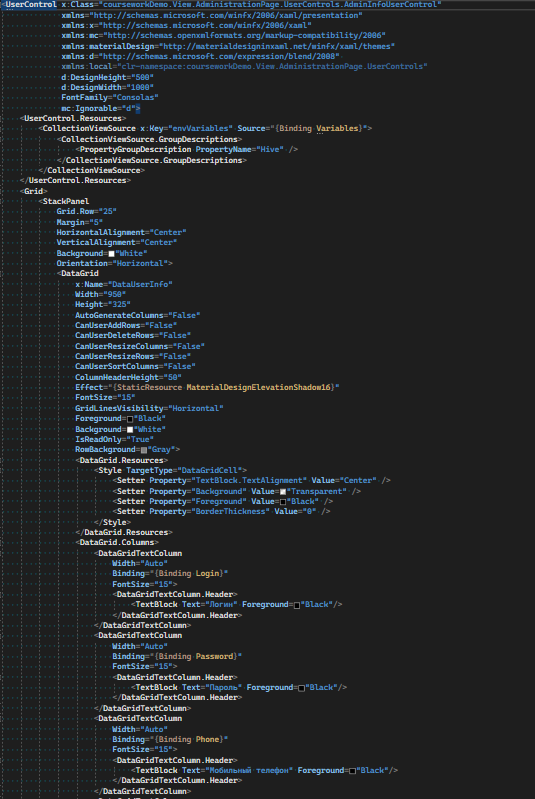


Рисунок 3.15

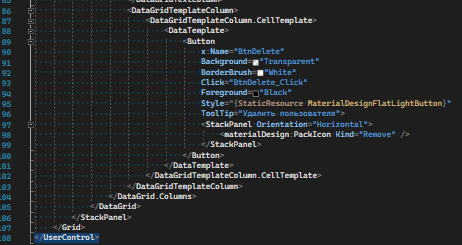


Рисунок 3.16

Далее перейдём в файл «AdminInfoUserControl.cs» в котором будет реализована логика удаления пользователей. (Реализация представлена на рисунке 3.17)

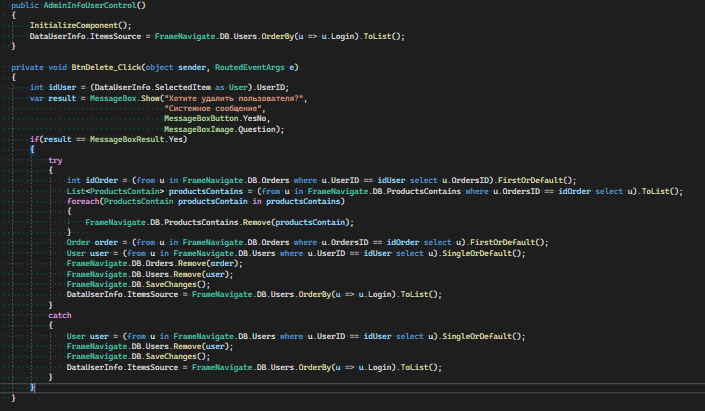


Рисунок 3.17

**4 Реализация UserPage**

В директории «UserPage» создадим страницу «MainUserPage». Также создадим директорию «UserControls» в которой будут храниться «Пользовательские элементы управления» «UserBasketConrtol».

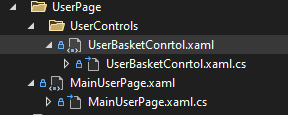


Рисунок 4.1 – Директория «UserPage»

Теперь создадим разметку «MainUserPage» (Реализация представлена на рисунках 4.2 – 4.4)



Рисунок 4.2



Рисунок 4.3

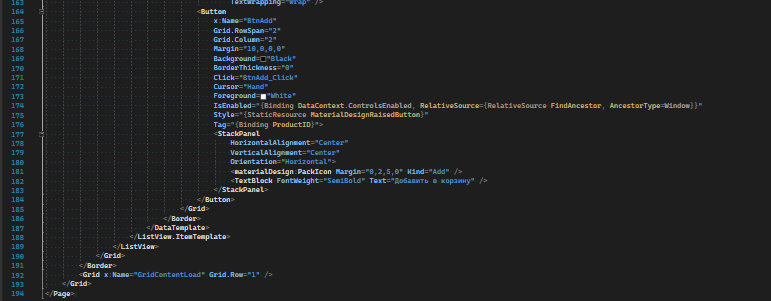


Рисунок 4.4

Далее переходим в файл «MainUserPage.cs» в котором будет содержаться логика поиска продуктов и добавления продукта в корзину. (Реализация представлена на рисунках 4.5)

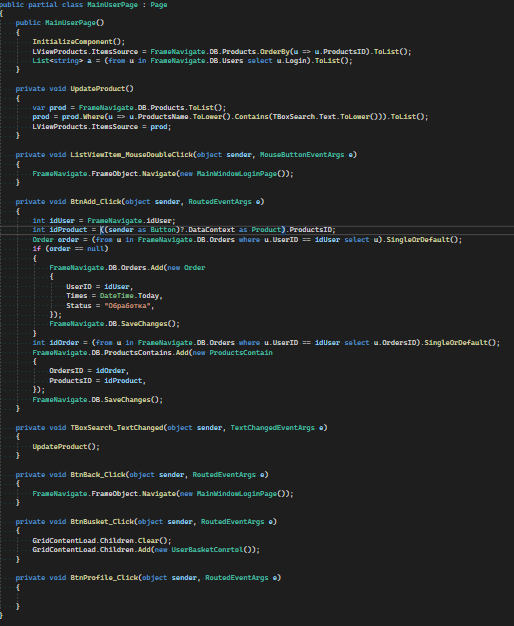


Рисунок 4.5

Теперь перейдём в директорию «UserControls» в которой перейдём в файл «UserBasketConrtol». Затем создадим разметку. (Реализация предоставлена на рисунках 4.6 - )



Рисунок 4.6

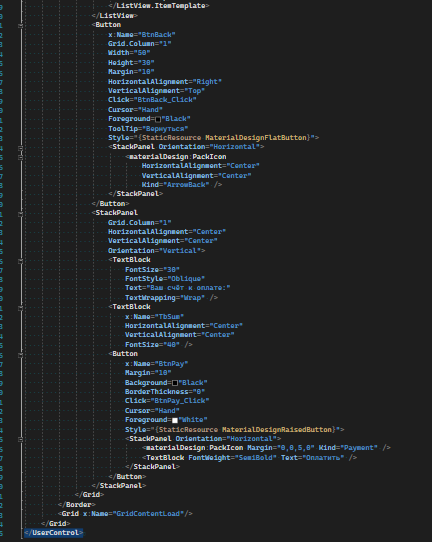


Рисунок 4.7

Далее переходим в файл «UserBasketConrtol.cs» в котором будет содержаться логика удаления товара из корзины и оплата всего заказа. (Реализация представлена на рисунке 4.8)

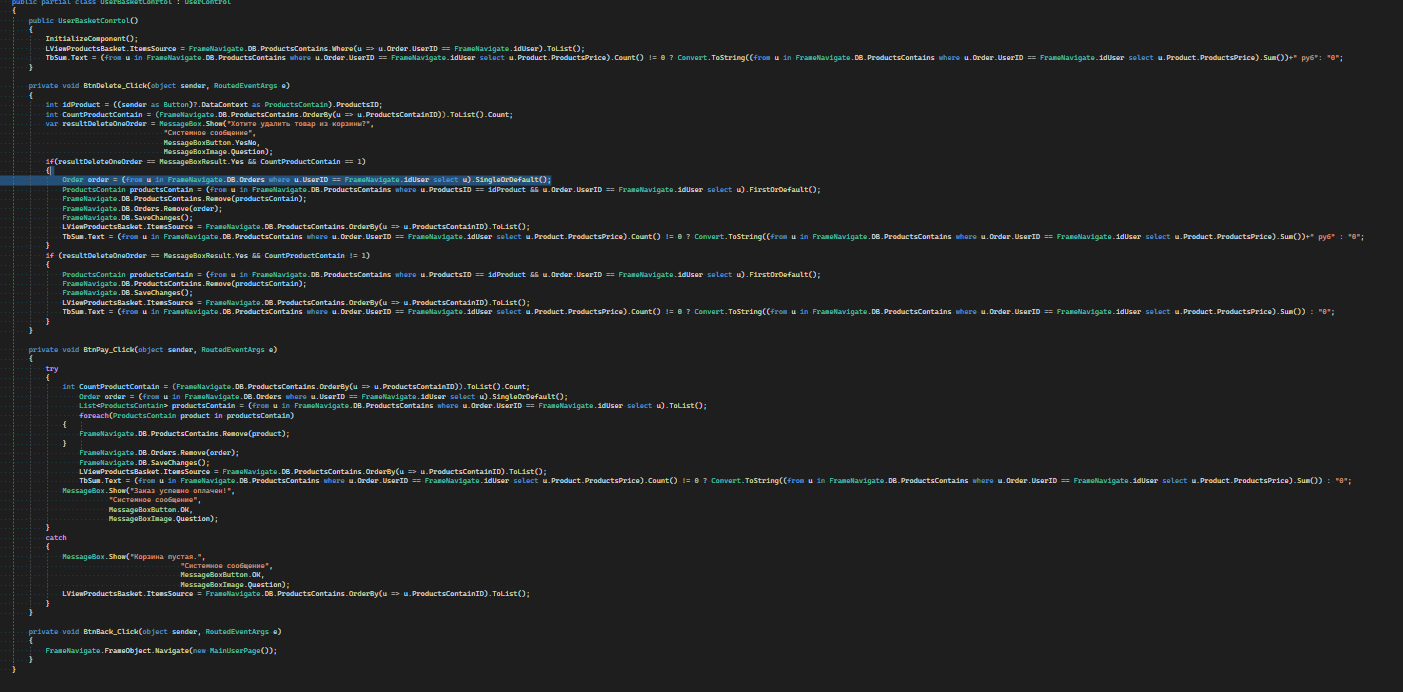


Рисунок 4.8