**1 Создание проекта, разметка MainWindow и реализация регистрации пользователей**

Создадим новый проект – выбираем «Приложение WPF (.Net Framework)» и нажимаем «Далее».

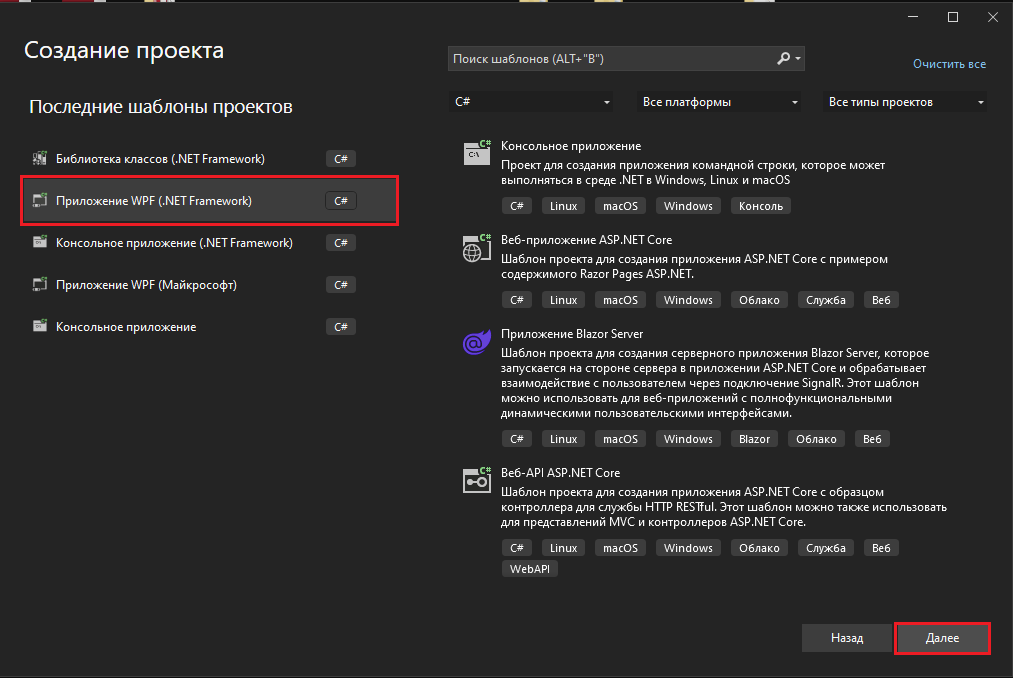


Рисунок 1.1 – Создание проекта

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

Теперь необходимо 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».

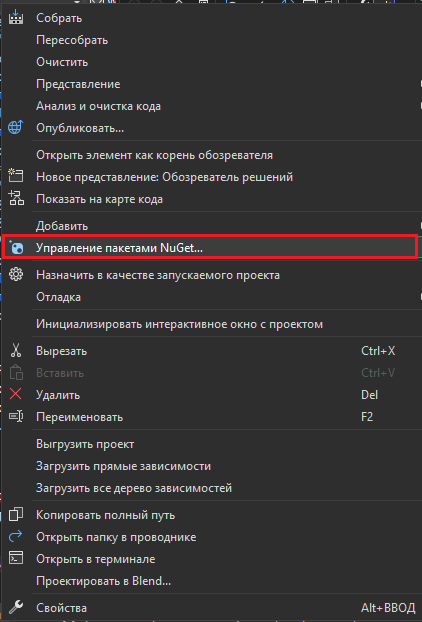


Рисунок 1.3 – Контекстное меню «Управление пакетами NuGet…»

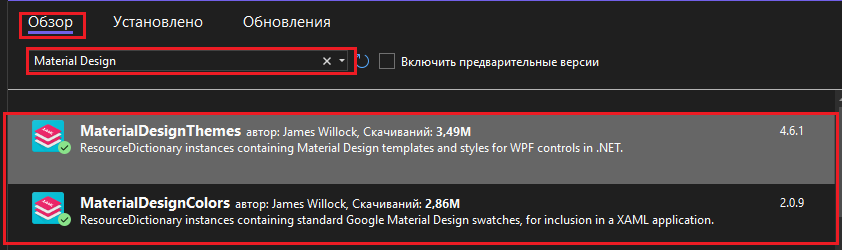


Рисунок 1.4 – Установка «MaterialDesignThemes» и «MaterialDesignColors».

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

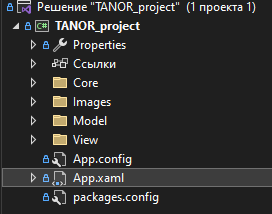


Рисунок 1.5 – Выбор файла «App.xaml»

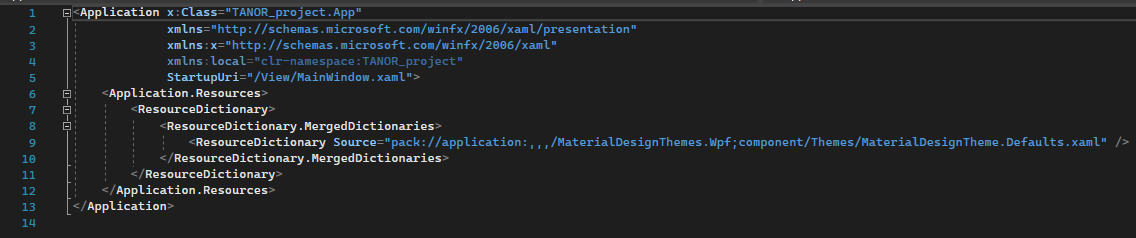


Рисунок 1.6 – Редактирование файла «App.xaml»

Далее переходим в директорию «Model». В данную директорию импортируем, используя «EntityFramework» объектно-ориентированную модель базы данных. После импорта переходим к разметке «MainWindow».

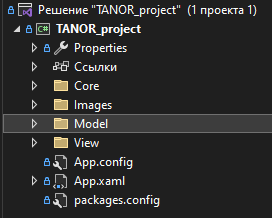


Рисунок 1.7 – Выбор директории файла «Model»

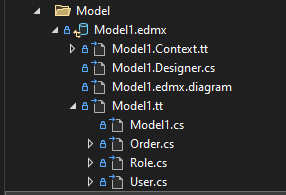


Рисунок 1.8 – Импорт данных

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) и последним параметром мы запретим пользователю менять размер окна (Разметка «MainWindow» предоставлена на рисунках 1.9 – 1.12).



Рисунок 1.9 – Разметка «MainWindow»

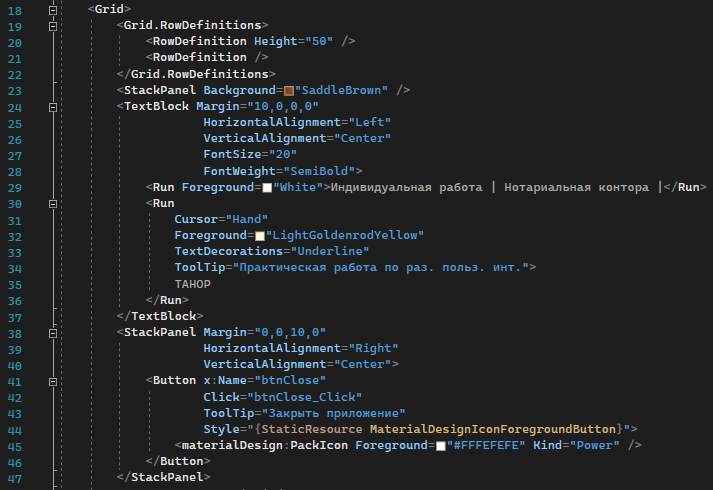


Рисунок 1.10 – Разметка «MainWindow»

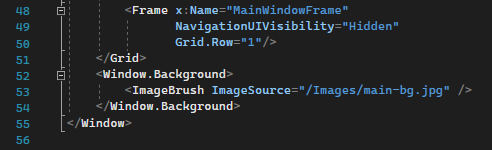


Рисунок 1.11 – Разметка «MainWindow»

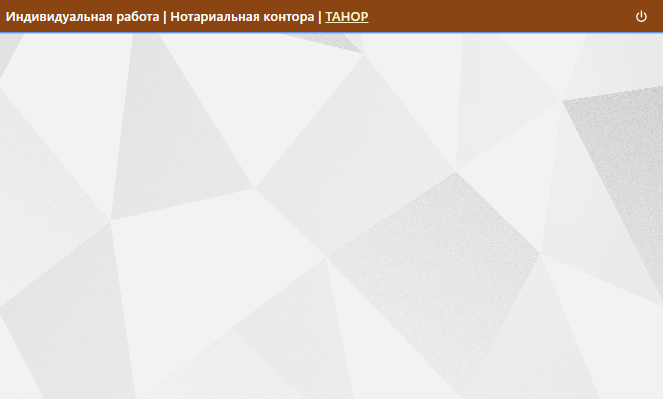


Рисунок 1.12 – Итог разметки «MainWindow»

Перейдем в директорию «Core». В этой директории создадим статический класс «FrameNavigate» с двумя статическими свойствами «FrameObject» (будет иметь тип данных «Frame») и «DB» (будет иметь тип данных «FreelanceWorkDBEntities»). Поле «FrameObject» будет использоваться для упрощенной навигации в приложении. Поле «DB» предоставить доступ объектно-ориентированной модели данных.

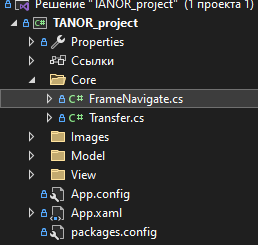


Рисунок 1.13 – Директория «Core»

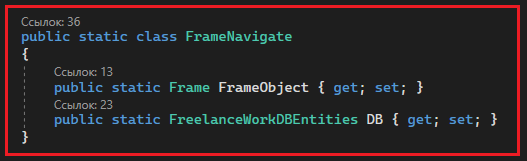


Рисунок 1.14 – Реализация класса «FrameNavigate»

Также перейдем в директорию «Core». В этой директории создадим статический класс «Transfer» с тремя статическими свойствами «Login», «idOrder» и «idUser» (будут иметь строковый тип данных).

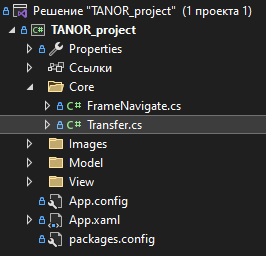


Рисунок 1.15 – Директория «Core»

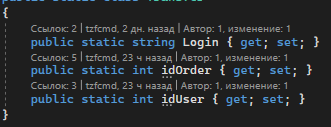


Рисунок 1.16 – Реализация класса «Transfer»

Далее переходим в директорию «View». Создадим пять директорий:

* UserPage – будет содержать страницы для роли пользователя;
* LoginPage – будет содержать страницы для входа и регистрации;
* LawyerPage – будет содержать страницы для роли работодателя;
* AdminPage – будет содержать страницы для роли администратора.
* InfoPage – будет содержать информационные страницы для всех все пользователей.

В директории «LoginPage» создаем две страницы «MainLoginPage» и «MainRegPage». Далее переходим в файл «MainWindow.xaml.cs». В этом файле свяжем поле (из класса «FrameNavigate») «FrameObject» с «MainWindowFrame», а в поле «DB» создадим экземпляр «FreelanceWorkDBEntities». Также реализуем два события: «Window\_MouseDown» – данное событие разрешить пользователю перемещать «окно» приложения; «BtnClose\_Click» – данное событие будет завершать работу приложения.

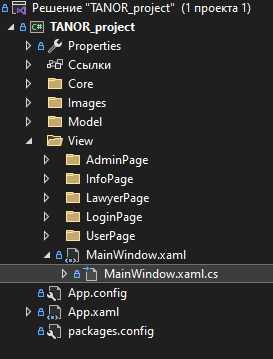


Рисунок 1.17 – Файл «MainWindow.xaml.cs»

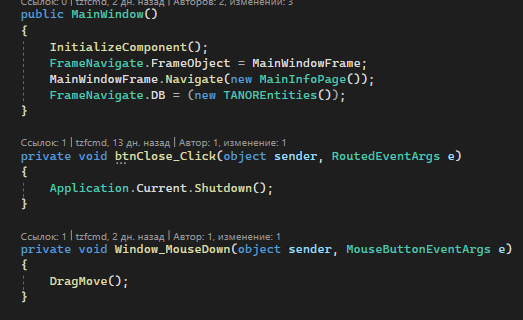


Рисунок 1.18 – Реализация логики в файле «MainWindow.xaml.cs»

Переходим в директорию «InfoPage». Реализуем 3 страницы, которые будут отвечать за представление информации о конторе для пользователя, а также содержать кнопку «Личные кабинет», по нажатию которой пользователь либо регистрируется, либо авторизуется в программе.

Реализуем главную страницу «MainInfoPage»(разметка на рисунках 1.20-1.23, логика на рисунке 1.24).

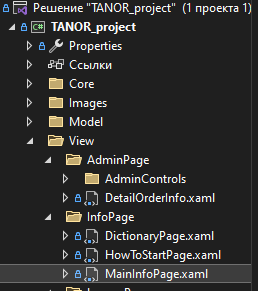


Рисунок 1.19 – Директория «InfoPage»



Рисунок 1.20 – Разметка «MainInfoPage»



Рисунок 1.21 – Разметка «MainInfoPage»

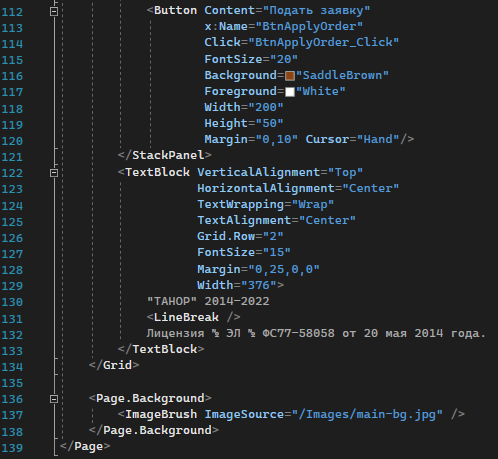


Рисунок 1.22 – Разметка «MainInfoPage»

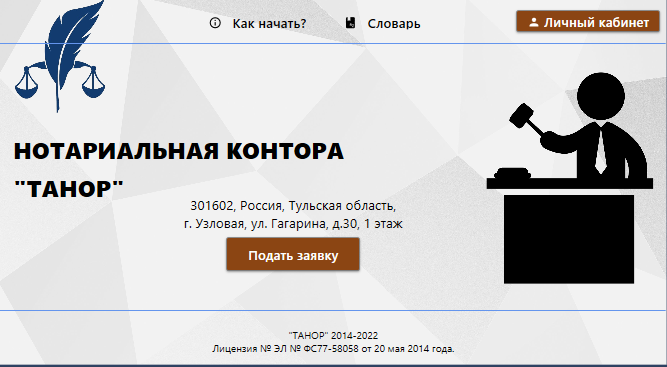


Рисунок 1.23 – Итог разметки «MainInfoPage»



Рисунок 1.24 – Реализация логики «MainInfoPage»

Далее реализуем информационные страницы «DictionaryPage» (рисунки 1.25 – 1.29) и «HowToStartPage» (рисунки 1.30-1.34), на которые пользователь переходит с помощью кнопок «Словарь» и «Как начать?», а по кнопке «Личный кабинет» он попадает на страницу регистрации.

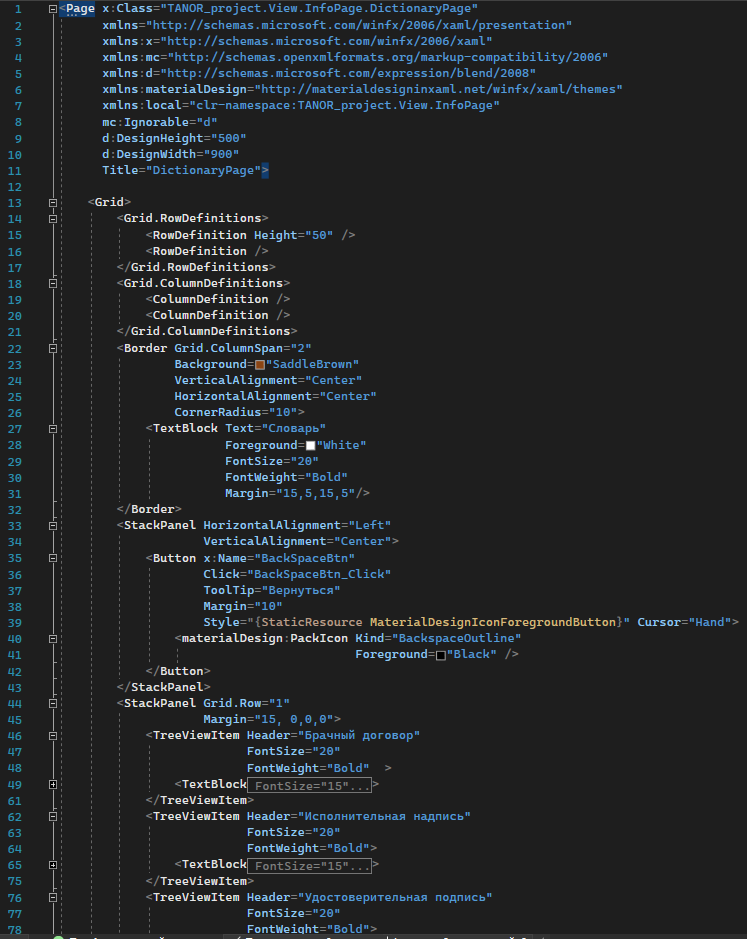


Рисунок 1.25 – Разметка «DictionaryPage»

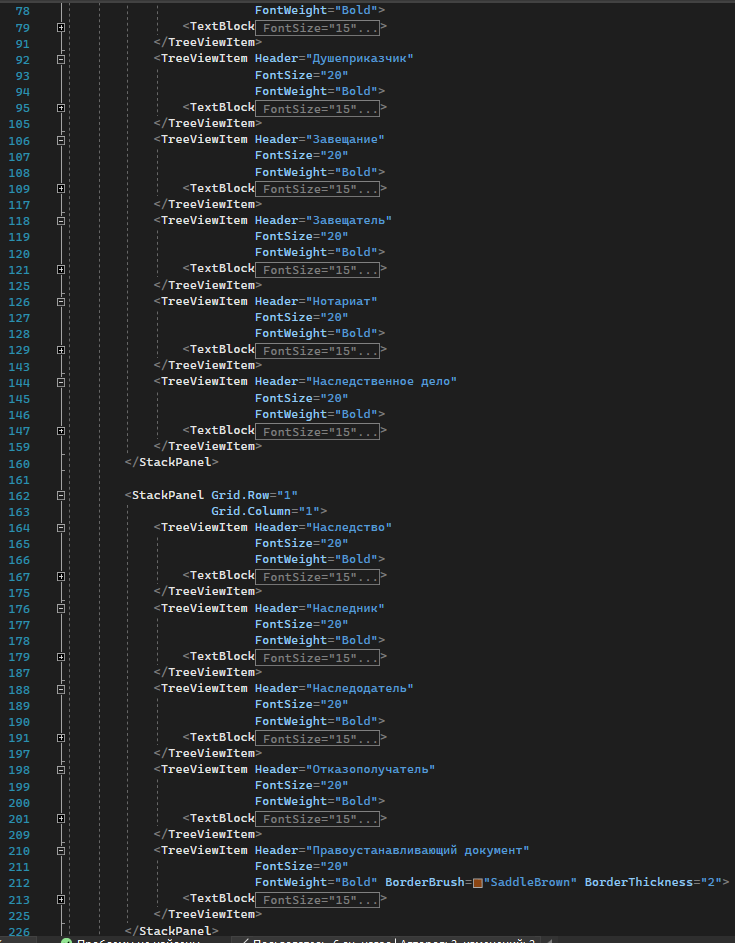


Рисунок 1.26 – Разметка «DictionaryPage»

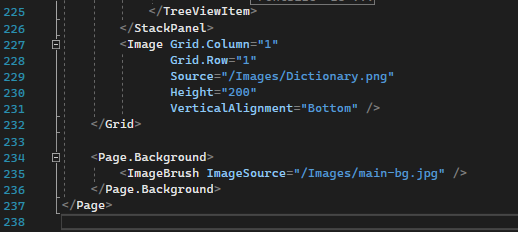


Рисунок 1.27 – Разметка «DictionaryPage»

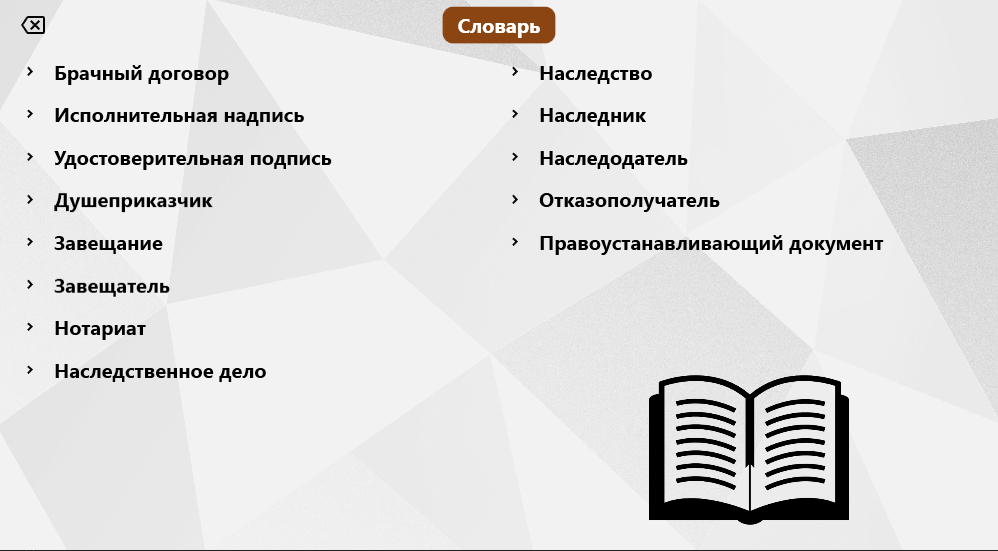


Рисунок 1.28 – Итог разметки «DictionaryPage»

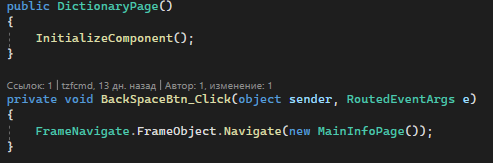


Рисунок 1.29 – Реализация логики «DictionaryPage»

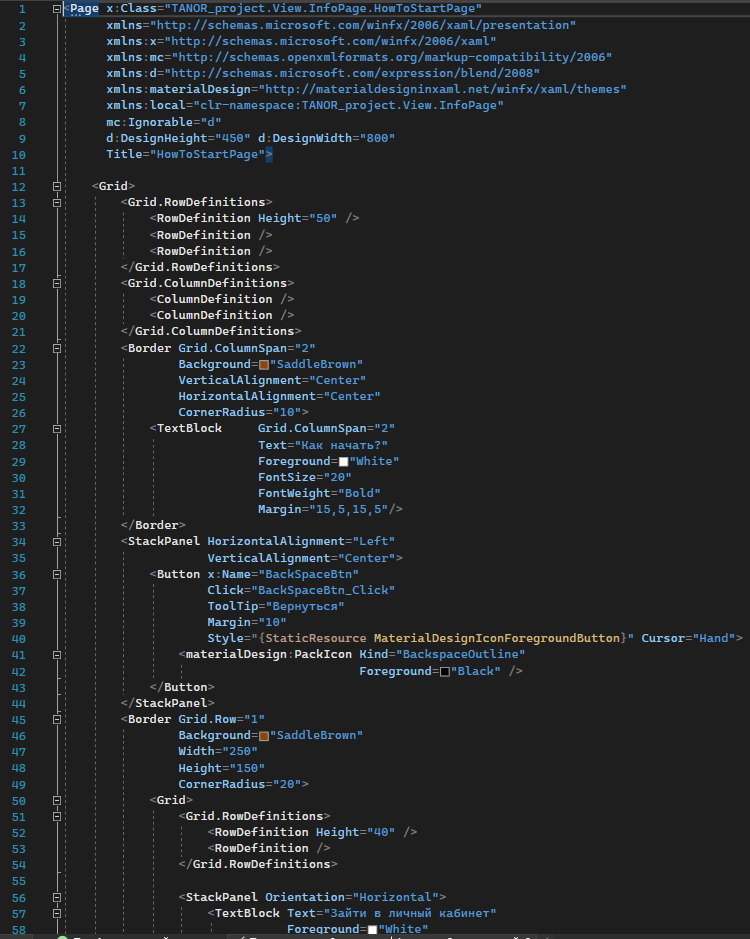


Рисунок 1.30 – Разметка «HowToStartPage»

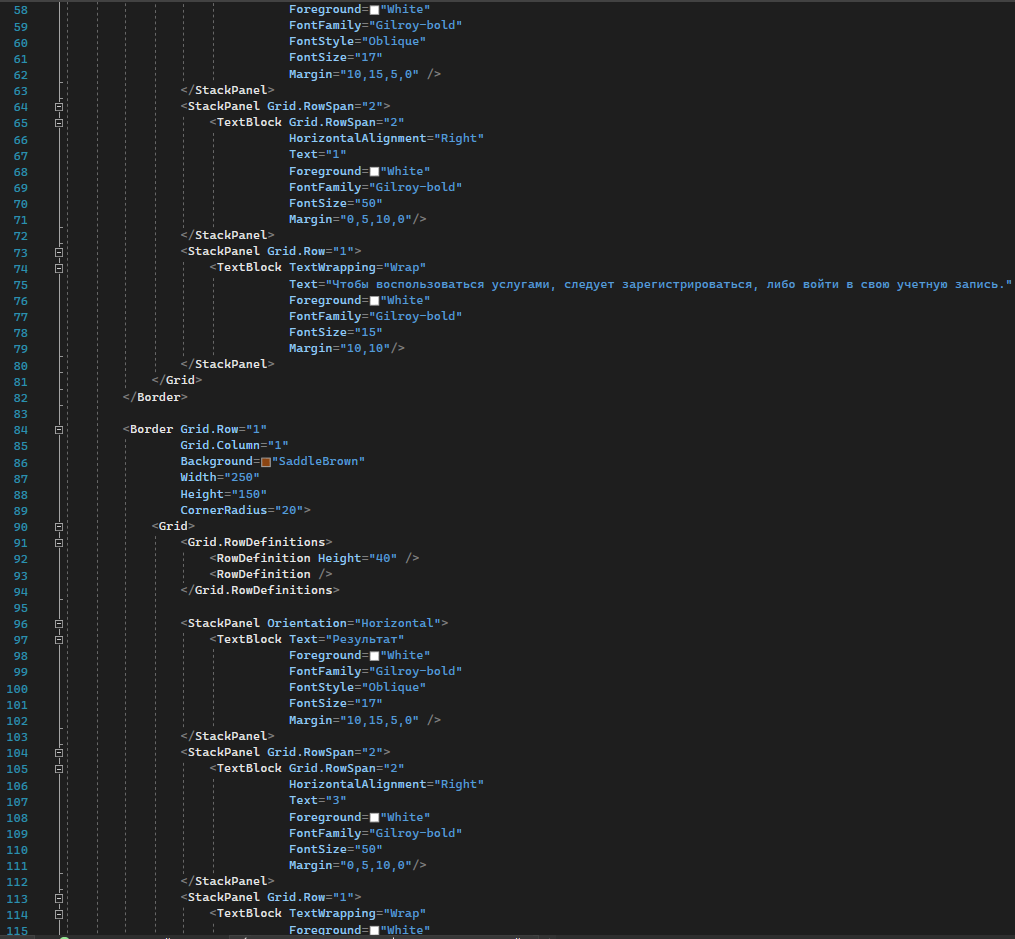


Рисунок 1.31 – Разметка «HowToStartPage»

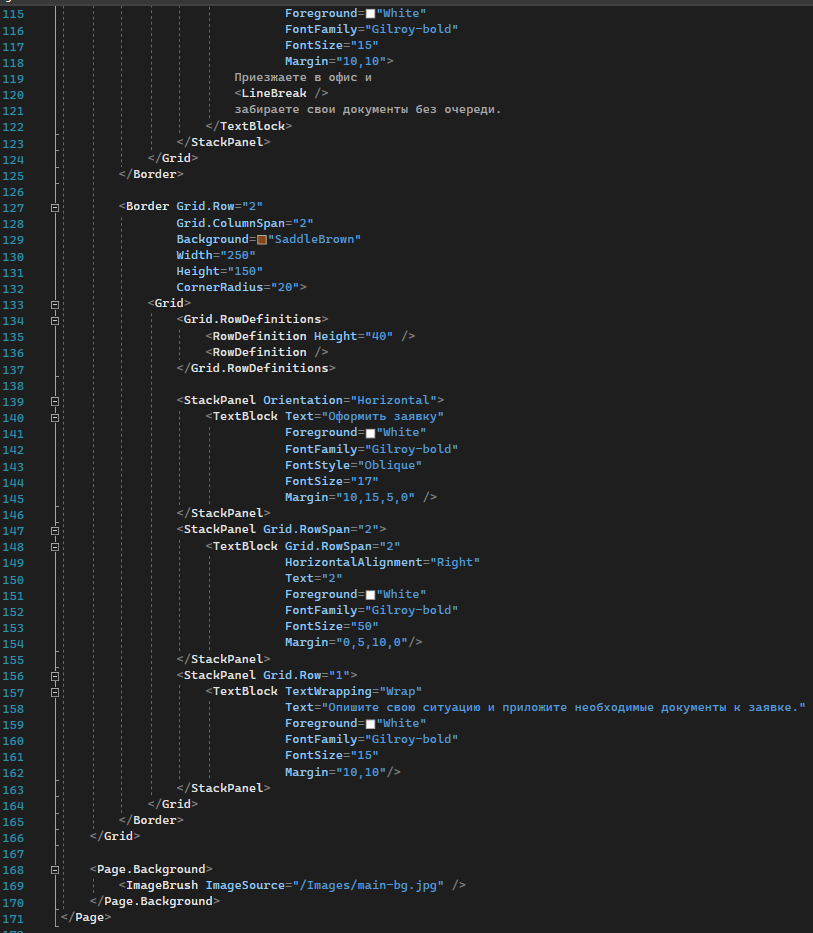


Рисунок 1.32 – Разметка «HowToStartPage»



Рисунок 1.33 – Итоги разметки «HowToStartPage»

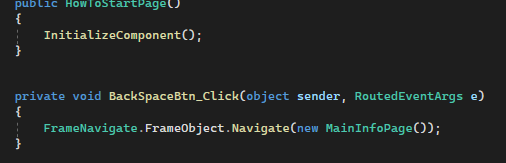


Рисунок 1.34 – Реализация логики «HowToStartPage»

Переходим на страницу «MainRegPage». Импортируем пространство имен «materialDesign» и переходим к разметке (Разметка «MainWindowLoginPage» приведена на рисунках 1.35 – 1.39)

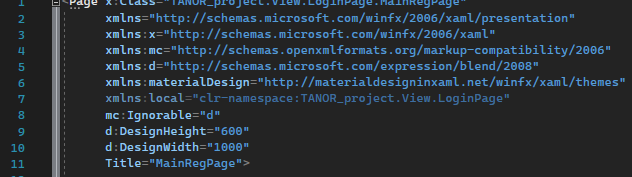


Рисунок 1.35 – Разметка «MainRegPage»



Рисунок 1.36 – Разметка «MainRegPage»

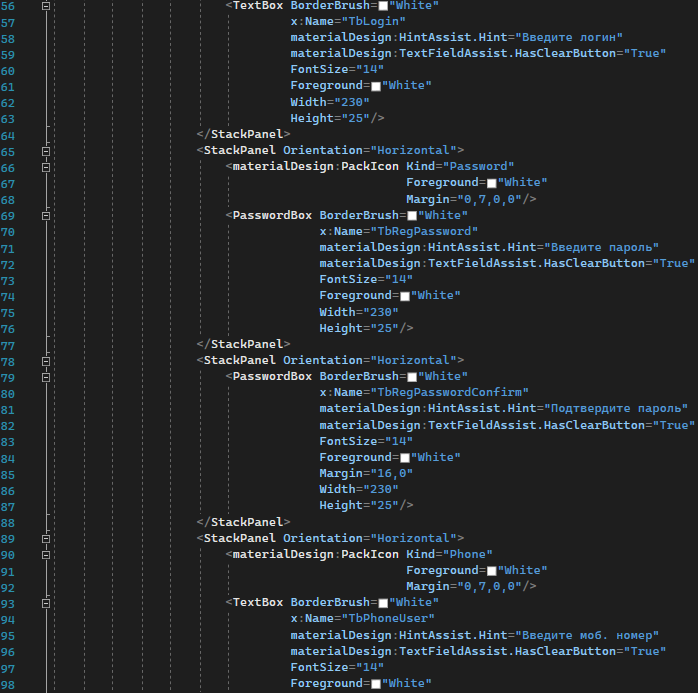
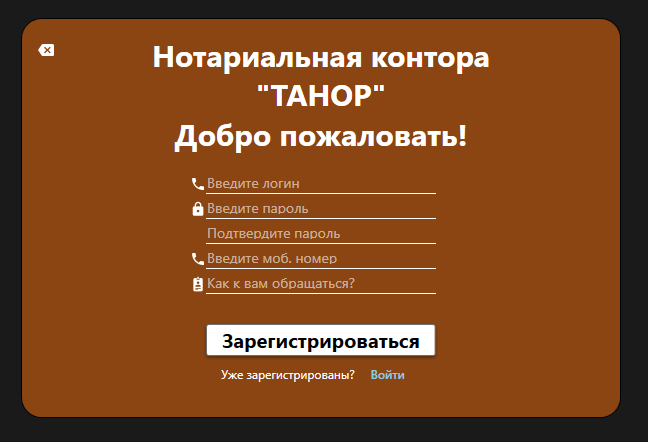


Рисунок 1.37 – Разметка «MainRegPage»



Рисунок 1.38 – Разметка «MainRegPage»



1.39 – Результат разметки «MainLoginPage»

Далее переходим к файлу «MainRegPage.xaml.cs». Этот файл будет содержать логику регистрации при клике на кнопку «Зарегистрироваться» (Реализация логики приведена на рисунках 1.40 – 1.41). Нажав по ней, или по кнопку «Войти» так или иначе будет происходить переход на «MainLoginPage».

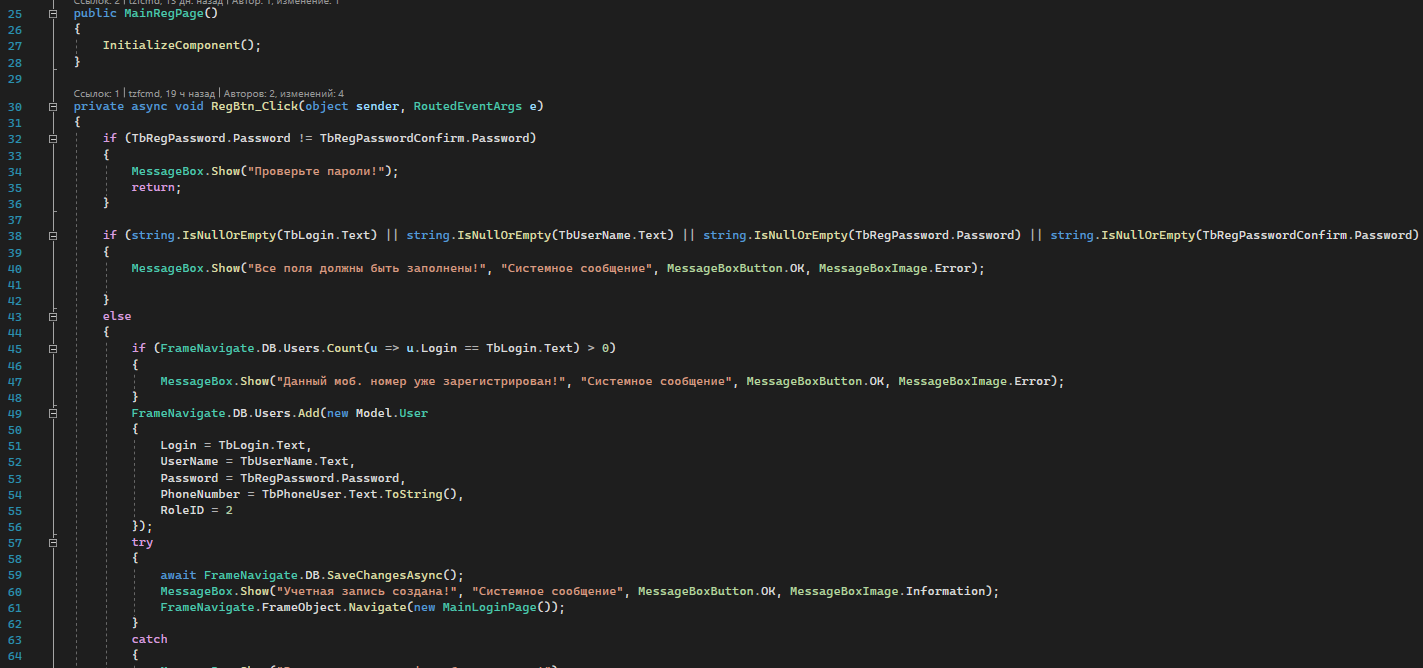


Рисунок 1.40 – Реализация логики «MainRegPage»

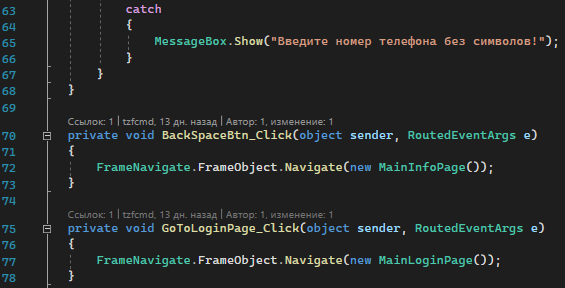


Рисунок 1.41 – Реализация логики «MainRegPage»

Переходим на страницу «MainLoginPage». Импортируем пространство имен «materialDesign» и переходим к разметке (Разметка «MainLoginPage» приведена на рисунках 1.42 – 1.45)

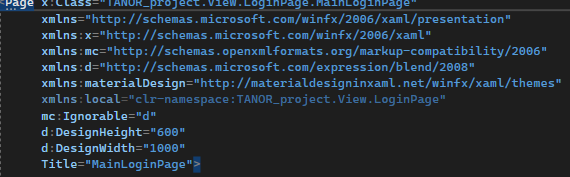


Рисунок 1.42 – Разметка «MainLoginPage»



Рисунок 1.43 – Разметка «MainLoginPage»

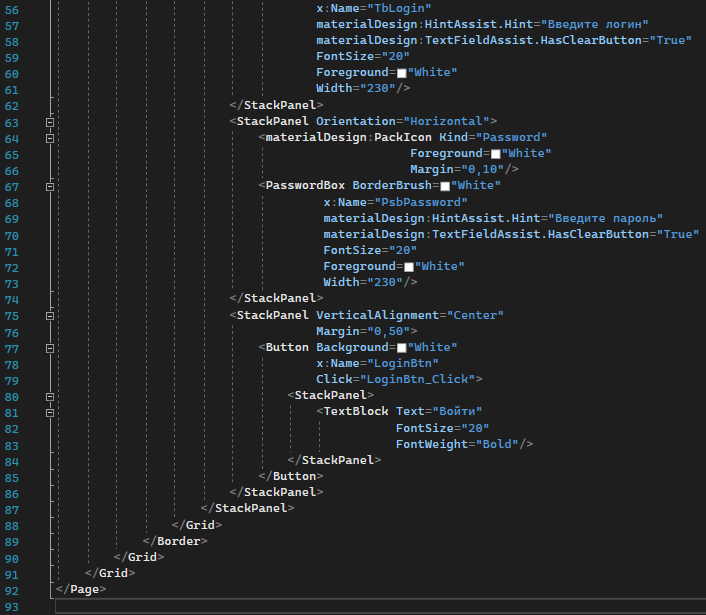


Рисунок 1.44 – Разметка «MainLoginPage»

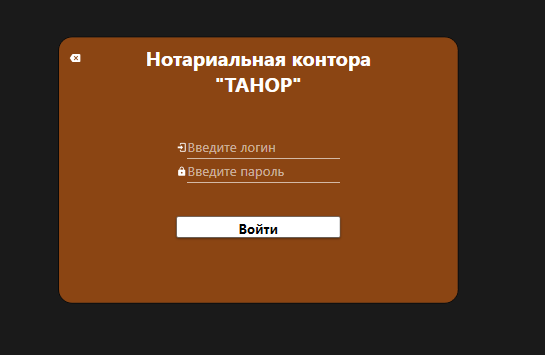


Рисунок 1.45 – Результат разметки «MainLoginPage»

Далее переходим в файле «MainLoginPage.xaml.cs». Этот файл будет содержать логику авторизации при клике на кнопку «Войти» (Реализация логики приведена на рисунках 1.46 – 1.47). В зависимости от роли пользователя, он будет попадать на «MainUserPage» / «MainLawyerPage» / »DetailOrderInfo»

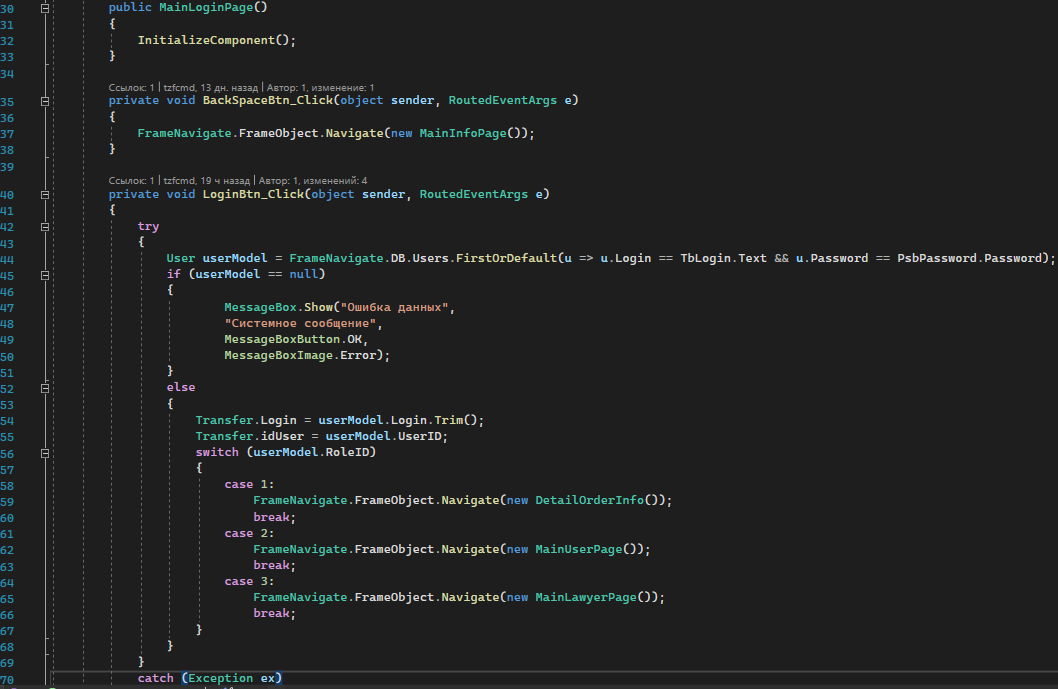


Рисунок 1.46 – Реализация логики «MainLoginPage»

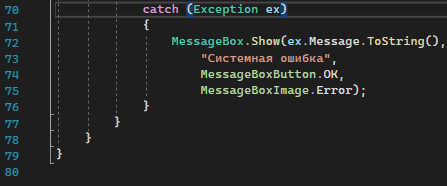


Рисунок 1.47 – Реализация логики «MainLoginPage»

**2 Реализация MainUserPage, UserCreateOrder**

Если обычный пользователь удачно авторизовался, то он попадает на «MainUserPage», где отображаются его активные заказы, а также имеются кнопки «Выйти», и «Заказать услугу» - которая переводит пользователя на страницу для оформения заказа «UserCreateOrder».

Разметка и реализация логики представлена на рисунках 2.1-2.5

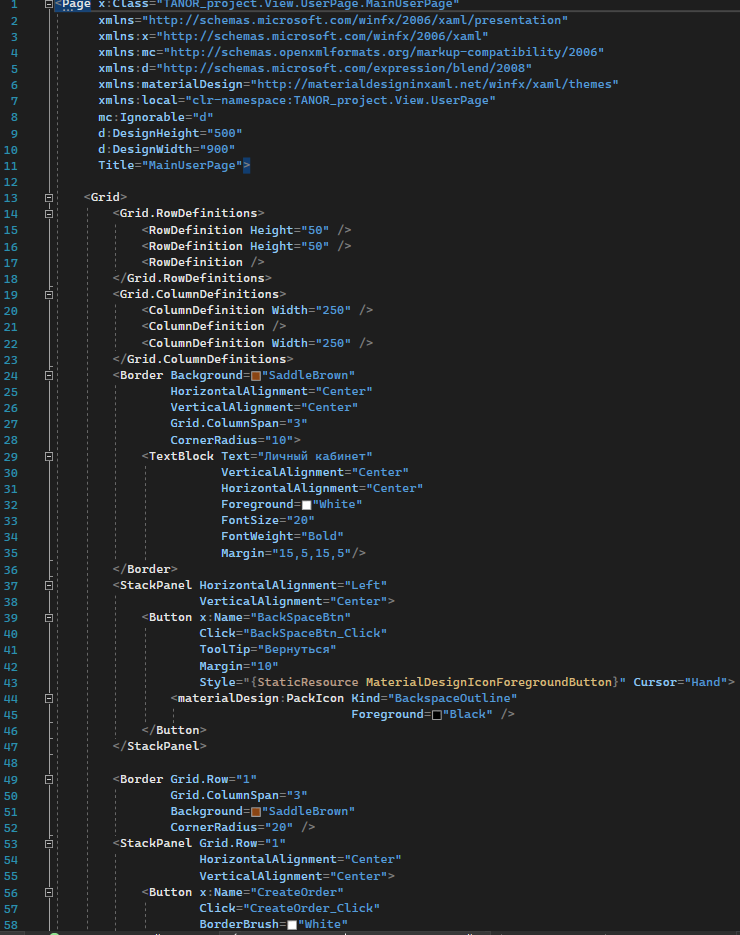


Рисунок 2.1 – разметка «MainUserPage»



Рисунок 2.2 – разметка «MainUserPage»

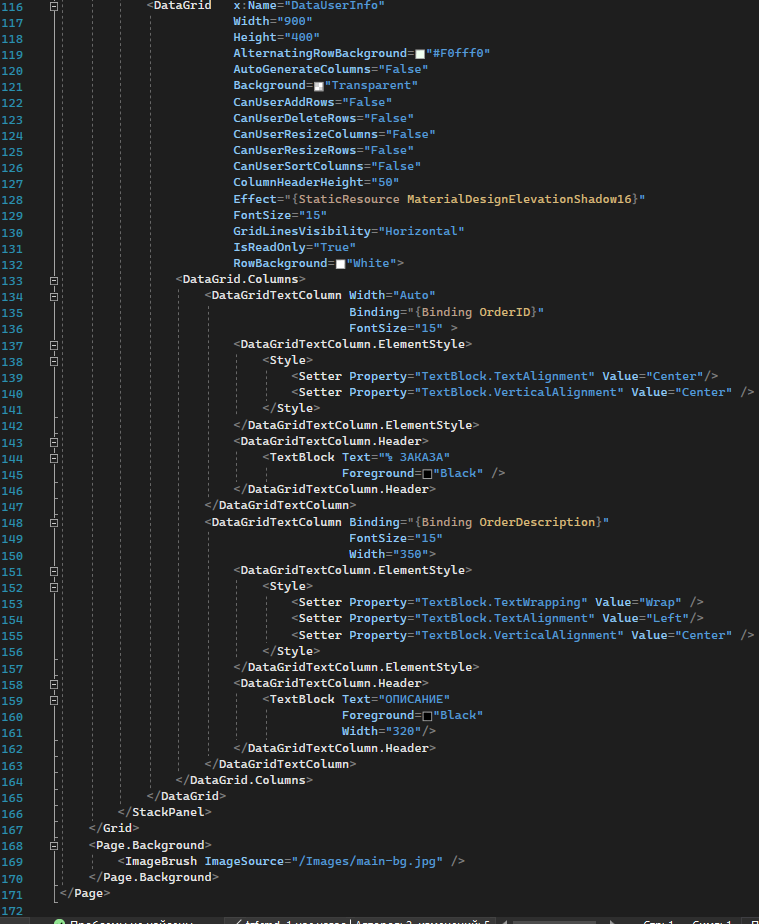


Рисунок 2.3 – разметка «MainUserPage»

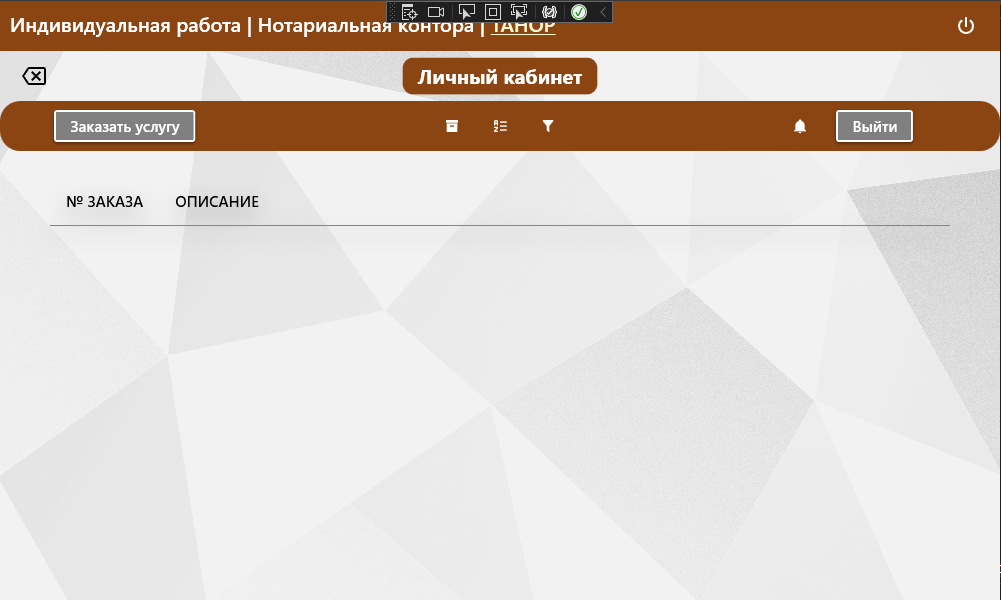


Рисунок 2.4 – результат разметки «MainUserPage»

Далее реализуем логику работы страницы, а именно кнопок «Заказать услугу», «Выйти» и «Вернуться» (представлено на рисунке 1.37)

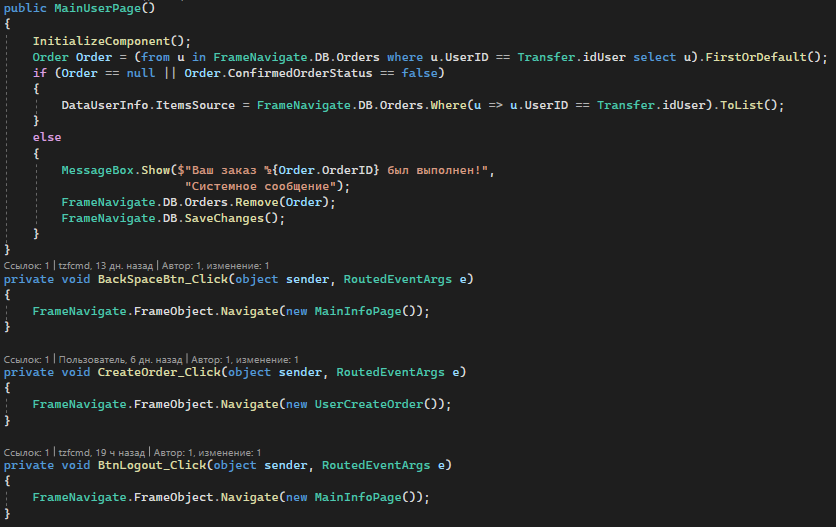


Рисунок 2.5 – реализация логики «MainUserPage»

Переходим к разметке и реализации «UserCreateOrder», где будет происходить оформление заказа.

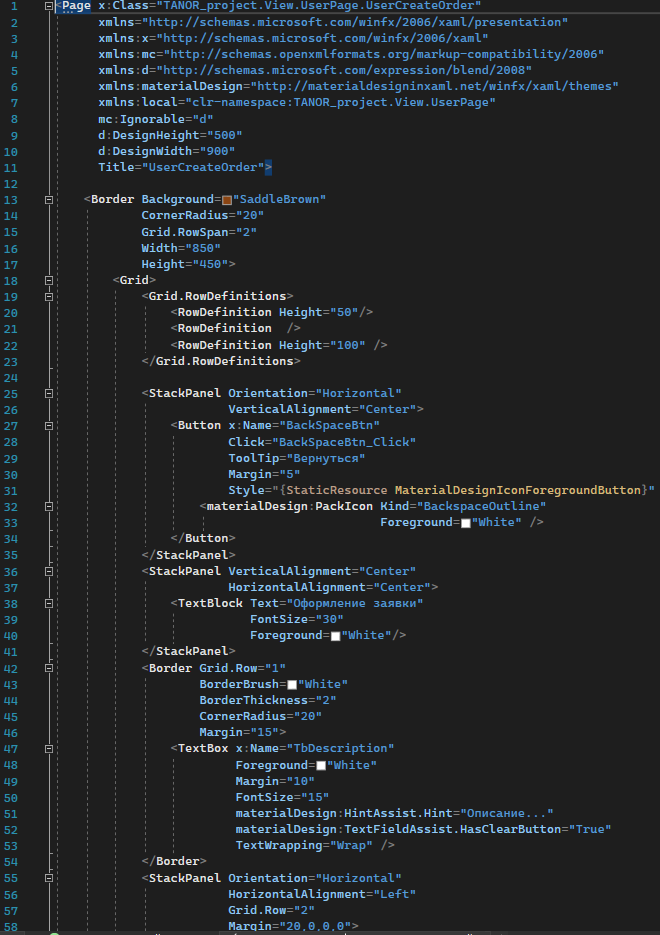


Рисунок 2.6 – разметка «UserCreateOrder»



Рисунок 2.7 – разметка «UserCreateOrder»

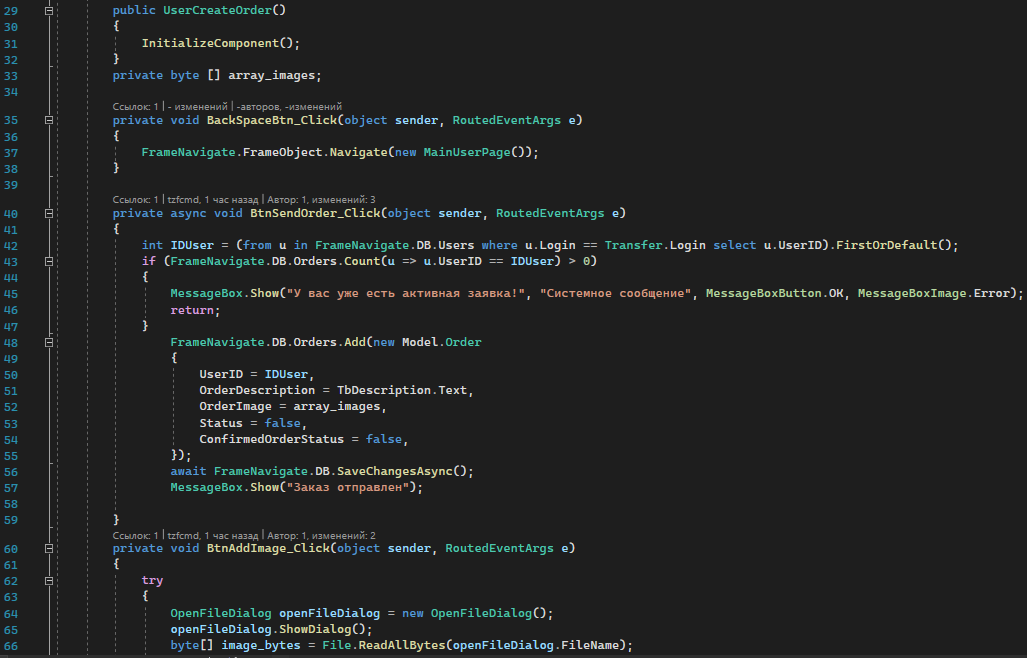


Рисунок 2.8 – реализация логики «UserCreateOrder»

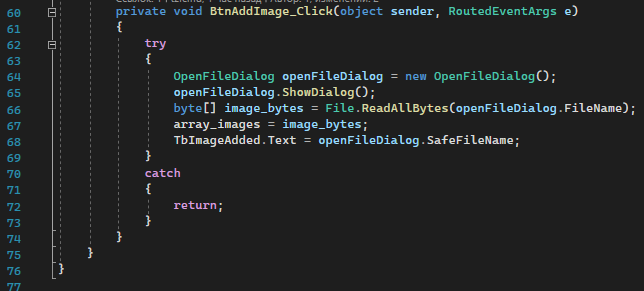


Рисунок 2.9 – реализация логики «UserCreateOrder»

**3 Реализация «LawyerPage» и «AdminPage»**

Отправленный пользователем заказ попадает сначала к Администратору («DetailOrderInfo»), который его проверяет и либо отклоняет, либо направляет его уже самому сотруднику нотариуса («MainLawyerPage»), который им занимается. На странице администратора, есть кнопки «Полное описание заказа», «Подтвердить» и «Отклонить», у нотариуса же похожий вид страницы, но нет кнопки «Отклонить», а есть только «Подвердить», которая отвечает за то, что нотариус выполнил свою работу и пользователь может приехать в офис нотариуса и забрать его документы (или пр.). У Администратора и Нотариуса есть свои пользовательские элементы интерфейса («AdminFullOrderDescriptionUC», «LawyerFullOrderDesctiptionUC»).



Рисунок 3.1 – разметка «DetailOrderInfo»



Рисунок 3.2 – разметка «DetailOrderInfo»

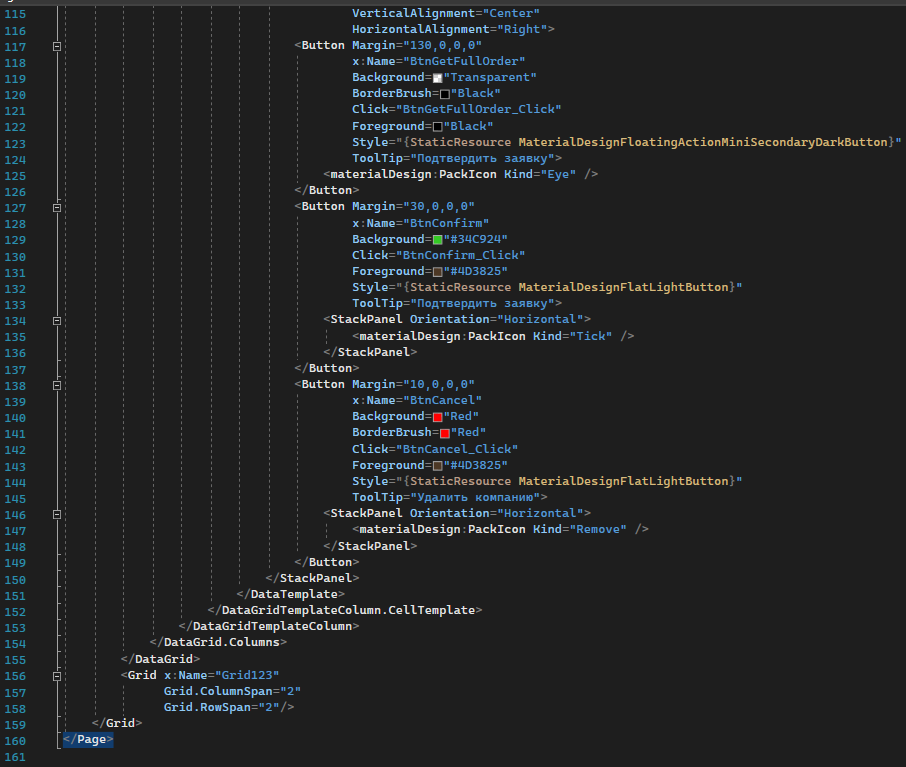


Рисунок 3.3 – разметка «DetailOrderInfo»

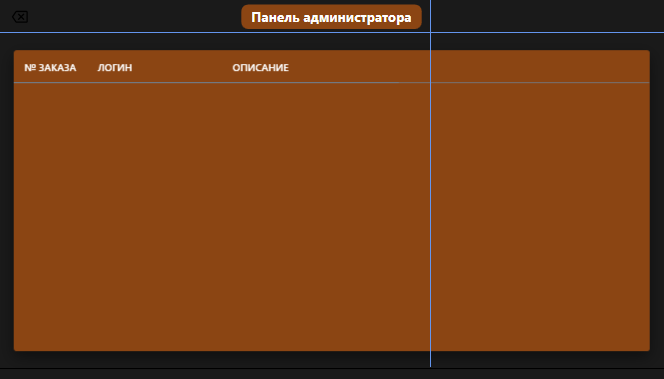


Рисунок 3.4 – Результат разметки «DetailOrderInfo»

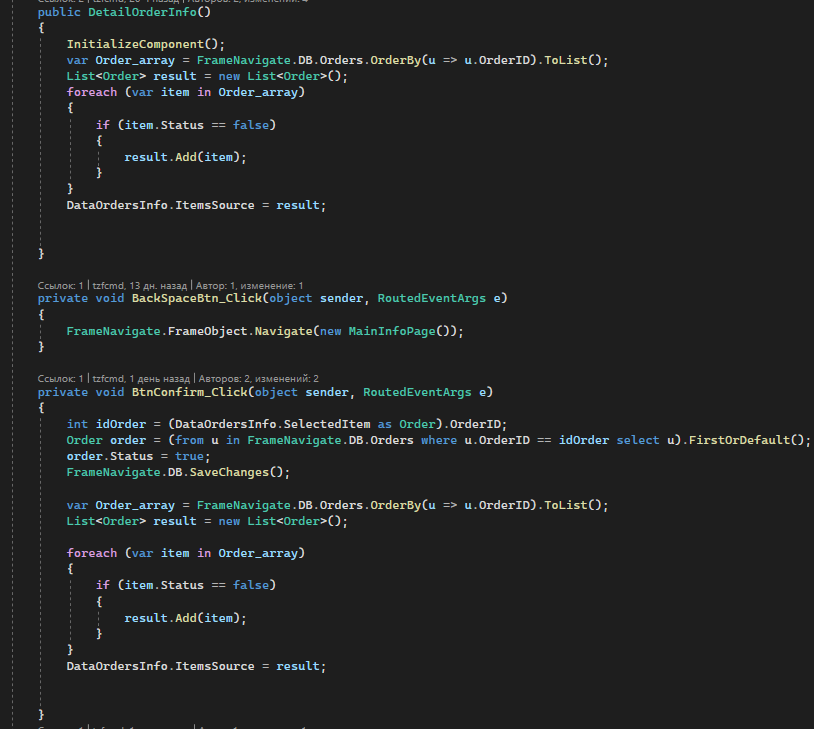


Рисунок 3.5 – реализация логики «DetailOrderInfo»

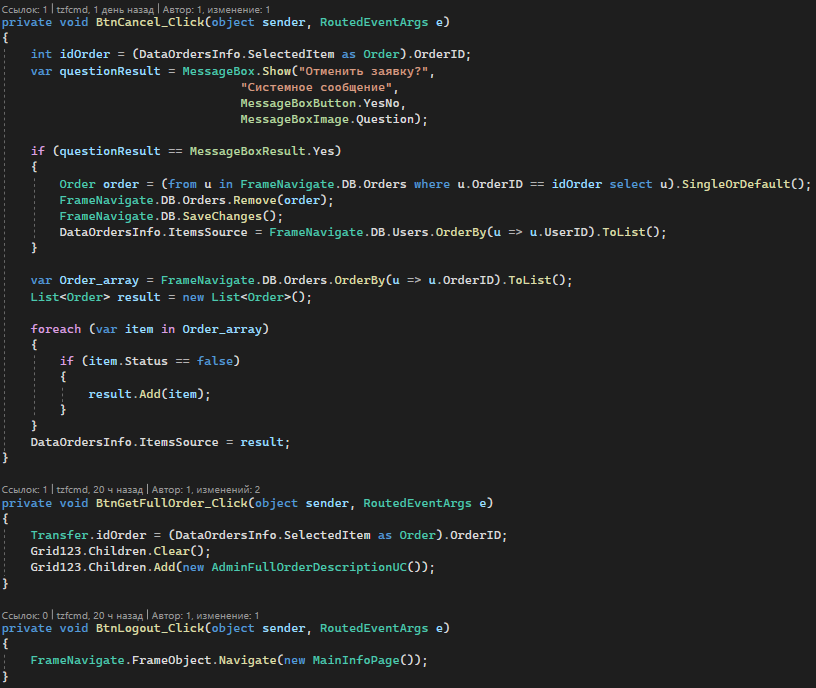


Рисунок 3.6 – реализация логики «DetailOrderInfo»

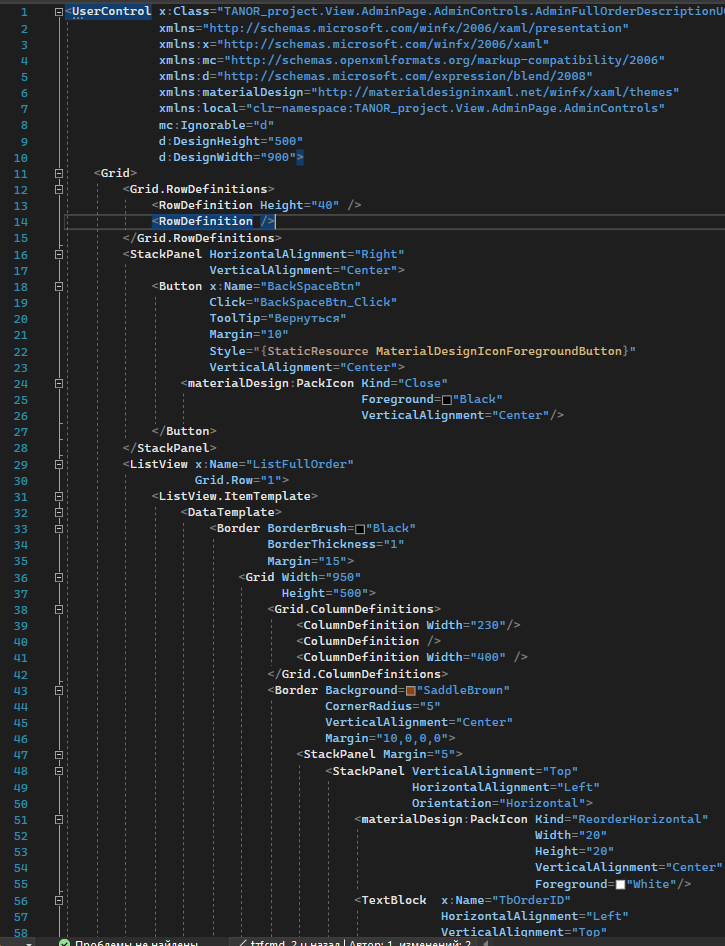


Рисунок 3.7 – разметка «AdminFullOrderDescriptionUC»



Рисунок 3.8 – разметка «AdminFullOrderDescriptionUC»

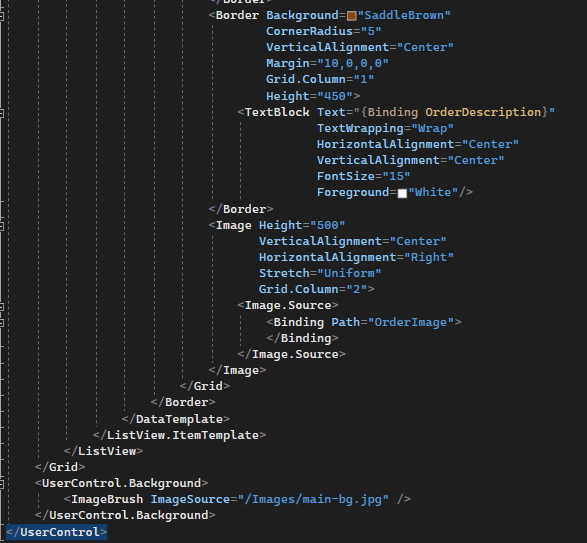


Рисунок 3.9 – разметка «AdminFullOrderDescriptionUC»



Рисунок 3.10 – Результат разметки «AdminFullOrderDescriptionUC»

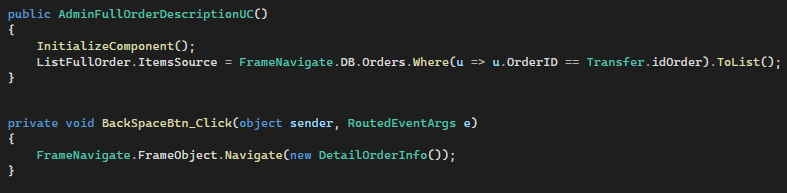


Рисунок 3.11 – реализация логики «AdminFullOrderDescriptionUC»



Рисунок 3.12 – разметка «LawyerFullOrderDesctiptionUC»



Рисунок 3.13 – разметка «LawyerFullOrderDesctiptionUC»

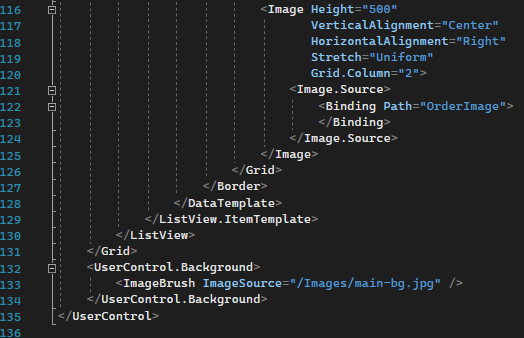


Рисунок 3.14 – разметка «LawyerFullOrderDesctiptionUC»

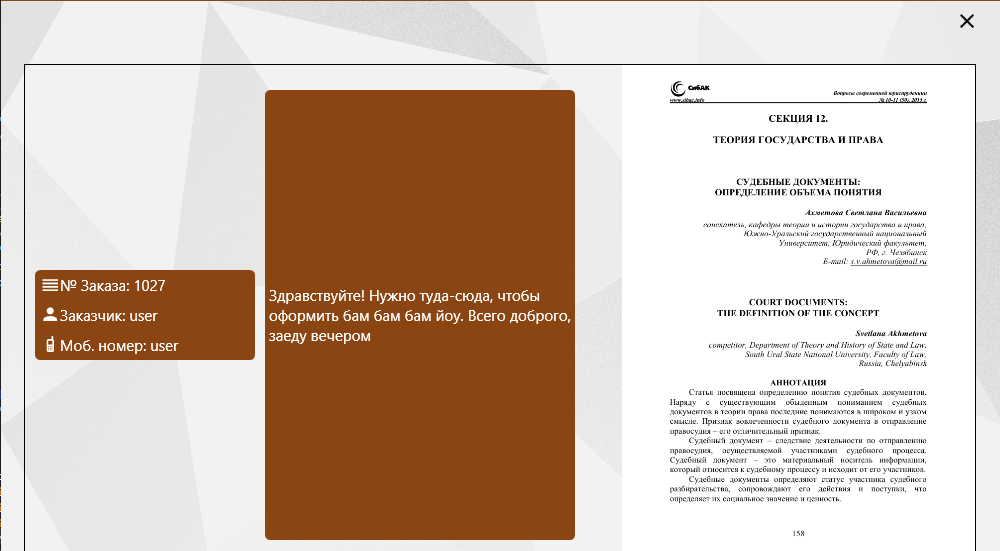


Рисунок 3.14 – Результат разметки «LawyerFullOrderDesctiptionUC»

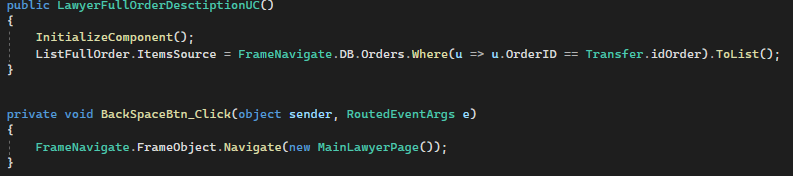


Рисунок 3.15 – Реализация логики «LawyerFullOrderDesctiptionUC»